Package 'nlr'

July 31, 2019

Type Package

Title Nonlinear Regression Modelling using Robust Methods

Version 0.1-3

Date 2019-07-30

Author Hossein Riazoshams

Maintainer Hossein Riazoshams <riazihosein@gmail.com>

Description Non-Linear Robust package is developed to handle the problem of outliers in nonlinear regression, using robust statistics. It covers classic methods in nonlinear regression as well. It has facilities to fit models in the case of auto correlated and heterogeneous variance cases, while it include tools to detecting outliers in nonlinear regression. (Riazoshams H, Midi H, and Ghilagaber G, (2018, ISBN:978-1-118-73806-1). Robust Nonlinear Regression, with Application using R, John Wiley and Sons.)

License GPL-2

LazyData yes

Imports MASS,nlme, robcor, TSA,tseries,stats, GA, quantreg

Depends R (>= 3.6.0), methods

URL http://www.riazoshams.com/nlr/

NeedsCompilation no

Repository CRAN

Date/Publication 2019-07-31 12:40:02 UTC

R topics documented:

`atyps` *Find atypical points.*

Description

This is extension of `nlout` for which add the jacobian leverage to the output. It is designed for feature development.

Usage

```
atyps(nlfited)
```

Arguments

`nlfited`	An `nl.fitt`, `nl.fitt.gn`, `nl.fitt.rob`, `nl.fitt.rgn` obejct after a nonlinear model fitted before.

Details

Compute outlier detection measures by calling `nlout` function and the ncompute jacobian leverages then dispatches the output.

Value

List of jacobian leverage and list of outlier detection measures derived by `nlout`.

- jlev`jlev` jacobian leverages.
- nloutlist of output from `nlout` function.

Note

Used mostly for internal purposes.

Author(s)

Hossein Riazoshams, Jan 2010. Email: <riazihosein@gmail.com> URL `http://www.riazoshams.com/nlr/`

References

Riazoshams H, Midi H, and Ghilagaber G, 2018,. Robust Nonlinear Regression, with Application using R, Joh Wiley and Sons.

See Also

nlsqr

Examples

```
#Example
#d<-list(xr=Weights$Date, yr=Weights$Weight)
# wmodel <- nlr(nlrobj1[[2]],data=d,control=nlr.control(method = "OLS",trace=TRUE))
 #a=nlout(wmodel)
#atyps(wmodel)
```

---

bmn.ir *Iran Broad Mony data.*

Description

Iran Broad Mony data. Broad money (current LCU)

Usage

```
bmn.ir
```

Format

The format is: chr "bmn.ir"

- yearyear
- bmn Broad money

Details

Iran Broad Mony data. Broad money (current LCU) from 1960 to 2010.

Source

http://www.worldbank.org/

References

worldbank.com

Examples

```
data(bmn.ir)
## maybe str(bmn.ir) ; plot(bmn.ir) ...
```

callorNULL-class *Class* "or classes"

Description

A set of or classes for compatibility purposes. It is used to be able to identify null values.

- callorNULLcall or NULL class.
- characterorNULL character or null.
- expressionorNULL expression or null.
- fittmethodorNULL fittmethod or null.
- functionorNULL function or null.
- integerorNULL integer or null.
- listorNULL list or null.
- logicalorNULL logical or null.
- matrixororNULL matrix or null.
- nl.fitt.roborNULL nl.fitt.rob or null.
- fittorNULL fitt or null.
- nl.formorNULL nl.form or null.
- nl.numericorNULL numeric or null.
- vectororMatrix vector or Matrix.
- vectororNULL vector or null.

Objects from the Class

A virtual Class: No objects may be created from it.

methods

No methods defined with class "callorNULL" in the signature.

Note

or classes are created in `nlr` for compatibility purpose with `splus`.

Author(s)

Hossein Riazoshams, 2013. Email: <riazihosein@gmail.com> URL http://www.riazoshams.com/nlr/

References

Riazoshams H, Midi H, and Ghilagaber G, 2018,. Robust Nonlinear Regression, with Application using R, Joh Wiley and Sons.

See Also

nlr

Examples

```
setClassUnion("nl.fittorNULL", c("nl.fitt", "NULL"))
```

---

carbon *Carbon Dioxide data.*

Description

Carbon dioxide traped in iceberg during history.

Usage

```
data(carbon)
```

Format

The format is: data.frame chr "carbon"

- year: year of gas traped in iceberg.
- co2: measured Carbon Dioxide.

Details

UNEP (1989) presented the Methane Gas and Carbon Dioxide Gas collected from the Gas trapped in icebergs in south pole from 8000 years ago.

Source

UNEP (1989), Environmental data report / prepared for UNEP by the GEMS Monitoring and Assessment Research Centre, London, UK, in co-operation with the World Resources Institute, Washington, D.C.

References

Riazoshams, H., Miri, H., (2013) Application of Robust Nonlinear Regression, case study for modeling the greenhouse gases, Methane and Carbon Dioxide concentration in atmosphere. International Conference on Mathematical Science and Statistics (ICMSS 2013), Kula Lumpur, Malaysia.

Examples

```
data(carbon)
 carbon$year
 carbon$co2
 crbdt<-list(xr=nlr::carbon$year,yr=nlr::carbon$co2)
 ScalExp<- convexpr2nlform(yr ~ p1 + exp(-(p2 - p3 * xr)),
                          selfStart=function(data){
                              y1 <-as.double(data$yr)
                              p1<-min(y1)
                              y<-log(y1-p1+10*.Machine$double.eps)
                              x<-as.double(data$xr)
                              b1<-lm(y~x)
                              p2<- -b1$coefficients[1]
                              p3<- b1$coefficients[2]
                              return(list(p1=p1,p2=p2,p3=p3))
                          },
                          name="Scaled Exp convex",
                          start=list(p1=700,p2=21,p3=0.01)
)
 carbon.ols <- nlr(formula=ScalExp, data=crbdt,
                   control=nlr.control(method="OLS"))
plot(carbon.ols,control=nlr.control(history=TRUE))
```

---

convexpr2nlform *Convert* expression *to* nl.form

Description

Convert two sided (or one sided) expression formula to `nl.form` object using `derive3` from `MASS` library.

Usage

```
convexpr2nlform(form, namesdata=NULL, start, inv = NULL, name="User Defined",...)
```

Arguments

`form`	Must be one sided expression (defined by ~formula) or two sided (response~predictor), nonlinear regression function, include parameters, response and predictor variables.
`namesdata`	optional character vector of name of data include independent and possibly dependent in two sided fomula.
`start`	list of parameters, for which the gradinet and hessian will be computed.
`name`	A character name for the model
`inv`	inverse of the nonlinear functin model
`...`	Ane extra argument pass to `nl.form`

Details

nlr package is gradient based algorithm, is based nl.form object in which gradient and hessian is available. If a nonlinear regression model formula is one sided or two sided formula and its gradient and hessian exist, the convexpr2nlform convert it to nl.form object by calling derive3 from MASS library. Although the existence of derivative is strong assumption but using advance programs can acheive high precision computing.

Value

nl.form object of the nonlinear regression function.

formula:	formula one sided or two sided with gradinet and hessian as attribute.
formtype:	="formula"
p:	=length(start) is number of parameters.
name:	="User Defined"
par:	=start parameters.
dependent	character vector of name of dependent variable.
independent:	character vector of name of independent variable.
origin:	=form

Note

If the derivatives does not exist in nlr function eplicitly the derivative option must set to derivative free. The namesdata is not functional in this version, implemented for further development. The name of parameters will be constructed from start arguments and the name of independent and dependent variables will be derived from the rest of variables embeded in the form expression.

Author(s)

Hossein Riazoshams, May 2014. Email: <riazihosein@gmail.com> URL http://www.riazoshams.com/nlr/

References

Rizo ML 2008 Statistical Computing with R The R Series. Chapman & Hall/CRC The R Series.

See Also

nl.form, nlr

Examples

```
## The function is currently defined as
nlf=convexpr2nlform(yr ~ (a)*(exp(-b*xr)-exp(-c*xr)), start = list(a=.05,b=4.39,c=21.6))
nlf
```

convfkt2nlform *Convert* fktlist *objects to* nl.form.

Description

Convert fktlist objects defined by Bunke et.al to nl.form. It calculate gradient and hessian using derive3 function.

Usage

```
convfkt2nlform(fktlistex, namesdata = NULL)
```

Arguments

fktlistex	fktlistex object include fkt, par, p, ccode, lambda, case, inv, name, defined by Bunke et al (1998)
namesdata	Name of data, is not functional now, used for feature development.

Details

fktlist defined by Bunke et al (1998) convfkt2nlform function convert this object to nl.form object.

Value

nl.fomr object with gradient and hessian attributes.

Note

Derivatives must not be included in fktlistext object will be added as attributes to response or independent variables.

Author(s)

Hossein Riazoshams, May 2014. Email: <riazihosein@gmail.com> URL http://www.riazoshams.com/nlr/

References

Bunke, O., Droge, B., Polzehl, J. Splus tools for model selection in nonlinear regression (1998) Computational Statistics, 13 (2), pp. 257-281.

See Also

See Also as convexpr2nlform, nl.form.

Examples

```
## fktlistex created by Bunke et.al
convfkt2nlform(fktlistex1[[1]])
```

`cow` *Cow Data*

Description

Milk production amount for a single cow within a year.

Usage

```
data("cow")
```

Format

The format is: data.frame chr "cow"

- Day: Day from 1 to end of the year.
- Milk: Milk produced in Litre.

Details

The data collected within a year of a milk roductin from a fluk of cows. `cow` variable is reported for a single cow.

Source

A cow farm production in Fars province of iran.

References

Riazoshams H, Midi H, and Ghilagaber G, 2018,. Robust Nonlinear Regression, with Application using R, Joh Wiley and Sons.

Examples

```
data(cow)
```

`curvature` *Calculate (IE) Intrinsic curvature and (PE) Parameter curvature.*

Description

IE and PE are measures to identify the linear approximation of nonlinear model is appropriate or no. This function may not be called explicitly by user.

Usage

```
curvature(gradient, hessian, sigma)
```

Arguments

`gradient`	n by p gradient of fited model.
`hessian`	n by p by p array of hessian for the nonlinear model.
`sigma`	estimated standard deviation.

Details

Gauss Newton method of estimation is based on linear approximation to nonlinear model. The linear approximation to function might not be appropriate. PE and IE is used to identify the parameter effect and intrinsic effect of model. Big values represent the linear approximation to nonlinear model is not correct.

Value

List of curvature values.

- pe: Parameter Effect curvature.
- int: Intrinsic effect curvature.
- a: A matrix.
- cutf: cut of point

$$1/\sqrt{F(.95, p, n-p)}$$

, if PE or IE be biger than cut of point then either of them has larg curvature.

Note

curvature is a model checking tool. From the OLS estimate output included carvature that can be accessed by `curvature` slot of the output object, therefore do not need to be called explicitly by user.

Author(s)

Hossein Riazoshams, May 2014. Email: <riazihosein@gmail.com> URL `http://www.riazoshams.com/nlr/`

References

Bates, D.M., and Watts, D. G. (1980). Relative curvature measures of nonlinearity, J. R. statistic. Ser. B 42: 1-25.

Examples

```
 crbdt<-list(xr=nlr::carbon$year,yr=nlr::carbon$co2)
 ScalExp<- convexpr2nlform(yr ~ p1 + exp(-(p2 - p3 * xr)),
                        selfStart=function(data){
                            y1 <-as.double(data$yr)
                            p1<-min(y1)
                            y<-log(y1-p1+10*.Machine$double.eps)
                            x<-as.double(data$xr)
                            b1<-lm(y~x)
                            p2<- -b1$coefficients[1]
                            p3<- b1$coefficients[2]
                            return(list(p1=p1,p2=p2,p3=p3))
                        },
                        name="Scaled Exp convex",
                        start=list(p1=700,p2=21,p3=0.01)
)
 carbon.ols <- nlr(formula=ScalExp, data=crbdt,
                   control=nlr.control(method="OLS"))
carbon.ols$curvature
```

---

db.Fault *Fault database*

Description

Codes of errors used in "nlr" functions.

Usage

```
db.Fault
```

Format

The format is:data.frame chr "db.Fault"

- FL: (Fault Logic) is true if mesage is error and program terminate, False if message is warning and result might not be accurate.
- FN: (Fault Number) is a code for the error mesage.
- FT: (Fault Text) is error text.
- FF: (Fault File) is the first origin of fault createor, but during the computation will represent the function that raised error.

Details

db.Fault is the database of errors that might occure in all functions of the nlr package. Termination of the functions can be due to error or warning. If error happened no output is returned, if warning happened output returned but might not be reliable. Warning kight happen for example if number of iteration exceeded the maximum number of iteration. Then a sugestion will be displayed for remedy the problem. The returned variable is a "Fault" that display the mesage code, number, and file that error happened.

Source

Robust Nonlinear Regression, Theories and Methods with Practical Guides for R Packages. Riazoshams et al.

References

Riazoshams H, Midi H, and Ghilagaber G, 2018,. Robust Nonlinear Regression, with Application using R, Joh Wiley and Sons.

Examples

```
data(db.Fault)    # load data
db.Fault         # variable name
```

---

db.method *methods database*

Description

Database for methods used in "nlr" package functions.

Usage

```
db.method
```

Format

The format is:data.frame chr "db.method"

- `methodID`: Object of class `"numeric"` code for the method.
- `method`:Object of class `"character"` name of the method used in estimation and other computation procedures.
- `detail`:Object of class `"character"` detail text description of the method.
- `methodBR`:Object of class `"numeric"` (method Branche) branch for the method used in iteration.
- `detailBR`:Object of class `"character"` detail of the branch.
- `subroutine`:Object of class `"character"` the function, subroutine, that the result constructed from.

Details

Each function use a method for estimation, and each method depending on numerical computation have a branch. Any output object have a "fittmethod" object which the values come from "db.method" database.

Source

Robust Nonlinear Regression, Theories and Methods with Practical Guides for R Packages. Riazoshams et al.

References

Riazoshams H, Midi H, and Ghilagaber G, 2018,. Robust Nonlinear Regression, with Application using R, Joh Wiley and Sons.

Examples

```
data(db.method)         # load data
db.method              # variable name
```

---

db.methodBR *method branches database*

Description

Sub method used in procedures in functions.

Usage

```
db.methodBR
```

Format

The format is: data.frame chr "db.methodBR"

- methodBR: method branch.
- detailBR: detail and long description of the method.

Details

Any method used in functions might have some sub branch depending the procedure it used. The codes and detail is from "db.methodBR" database. The result objects from function have "fittmethod" sub object that save methods and sub methods used in the function.

Source

Robust Nonlinear Regression, Theories and Methods with Practical Guides for R Packages. Riazoshams et al.

References

Riazoshams H, Midi H, and Ghilagaber G, 2018,. Robust Nonlinear Regression, with Application using R, Joh Wiley and Sons.

Examples

```
data(db.methodBR)        # load data
db.methodBR             # variable name
## maybe str(db.methodBR) ; plot(db.methodBR) ...
```

`dfr.corrts` *Derivative free Two Stage estimate*

Description

Derivative free two stage estimate for nonlinear regreession model with autocorrelated error.

Usage

```
dfr.corrts(formula, data, start = getInitial(formula, data),
control = nlr.control(tolerance = 0.001, minlanda = 1/2^10,
maxiter = 25 * length(start)), correlation = 1, ...)
```

Arguments

`formula`	nl.form object of the nonlinear function model. See `nl.form` object.
`data`	list of data with the response and predictor as name of variable.
`start`	list of starting value parameter, name of parameters must be represented as names of variable in the list.
`control`	nlr.control object, include tolerance, maxiter,... see `nlr.control`.
`correlation`	correlation structure, at the moment parameter of AR(p) process.
`...`	any argument pass to `formula`

Details

In first stage nonlinear regression parameter estimate and in second stage autocorrelation structure estimate and finally the generalized least square estimates the function model parameters.

In this function all stages compute by derivative free methods, which minimization methods uses Nelder-Mead method.

Value

`fited`	`nl.fitt.gn` object generated by `nlsnm` function.
`tm`	fitted time series model for residuals.

Note

This function currently run with AR process. The robust estimate is don by `nl.robcorrts` function. This function will be called from `nlr` by providing `correlation` as correlation structure and `derivfree`. It is under developement and for internal use, user might not call it directly, it is more efficient to call from `nlr` function with mentioned arguments.

Author(s)

Hossein Riazoshams, Jul 2009. Email: <riazihosein@gmail.com> URL `http://www.riazoshams.com/nlr/`

References

Riazoshams, H., Midi, H., Sharipov, O. S.H, (2010). The Performance of Robust Two Stage Estimator in Nonlinear Regression with autocorrelated Error, Communications in Statistics - Simulation and Computation, 39: 1251-1268.

See Also

`nl.robcorrts`, `nlsqr.gn`, `nl.fitt.gn`, `nlr.control`, `nlsnm`

Examples

```
# The direct call of nlr call dfr.corrts.
p1<- 8.06e+10
p2<- 1e11
p3<-1970
p4=6
chstart2 <- list(p1=p1,p2=p2,p3=p3,p4=p4)
irandt<-nlr::trade.ir
dfrir<- dfr.corrts (nlrobj5[[4]],data=list(xr=irandt[,1],yr=irandt[,2]),start=chstart2,
control=nlr.control(trace=TRUE),correlation = 2)
dfrir$fited$parameters
```

`dfr.hetro` *Derivative free (CME)*

Description

(CME) Classic multi stage estimate for nonlinear regression with heteroscedastic error, when variance is function of unkown parameters. The variance function model parameter estimate using pseudo chi-square likelihood of computed sample variance. `dfr.hetro` is derivative free estimate CME.

Usage

```
dfr.hetro(formula, data, start = getInitial(formula, data),
control = nlr.control(tolerance = 1e-05, minlanda = 1/2^10,
maxiter = 25 * length(start)), varmodel, tau = NULL, ...)
```

Arguments

formula	nl.form object of the nonlinear function model.
data	list of data include responce and predictor.
start	list of parameter values of nonlinear model function (θ.
control	list of nlr.control for controling convergence criterions.
varmodel	nl.fomr object of variance function model for heteroscedastic variance.
tau	list of initial values for variance model function varmodel argument.
...	extra arguments to nonlinear regression model, heteroscedastic variance function, robust loss function or its tuning constants, or optimization functions.

Details

In stage 1 the nonlinear model parameter estimates by Classic OLS, Stage 2 compute sample variance of data, Stage 3 estimate the parameter of variance function model by maximizing the chi-square pseudo-likelihood function. Stage 4 estimate the final value of function model parameter by generalized least square. For optimization the derivative free Nelder-Mead is used.

Value

generalized fitt object nl.fitt.gn. The hetro slot include parameter estimate and other information of fitt for heteroscedastic variance model.

parameters	nonlinear regression parameter estimate of θ.
correlation	of fited model.
form	nl.form object of called nonlinear regression model.
response	computed response.
predictor	computed (right side of formula) at estimated parameter with gradient and hessian attributes.
curvature	list of curvatures, see curvature function.
history	matrix of convergence history, collumns include: convergence index, parameters, minimized objective function, convergence criterion values, or other values. These values will be used in plot function in ploting history.
method	fittmethod object of method used for fitt.
data	list of called data.
sourcefnc	Object of class "callorNULL" source function called for fitt.
Fault	Fault object of error, if no error Fault number = 0 will return back.
vm	covariance matrix, diagonal of variance model predicted values.
rm	cholesky decomposition of vm.
gresponse	transformed of response by rm, include gradinet and hessian attributes.
gpredictor	transformed of predictor by rm, include gradinet and hessian attributes.
hetro	nl.fitt object of fited variance odel:

- parametersestimate of variance parameter τ

- formnl.form object of called varmodel.
- predictorvariance model computed at estimated parameter, $H(x;\hat{\tau})$
- responsesample variance computed used as response variable.
- historymatrix of convergence history, collumns include: convergence index, parameters, minimized objective function, convergence criterion values, or other values.
- methodfittmethod object of method used for fitt.
- dataresponse (z_i) and predictor t variable values, used to computing the variance model.
- sourcefncObject of class "callorNULL" source function called for fitt.
- Faultfault object of error, if no error Fault number = 0 will return back.

Note

Heteroscedastic variance can have several cases, this function assume variance is parameteric function of predictor ($H(t;\tau)$). If data does not include the predictor variable of varmodel (t), the predicted of function model $f(x;\hat{\theta})$ will replace for (t), otherwise user have to defin (t) or (x) as predictor variable of (H).

dfr.hetro is derivative free it is slow convergence, while nl.hetro is derivative based estimate is effectively fast method.

Author(s)

Hossein Riazoshams, May 2014. Email: <riazihosein@gmail.com> URL http://www.riazoshams.com/nlr/

References

Riazoshams, H,. 2010. Outlier detection and robust estimation methods for nonlinear regression having autocorrelated and heteroscedastic errors. PhD thesis disertation, University Putra Malaysia.

Riazoshams H, Midi H, and Ghilagaber G, 2018,. Robust Nonlinear Regression, with Application using R, Joh Wiley and Sons.

See Also

fittmethod, nl.form, nl.fitt, nl.fitt.gn, nl.hetro

Examples

```
ntpstart22=list(p1=.12,p2=7,p3=1,p4=160)
ntpstarttau22=list(tau1=-1.24,tau2=2.56,tau3=.03042)
datalist=list(xr=ntp$dm.k,yr=ntp$cm.k)
datalist[[nlrobjvarmdls3[[2]]$independent]]<-ntp$dm.k
ntpfit<- dfr.robhetro(formula=nlrobj1[[16]],data=datalist,start=ntpstart22,
robfunc=nl.robfuncs[["hampel"]], tau=ntpstarttau22,
varmodel=nlrobjvarmdls3[[2]],robscale=TRUE,method="NM",control=nlr.control(tolerance=1e-4,
maxiter=150))
ntpfit$parameters
```

`dfr.hetroLS` *Derivative free CLSME.*

Description

Derivative free Classic Least square based Multi Stage Estimate (CLSME) for heteroscedastic error case.

Usage

```
dfr.hetroLS(formula, data, start = getInitial(formula, data), control = nlr.control
tolerance = 1e-04, minlanda = 1/2^10,maxiter = 25 * length(start)), varmodel,
tau = getInitial(varmodel, vdata), ...)
```

Arguments

`formula`	`nl.form` object of the nonlinear function model.
`data`	list of data include responce and predictor.
`start`	list of parameter values of nonlinear model function (θ. in $f(x, \theta)$).
`control`	list of `nlr.control` for controling convergence criterions.
`varmodel`	`nl.fomr` object of variance function model for heteroscedastic variance.
`tau`	list of initial values for variance model function `varmodel` argument.
`...`	extra arguments to nonlinear regression model, heteroscedastic variance function, robust loss function or its tuning constants.

Details

Least square based estimate for nonlinear regression with hetroscedastic error when variance is a general function of unkown parameters.

Value

generalized fitt object `nl.fitt.gn`. The `hetro` slot include parameter estimate and other information of fitt for heteroscedastic variance model.

`(parameters`	nonlinear regression parameter estimate of θ.
`correlation`	of fited model.
`form`	`nl.form` object of called nonlinear regression model.
`response`	computed response.
`predictor`	computed (right side of formula) at estimated parameter with gradient and hessian attributes.
`curvature`	list of curvatures, see `curvature` function.
`history`	matrix of convergence history, collumns include: convergence index, parameters, minimized objective function, convergence criterion values, or other values. These values will be used in `plot` function in ploting history.

method	fittmethod object of method used for fitt.
data	list of called data.
sourcefnc	Object of class "callorNULL" source function called for fitt.
Fault	Fault object of error, if no error Fault number = 0 will return back.
vm	covariance matrix, diagonal of variance model predicted values.
rm	cholesky decomposition of vm.
gresponse	transformed of response by rm, include gradinet and hessian attributes.
gpredictor	transformed of predictor by rm, include gradinet and hessian attributes.
hetro	nl.fitt object of fited variance odel:

- parametersestimate of variance parameter τ
- formnl.form object of called varmodel.
- predictorvariance model computed at estimated parameter, $H(x;\hat{\tau})$
- responsesample variance computed used as response variable.
- historymatrix of convergence history, collumns include: convergence index, parameters, minimized objective function, convergence criterion values, or other values.
- methodfittmethod object of method used for fitt.
- dataresponse (z_i) and predictor t variable values, used to computing the variance model.
- sourcefncObject of class "callorNULL" source function called for fitt.
- FaultFault object of error, if no error Fault number = 0 will return back.

Note

Heteroscedastic variance can have several cases, this function assume variance is parameteric function of predictor ($H(t;\tau)$). If data does not include the predictor variable of varmodel (t), the predicted of function model $f(x;\hat{\theta})$ will replace for (t), otherwise user have to defin (t) or (x) as predictor variable of (H).

dfr.hetroLS is derivative free it is slow convergence, while nl.hetroLS is derivative based estimate is effectively fast method. Since it is slow algorithm it is recomneded to use larger values for maximum number of iterations in nlr.control options.

Author(s)

Hossein Riazoshams, May 2014. Email: <riazihosein@gmail.com> URL http://www.riazoshams.com/nlr/

References

Riazoshams, H. (2012), Robustifying the Least Squares estimate of parameters of variance model function in nonlinear regression with heteroscedastic variance, Poster Presentation, Royal Statistical Society Conference (RSS) 2012, Telford, UK.

See Also

fittmethod, nl.form, nl.fitt, nl.fitt.gn, nl.hetroLS, nlr.control

Examples

```
ntpstart=list(p1=.12,p2=6,p3=1,p4=33)
ntpstarttau=list(tau1=-.66,tau2=2,tau3=.04)
datalist=list(xr=ntp$dm.k,yr=ntp$cm.k)
htls<- dfr.hetroLS(formula=nlrobj1[[15]], data=datalist, start= ntpstart,tau=ntpstarttau,
varmodel=nlrobjvarmdls3[[2]],control=nlr.control(tolerance=1e-8))
htls$parameters
```

`dfr.robhetro` *Derivative free (RME)*

Description

(RME) for nonlinear regression with heteroscedastic variance, when the variance of error is general parameteric function of unkown parameters. Robust form of CME (See `dfr.hetro`).

Usage

```
dfr.robhetro(formula, data, start = getInitial(formula, data), control = nlr.contro
tolerance = 1e-05, minlanda = 1/2^10, maxiter = 100 * length(start)), robfunc, varm
tau = NULL, method = "NLM", ...)
```

Arguments

`formula`	`nl.form` object of the nonlinear function model.
`data`	list of data include responce and predictor.
`start`	list of parameter values of nonlinear model function (θ).
`control`	list of `nlr.control` for controling convergence criterions.
`robfunc`	nl.form object of robust function used for downgrading.
`varmodel`	`nl.fomr` object of variance function model for heteroscedastic variance.
`tau`	list of initial values for variance model function `varmodel` argument.
`method`	="NLM" means using `nlmest.NLM` function,or ="NM" means using derivative free `nlmest.NM` function
`...`	extra arguments to nonlinear regression model, heteroscedastic variance function, robust loss function or its tuning constants.

Details

In stage 1 the nonlinear model parameter estimates by robust MM-estimate, Stage 2 compute robust sample variance of data, Stage 3 estimate the parameter of variance function model by maximizing the robustified form of chi-square pseudo-likelihood function. Stage 4 estimate the final value of function model parameter by generalized robust MM-estimate.

Value

`nl.fitt.rgn` for heterogeneous and autocorrelated error (nonlinear fitt robust generalized) will return.

`parameters`	nonlinear regression parameter estimate of θ.
`correlation`	of fited model.
`form`	`nl.form` object of called nonlinear regression model.
`response`	computed response.
`predictor`	computed (right side of formula) at estimated parameter with gradient and hessian attributes.
`curvature`	list of curvatures, see `curvature` function.
`history`	matrix of convergence history, collumns include: convergence index, parameters, minimized objective function, convergence criterion values, or other values. These values will be used in `plot` function in ploting history.
`method`	`fittmethod` object of method used for fitt.
`data`	list of called data.
`sourcefnc`	Object of class `"callorNULL"` source function called for fitt.
`Fault`	`Fault` object of error, if no error Fault number = 0 will return back.
`htheta`	robust loss value including gradient and hessain attributes.
`rho`	computed robust rho function, including gradient and hessain attributes.
`ri`	estimated residuals, including gradient and hessain attributes.
`curvrob`	curvature
`robform`	`nl.form` object of robust loss rho function.
`vm`	covariance matrix, diagonal of variance model predicted values.
`rm`	cholesky decomposition of vm.
`gresponse`	transformed of response by rm, include gradinet and hessian attributes.
`gpredictor`	transformed of predictor by rm, include gradinet and hessian attributes.
`hetro`	`nl.fitt.rob` object of fited variance odel: • parametersestimate of variance parameter τ • form`nl.form` object of called `varmodel`. • predictorvariance model computed at estimated parameter, $H(x;\hat{\tau})$ • responsesample variance computed used as response variable. • historymatrix of convergence history, collumns include: convergence index, parameters, minimized objective function, convergence criterion values, or other values. • method`fittmethod` object of method used for fitt. • dataresponse (z_i) and predictor `t` variable values, used to computing the variance model. • sourcefncObject of class `"callorNULL"` source function called for fitt. • Fault`Fault` object of error, if no error Fault number = 0 will return back. • hthetarobust loss value including gradient and hessain attributes, for variance model. In fact is loglikelihood values. • rhocomputed robust rho function, including gradient and hessain attributes.
`others`	$refvar reference variance. variance of zi's.

Note

Heteroscedastic variance can have several cases, this function assume variance is parameteric function of predictor ($H(t;\tau)$). If data does not include the predictor variable of `varmodel` (t), the predicted of function model $f(x;\hat{\theta})$ will replace for (t), otherwise user have to defin (t) or (x) as predictor variable of (H).

This function is derivative free form of `nl.robhetro` and robust form of `dfr.hetro`. Since it is slow algorithm it is recomneded to use larger values for maximum number of iterations in `nlr.control` options.

Author(s)

Hossein Riazoshams

References

Riazoshams, H,. 2010. Outlier detection and robust estimation methods for nonlinear regression having autocorrelated and heteroscedastic errors. PhD thesis disertation, University Putra Malaysia.

Riazoshams H, Midi H, and Ghilagaber G, 2018,. Robust Nonlinear Regression, with Application using R, Joh Wiley and Sons.

See Also

`dfr.hetro`, `nlr.control`, `fittmethod`, `nl.form`, `nl.fitt.rob`, `nl.fitt.rgn`, `nlr.control`

Examples

```
ntpstart=list(p1=.12,p2=6,p3=1,p4=33)
ntpstarttau=list(tau1=-.66,tau2=2,tau3=.04)
datalist=list(xr=ntp$dm.k,yr=ntp$cm.k)
rbhfitt <- dfr.robhetro(formula=nlrobj1[[16]],data=datalist,start=ntpstart,
robfunc=nl.robfuncs[["hampel"]],tau=ntpstarttau,varmodel=nlrobjvarmdls3[[2]],robscale=T,
method="NM",control=nlr.control(tolerance=1e-8))
rbhfitt$parameters
```

`dfr.robhetroLS` *Derivative free RGME.*

Description

Robust Generalized Multistage Estimate (RGME) for heteroscedastic error case, robust form of CLsME (See `dfr.hetroLS`)

Usage

```
dfr.robhetroLS(formula, data, start = getInitial(formula, data), control =
nlr.control(tolerance = 0.001, minlanda = 1/2^10,
maxiter = 25 * length(start)), robfunc, varmodel, tau = varmodel$par, method = "NM"
```

Arguments

formula	nl.form object of the nonlinear function model.
data	list of data include responce and predictor.
start	list of parameter values of nonlinear model function (θ. in $f(x, \theta)$).
control	list of nlr.control for controling convergence criterions.
robfunc	nl.form object of robust function used for downgrading.
varmodel	nl.fomr object of variance function model for heteroscedastic variance.
tau	list of initial values for variance model function varmodel argument.
method	="NLM" means using nlmest.NLM function,or ="NM" means using derivative free nlmest.NM function
...	extra arguments to nonlinear regression model, heteroscedastic variance function, robust loss function or its tuning constants.

Details

Robustified form of Least square based estimate for nonlinear regression with hetroscedastic error when variance is a general function of unkown parameters.

Value

return object nl.fitt.rgn for nonlinear regression wuth heterogeneous error.

parameters	nonlinear regression parameter estimate of θ.
correlation	of fited model.
form	nl.form object of called nonlinear regression model.
response	computed response.
predictor	computed (right side of formula) at estimated parameter with gradient and hessian attributes.
curvature	list of curvatures, see curvature function.
history	matrix of convergence history, collumns include: convergence index, parameters, minimized objective function, convergence criterion values, or other values. These values will be used in plot function in ploting history.
method	fittmethod object of method used for fitt.
data	list of called data.
sourcefnc	Object of class "callorNULL" source function called for fitt.
Fault	Fault object of error, if no error Fault number = 0 will return back.
htheta	robust loss value including gradient and hessain attributes.
rho	computed robust rho function, including gradient and hessain attributes.
ri	estimated residuals, including gradient and hessain attributes.
curvrob	curvature
robform	nl.form object of robust loss rho function.

vm covariance matrix, diagonal of variance model predicted values.

rm cholesky decomposition of vm.

gresponse transformed of response by rm, include gradinet and hessian attributes.

gpredictor transformed of predictor by rm, include gradinet and hessian attributes.

hetro `nl.fitt.rob` object of fited variance odel:

- parametersestimate of variance parameter τ
- form`nl.form` object of called `varmodel`.
- predictorvariance model computed at estimated parameter, $H(x;\hat{\tau})$
- responsesample variance computed used as response variable.
- historymatrix of convergence history, collumns include: convergence index, parameters, minimized objective function, convergence criterion values, or other values.
- method`fittmethod` object of method used for fitt.
- dataresponse (z_i) and predictor `t` variable values, used to computing the variance model.
- sourcefncObject of class `"callorNULL"` source function called for fitt.
- Fault`Fault` object of error, if no error Fault number = 0 will return back.
- hthetarobust loss value including gradient and hessain attributes, for variance model. In fact is loglikelihood values.
- rhocomputed robust rho function, including gradient and hessain attributes.

Note

Heteroscedastic variance can have several cases, this function assume variance is parameteric function of predictor ($H(t;\tau)$). If data does not include the predictor variable of `varmodel` (t), the predicted of function model $f(x;\hat{\theta})$ will replace for (t), otherwise user have to defin (t) or (x) as predictor variable of (H).

This function is derivative free form of `nl.robhetroLS` and robust form of `dfr.hetroLS`. Since it is slow algorithm it is recomneded to use larger values for maximum number of iterations in `nlr.control` options.

Author(s)

Hossein Riazoshams, May 2014. Email: <riazihosein@gmail.com> URL `http://www.riazoshams.com/nlr/`

References

Riazoshams, H. (2012), Robustifying the Least Squares estimate of parameters of variance model function in nonlinear regression with heteroscedastic variance, Poster Presentation, Royal Statistical Society Conference (RSS) 2012, Telford, UK.

See Also

`dfr.hetro`, `nlr.control`, `fittmethod`, `nl.form`, `nl.fitt.rob`, `nl.fitt.rgn`, `nlr.control`, `nl.robhetroLS`, `dfr.hetroLS`

Examples

```
"dfr.robhetroLS"
```

`dfr.robloss` *Robut loss function.*

Description

Resturn robust loss function for minimization purpose to find the M-estimate. It is used in `dfrmest.NLM` function for derivative free purpose. Gradient and hessian are computed numerically.

Usage

```
dfr.robloss(formula, data, start, robfunc, control = nlr.control(), rmat = NULL, ..
```

Arguments

`formula`	`nl.form` object of nonlinear regression model.
`data`	list of data include responce and predictor.
`start`	list of parameter values of nonlinear model function (θ in $f(x, \theta)$), initial values or increament during optimization procedure. It must include scale sigma (standard deviation), if not included Fault(9) will be returned.
`robfunc`	`nl.form` of rho function. It must include tuning constants k0 and k1.
`control`	list of `nlr.control` for controling convergence criterions.
`rmat`	R-Matrix for transforming, it might be cholesky decomposition of covariance matrix.
`...`	any other arguments might be used in formula, robfunc or tuning constants in rho function.

Details

Compute Loss function, sum of robust rho function to compute the M-estimate.

$$\ell(\theta) = \sum \rho\left(\frac{r_i}{\sigma}\right)$$

Standard deviation σ must be included in `start` argument list with the name `sigma`.

`gradient` and `hessian` attributes compute numerically.

Value

result <- list(htheta=htheta,rho=robvalue,ri=rsd,fmod=fmod,Fault=Fault2) list of output:

`htheta:`	sum of rho function, include attribute `"gradient"` and `"hessian"`
`rho:`	computed rho function and attributes of `"gradient"` and `"hessian"`
`ri:`	residuals
`fmod:`	computed function contains esponse and or its gradient and hessian predictor and or its gradient & hessian
`Fault:`	`Fault` object of error, if no error Fault number = 0 will return back.

Note

All functions should have `gradient` and `hessian` in attributes. For derivative free purpose the `dfr.robloss` can be used. It is designed for internal use, might not call directly by user.

Author(s)

Hossein Riazoshams, May 2014. Email: <riazihosein@gmail.com> URL `http://www.riazoshams.com/nlr/`

References

Riazoshams H, Midi H, and Ghilagaber G, 2018,. Robust Nonlinear Regression, with Application using R, Joh Wiley and Sons.

See Also

`nl.form`, `nlr.control`, `nlmest.NLM`, `dfr.robloss`, `robloss`

Examples

```
## The function is currently defined as
"dfr.robloss"
```

---

dfrmest.NLM *Derivative free MM-estimate*

Description

MM-estimate of a nonlinear function, Using Mixture of Newton and Levenberg-Marquardt method. Parameters estimates by robust MM-estimate by minimizing the sum of robust rho function. Required derivatives such as `gradient` and `hessian` are computed numericaly by the loss function `dfr.robloss`

Usage

```
dfrmest.NLM(formula, data, start = getInitial(formula, data), robfunc, control =
nlr.control(tolerance = 0.01, minlanda = 1/2^25,maxiter = 25 * length(start)), vm =
rm = NULL, ...)
```

Arguments

`formula`	nl.form object of the nonlinear function model. See `nl.form` object.
`data`	list of data with the response and predictor as name of variable. In heterogeneous case if it include response variable values of heterogenous variance function it asume variance function is function of predictor $H(x_i, \tau)$, otherwise it assume is a function of predictor $H(f(x_i, \theta), \tau)$.

start	list of starting value parameter, name of parameters must be represented as names of variable in the list.
robfunc	nl.form object of robust function used for downgrading.
control	nlr.control object, include tolerance, maxiter,... see nlr.control.
vm	optional covariance matrix of residuals, used for nonlinear generalized M-estimate.
rm	optional correlation matrix, used for nonlinear generalized M-estimate. rm is correlation matrix of vm, thus only vm is enough to be given. It can be given by user also but not necessary automatically will be calculated by argument eiginv(t(chol(vm))).
...	any other argument passed to formula, robfnc, or optimization function.

Details

This function is mixture of Levenberg Marquardt, Newton and Steepest descent, but using numerical derivatives. It is used to minimize the robust loss function using ρ function.

Due to wrong effect of outlier in creating singularity in hessian matrix the levenberg Marquardt is used to remedy the effect. Morover for fast convergence when hessian is non singular Newton with Steepest descent is applied.

Value

result is object of nl.fitt.rob (nonlinear fitt robust) for homogeneous variance, and nl.fitt.rgn for heterogeneous and autocorrelated error (nonlinear fitt robust generalized), see nl.fitt.rgn object detail.

parameters	nonlinear regression parameter estimate of θ.
correlation	of fited model.
form	nl.form object of called nonlinear regression model.
response	computed response.
predictor	computed (right side of formula) at estimated parameter with gradient and hessian attributes.
curvature	list of curvatures, see curvature function.
history	matrix of convergence history, collumns include: convergence index, parameters, minimized objective function, convergence criterion values, or other values. These values will be used in plot function in ploting history.
method	fittmethod object of method used for fitt.
data	list of called data.
sourcefnc	Object of class "callorNULL" source function called for fitt.
Fault	Fault object of error, if no error Fault number = 0 will return back.
htheta	robust loss value including gradient and hessain attributes.
rho	computed robust rho function, including gradient and hessain attributes.
ri	estimated residuals, including gradient and hessain attributes.
curvrob	curvature
robform	nl.form object of robust loss rho function.

if vm is not NULL the nl.fitt.rgn include following extra slots:

vm	covariance matrix, diagonal of variance model predicted values.
rm	cholesky decomposition of vm.
gresponse	transformed of response by rm, include gradinet and hessian attributes.
gpredictor	transformed of predictor by rm, include gradinet and hessian attributes.

Note

starting values start it must contains initial value for 'sigma'. This function is called from nlr with derivfree=TRUE and method="MM" in control argument, it is more effient to be called from nlr not directly by user.

Author(s)

Hossein Riazoshams, Jan 2010. Email: <riazihosein@gmail.com> URL http://www.riazoshams.com/nlr/

References

Riazoshams H, Midi H, and Ghilagaber G, 2018,. Robust Nonlinear Regression, with Application using R, Joh Wiley and Sons.

See Also

dfr.robloss, nlmest.LM, nlmest.WF, nl.form, nl.fitt.rob, nl.fitt.rgn, nlr.control

Examples

```
datalist=list(xr=ntp$dm.k,yr=ntp$cm.k)
ntpstart=list(p1=.12,p2=6,p3=1,p4=33)
ntpstarttau=list(tau1=-.66,tau2=2,tau3=.04)
fittnml <- dfrmest.NLM(formula=nlrobj1[[16]], data = datalist, start=ntpstart,
robscale = TRUE, robfunc = nl.robfuncs[["huber"]],control=
nlr.control(tolerance=1e-4,trace=TRUE))
fittnml$parameters
## The function is currently defined as
"dfrmest.NLM"
```

DrugKenakin *Responses to the concentration of an agonist in a functional assay.*

Description

Kenakin used a set of responses to the concentration of an agonist in a functional assay.

Usage

```
data("DrugKenakin")
```

Format

The format is: data.frame chr "DrugKenakin"

- Concentration Concentration, predictor.
- Response response, response variable.

Details

Kenakin used a set of responses to the concentration of an agonist in a functional assay. They fit the following model to their data. In this data, observation 5 has an outlier in the response direction.

Source

Kenakin TP. A Pharmacology Primer: Theory, Applications, and Methods. Third Edition Academic Press; 2009. pp. 286-287.

References

Kenakin TP. A Pharmacology Primer: Theory, Applications, and Methods. Third Edition Academic Press; 2009. pp. 286-287.

Examples

```
data(DrugKenakin)
DrugKenakin
```

---

eiginv *Inverse of matrix using eigenvalues.*

Description

Compute the inverse of matrix using spectoral decomposition, using eigenvalues and eigen vectors of matrix.

Usage

```
eiginv(mtrx, stp = T, symmetric = all(mtrx == t(mtrx)))
```

Arguments

mtrx	square matrix to compute the inverse.
stp	if stp=T when error happened stop running program, if stp=F, does not stop program but return back Fault object.
symmetric	Used for computing eigenvalues, if symmetric=T the matrix is symetric, if symmetric=F the matrix is not symetric.

Details

`eiginv` function compute the inverse of matrix using spectoral decomposition

$$A_{k\times k} = \mathbf{P}\Lambda\mathbf{P}'$$

where

$$\mathbf{P} = [e_1, \ldots, e_k]$$

$$\Lambda = diag(\lambda_i)$$

in which λ_i is eigenvalues of matrix A coresponding to eigenvector e_i. Then the inverse is:

$$A^{-1} = \mathbf{P}\Lambda^{-1}\mathbf{P}'$$

Value

If matrix is positive definit, that is all eigenvalues are positive, return the inverse of matrix, if matrix is not positive definit returns `Fault` object with fault number=9, means the matrix is not positive definit.

Note

This function mostly used in optimization subroutindes, thus the inverse of negative definit matrix returned as fault. In contrast the `indifinv` function return back the same inverse eventhough the matrix is not positive definit.

Author(s)

Hossein Riazoshams, Jan 2010. Email: <riazihosein@gmail.com> URL `http://www.riazoshams.com/nlr/`

References

Riazoshams H, Midi H, and Ghilagaber G, 2018,. Robust Nonlinear Regression, with Application using R, Joh Wiley and Sons.

See Also

`indifinv`

Examples

```
 a1=matrix(c(1,2,3,4,5,6,7,8,9),nrow=3)
eiginv(a1)
```

evald *eval function.*

Description

Implemented form of eval function with dots argument.

Usage

```
evald(expr, envir = parent.frame(),
enclos = if (is.list(envir) || is.pairlist(envir)) parent.frame() else baseenv(), .
```

Arguments

expr	an object to be evaluated.
envir	the environment in which expr is to be evaluated. May also be NULL, a list, a data frame, a pairlist or an integer as specified to sys.call.
enclos	Relevant when envir is a (pair)list or a data frame. Specifies the enclosure, i.e., where R looks for objects not found in envir. This can be NULL (interpreted as the base package environment, baseenv()) or an environment.
...	other arguments pass to expr.

Details

this function works exactly similar to eval except accept ... argument, it is a compatibility adjustment to SPLUS.

Value

The result of evaluating the object: for an expression vector this is the result of evaluating the last element.

Note

This is a specialisd for nlr package to imitate eval function for special uses, is not intended to be used by user.

Author(s)

Hossein Riazoshams, Apr 2014. Email: <riazihosein@gmail.com> URL http://www.riazoshams.com/nlr/

References

Riazoshams H, Midi H, and Ghilagaber G, 2018,. Robust Nonlinear Regression, with Application using R, Joh Wiley and Sons.

See Also

eval, nl.form

Examples

```
## The function is currently defined as
"evald"
```

Fault *Class* "Fault"

Description

Error object returns from all functions in nlr. It may return no fault or warning or error message.

Objects from the Class

Objects can be created by calls of the form new("Fault",FL = F,FN=0,FT=NULL,FF=NULL,pnt=F).

if pnt=T then it display the created object result.

Slots

FL: Object of class "logical" Fault Logic if is TRUE error occured otherwise no error occured.

FN: Object of class "numeric" Fault Number, code for error, see notes bellow. All codes can be seen in db.Fault

FT: Object of class "character" Fault Text, explain the error.

FF: Object of class "character" Fault File, the function that error raised in.

methods

$ signature(x = "Fault"): return slots.

Note

db.Fault variable include error codes and expression used in Fault.

Author(s)

Hossein Riazoshams, May 2014. Email: <riazihosein@gmail.com> URL http://www.riazoshams.com/nlr/

References

Riazoshams H, Midi H, and Ghilagaber G, 2018,. Robust Nonlinear Regression, with Application using R, Joh Wiley and Sons.

See Also

db.Fault

Examples

```
showClass("Fault")
db.Fault        # Error codes
Fault(FN=1)
```

---

fittmethod *Class* "fittmethod"

Description

Fitt method object, it store the method used for estimation, runing function.

Objects from the Class

Objects can be created by calls of the form new("fittmethod",...).

Slots

methodID: Object of class "numeric" code for the method.

method: Object of class "character" name of the method used in estimation and other computation procedures.

detail: Object of class "character" detail text description of the method.

methodBR: Object of class "numeric" (method Branche) branch for the method used in iteration.

detailBR: Object of class "character" detail of the branch.

subroutine: Object of class "character" the function, subroutine, that the result constructed from.

lossfunction: Object of class "character" objective loss function used by subroutine to optimize.

subroutineBR: Object of class "character" sub subroutine, called for optimize.

Extends

Class "fittmethodorNULL", directly.

Methods

$ signature(x = "fittmethod"): access slots.

Note

This is for internal use, might not be called directly by user. All the fited objects, `nl.fitt`, `nl.fitt.gn`, `nl.fitt.rgn`, `nl.fitt.rob`, have a method slot of object type `fittmethod`. It can be used by `recalc` method to recalculate the fitt.

Author(s)

Hossein Riazoshams, May 2014. Email: <riazihosein@gmail.com> URL `http://www.riazoshams.com/nlr/`

References

Riazoshams H, Midi H, and Ghilagaber G, 2018,. Robust Nonlinear Regression, with Application using R, Joh Wiley and Sons.

See Also

`nl.fitt`, `nl.fitt.gn`, `nl.fitt.rgn`, `nl.fitt.rob`

Examples

```
fittmethod(methodID=1)
```

`fullqr` *full rank QR decomposition.*

Description

Is actual QR decomposition matrix, created for compatibility to `nlr` package functions.

Usage

```
fullqr(x)
```

Arguments

`x`	A Square matrix.

Details

Compute QR matrices, and provide explicit matrix form of lower triangular and upper triangular matrices.

Value

List of standard matrix form to be used in internal functions in nlr.

- qq matrix.
- rr matrix.
- q2q2 is from partition of Q=[q1|q2] matrix.
- r1 r1 is from partition of R=[R1/0] matrix.
- rinvis generalized inverse of r1.
- arQR decomposition result from qr function.

Note

Used mostly for internal purposes.

Author(s)

Hossein Riazoshams, Jan 2010. Email: <riazihosein@gmail.com> URL http://www.riazoshams.com/nlr/

References

Riazoshams H, Midi H, and Ghilagaber G, 2018,. Robust Nonlinear Regression, with Application using R, Joh Wiley and Sons.

See Also

nlsqr

Examples

```
a1=matrix(c(1,2,3,4,5,6,7,8,9),nrow=3)
fullqr(a1)
```

I *Inhibit Interpretation/conversion of object*

Description

This is AsIs function for compatibility purpose. See AsIs help.

Usage

```
I(x)
```

Arguments

x	an object

Details

Function has two main usage. See `AsIs` help.

Value

A copy of object with class "AsIs" prepared to the class(es).

Note

This is I, and AsIs function in base package, inserted here for compatibility to change of package versions.

Author(s)

Chambers, J. M. (1992)

References

Chambers, J. M. (1992) Linear models. Chapter 4 of statistical models in S eds J. M. Chamber and T. J. Hastie, Wadsworth & Brooks/Cole.

See Also

`AsIs`

Examples

```
## The function is currently defined as
"I"
```

`indifinv` *Indefinite Inverse of matrix.*

Description

Compute the inverse of matrix using spectoral decomposition, using eigenvalues and eigen vectors of matrix.

Usage

```
indifinv(mtrx, stp = T, symmetric = all(mtrx == t(mtrx)))
```

Arguments

mtrx	square matrix to compute the inverse.
stp	if stp=T when error happened stop running program, if stp=F, does not stop program but return back Fault object.
symmetric	symmetricUsed for computing eigenvalues, if symmetric=T the matrix is symetric, if symmetric=F the matrix is not symetric.

Details

eiginv function compute the inverse of matrix using spectoral decomposition

$$A_{k\times k} = \mathbf{P}\Lambda\mathbf{P}'$$

where

$$\mathbf{P} = [e_1, \ldots, e_k]$$
$$\Lambda = diag(\lambda_i)$$

in which λ_i is eigenvalues of matrix A coresponding to eigenvector e_i. Then the inverse is:

$$A^{-1} = \mathbf{P}\Lambda^{-1}\mathbf{P}'$$

Value

If matrix eigenvalues are not zero return the inverse of matrix, otherwise returns Fault object with fault number=9, means the matrix is not positive definit.

Note

eiginv return back error if negative eigen values occured, means singular matrix. But indifinv raise error if infinite or null values occured. The eigenvalues decomposition is used as eiginv.

Author(s)

Hossein Riazoshams, Jan 2010. Email: <riazihosein@gmail.com> URL http://www.riazoshams.com/nlr/

References

Riazoshams H, Midi H, and Ghilagaber G, 2018,. Robust Nonlinear Regression, with Application using R, Joh Wiley and Sons.

See Also

eiginv

Examples

```
a1=matrix(c(1,2,3,4,5,6,7,8,9),nrow=3)
indifinv(a1)
```

`individ` *Split individuals*

Description

Hessian of a function is symetric matrix, `individ` function constract lower diagonal parts and store them in collumns of a matrix. Thus the only unique hessians will be given.

Usage

```
individ(hessian)
```

Arguments

`hessian`	Three dimentional $n \times p \times p$ array of hessian.

Details

It is used in `curvature` function to compute the curvatures.

Value

$n \times (p * p + 1)$ matrix of hessian values, constracted from the lower triangular of hessian.

Note

It is created for internal use in `curvature` function, it ight not be called by user explicitly.

Author(s)

Hossein Riazoshams, May 2014. Email: <riazihosein@gmail.com> URL `http://www.riazoshams.com/nlr/`

References

Bates, D.M., and Watts, D. G. (1980). Relative curvature measures of nonlinearity, J. R. statistic. Ser. B 42: 1-25.

See Also

`curvature`

Examples

```
## The function is currently defined as
"individ"
```

`is.Fault` *Check error*

Description

Error of a procedure in `nlr` package will be saved in `Fault` slot of fited object with object type class `Fault`. `is.Fault` read the `Fault` slot or a `Fault` object to detect error happened or no. The value inside `Fault$FL` will show the result.

Usage

```
is.Fault(obj)
```

Arguments

`obj`	all fited objects include a slot of `Fault` object to represent the error. `obj` can be fitt objects: `nl.fitt`, `nl.fitt.gn`, `nl.fitt.rob`, `nl.fitt.rgn`, `Fault`.

Details

`is.Fault` is imitating other `is.` primitive functions.It is implemented for debuging purpose in `nlr`.

Value

logical value `TRUE` or `T` if error happened, `FALSE` or `F` if result is not error.

Note

The `Fault` object can be warning, but not error, in this case you can use `is.Warn` or `is.Faultwarn` to detect warnings.

Author(s)

Hossein Riazoshams, May 2014. Email: <riazihosein@gmail.com> URL `http://www.riazoshams.com/nlr/`

References

Riazoshams H, Midi H, and Ghilagaber G, 2018,. Robust Nonlinear Regression, with Application using R, Joh Wiley and Sons.

See Also

`Fault`, `is.Warn` or `is.Faultwarn`, `nl.fitt`, `nl.fitt.gn`, `nl.fitt.rob`, `nl.fitt.rgn`

Examples

```
is.Fault(1)
is.Fault(Fault(FN=1))
```

---

is.Faultwarn *Check error or warning*

Description

Return TRUE if error oe warning occured. Error or warning of a procedure in `nlr` package will be saved in `Fault` slot of fited object with object type class `Fault`. `is.Faultwarn` read the `Fault` slot or a `Fault` object to detect error happened or no. The value inside `Fault$FN` will display the error code. If is not zero means an error or warning raised.

Usage

```
is.Faultwarn(obj)
```

Arguments

`obj`	all fited objects include a slot of `Fault` object to represent the error. `obj` can be fitt objects: `nl.fitt`, `nl.fitt.gn`, `nl.fitt.rob`, `nl.fitt.rgn`, `Fault`.

Details

`is.Faultwarn` is imitating other `is.` primitive functions.It is implemented for debuging purpose in `nlr`.

Value

logical value `TRUE` or `T` if error or warning occured, `FALSE` or `F` if result is not error nor warning. Technicaly if The slot `FN` of `Fault` object if is not zero means error or warning raised, in that case other slots represent the error or warning informations.

Note

The `Fault` object can be error or warning, `is.Warn` check for warning only and `is.Fault` check the error.

Author(s)

Hossein Riazoshams, May 2014. Email: <riazihosein@gmail.com> URL `http://www.riazoshams.com/nlr/`

References

Riazoshams H, Midi H, and Ghilagaber G, 2018,. Robust Nonlinear Regression, with Application using R, Joh Wiley and Sons.

See Also

Fault, is.Warn or is.Fault, nl.fitt, nl.fitt.gn, nl.fitt.rob, nl.fitt.rgn

Examples

```
is.Faultwarn(Fault(FN=1))
```

---

is.inf *Check IEEE Arithmetic Values*

Description

Returns a logical vector or matrix describing the type of numeric elements present. This distinguishes between infinite values, NaN's, missing values and ordinary numbers.

Usage

```
is.inf(x)
```

Arguments

x	numeric vector of values. Check performs for all values.

Details

These functions are created for compatibility with SPLUS.

Value

Returns an object similar to the input which is filled logical values. Values will be false for vectors that are not of mode "numeric".

is.infinite is TRUE for values of x that are either plus or minus infinity.

is.inf is an abbreviation for is.infinite

Note

In nlr package this function use in eval method of nl.form to identify the computed response and predictor is right.

Author(s)

Hossein Riazoshams, 2013. Email: <riazihosein@gmail.com> URL http://www.riazoshams.com/nlr/

References

Riazoshams H, Midi H, and Ghilagaber G, 2018,. Robust Nonlinear Regression, with Application using R, Joh Wiley and Sons.

See Also

nl.form

Examples

```
aaray=c(0,3/0)
is.inf(aaray)
is.missing(aaray)
```

---

is.Warn *Check warning*

Description

Return TRUE if warning occured. Error or warning of a procedure in nlr package will be saved in Fault slot of fited object with object type class Fault. is.Warn read the Fault slot or a Fault object to detect warning happened or no. The value inside Fault$FN will display the error code and Fault$FL represent error, both together can be used to identify warning.

Usage

```
is.Warn(obj)
```

Arguments

obj	all fited objects include a slot of Fault object to represent the error. obj can be fitt objects: nl.fitt, nl.fitt.gn, nl.fitt.rob, nl.fitt.rgn, Fault.

Details

is.Warn is imitating other is. primitive functions.It is implemented for debuging purpose in nlr. Warning might happen for example when maximum number of iteration exceeded, in this case the result might not be reliable and options should be changed. Another example might be when an infinit, null or rmissing value happened, in some cases not any result might be returned. For example if a derivative of a nonlinear function does not exist, the gradient values might have null values then the Nelder-Mead derivative free should be used.

Value

logical value TRUE or T if warning occured, FALSE or F if result is not warning. Technicaly if The slot FN of Fault object if is not zero means error or warning raised and FL is true if error occured. Other slots represent the error or warning informations.

Note

The Fault object can be error or warning, is.Faultwarn check for error or warning and is.Fault check the error.

Author(s)

Hossein Riazoshams, May 2014. Email: <riazihosein@gmail.com> URL http://www.riazoshams.com/nlr/

References

Riazoshams H, Midi H, and Ghilagaber G, 2018,. Robust Nonlinear Regression, with Application using R, Joh Wiley and Sons.

See Also

Fault, is.Faultwarn or is.Fault, nl.fitt, nl.fitt.gn, nl.fitt.rob, nl.fitt.rgn

Examples

```
is.Warn(Fault(FN=2))    # Not warning
is.Warn(Fault(FN=3))    # warning
```

---

jaclev *Jacobian Leverage for nonlinear regression.*

Description

Compute the Jacobian Leverage, generalized for nonlinear case.

Usage

```
jaclev(gradient, hessian, rsd)
```

Arguments

gradient	$n \times p$ gradient of nonlinear function.
hessian	three simentional $n \times p \times p$ of hessian of nonlinear regression function.
rsd	$n \times 1$ residual vector.

Details

Jacobian leverage, generalized form of hat matrix for nonlinear regression.

Value

$n \times n$ matrix of jacobian leverages.

Note

Jacobian leverage for nonlinear regression is direct definition of perturbing response, thus it is free from the problems due to linear approximation of nonlinear function.

Author(s)

Laurent. R. T. ST., and Cook.

References

Laurent. R. T. ST., and Cook. R. D. (1992). Leverage and Superleverage in Nonlinear Regression, Journal of the American Statistical Association 87(420): 985-990.

See Also

`nl.fitt`, `nl.fitt.gn`

Examples

```
## The function is currently defined as
"jaclev"
```

Lakes *Lakes Data*

Description

Lakes data represent a multivariate predictor nonlinear model.

Usage

```
data(Lakes)
```

Format

The format is:data.frame chr "Lakes"

- tn: mean annual total nitrogen concentration, response variable.
- nin: average influence nitrogen concentration, predictor variable one.
- tw: water retention time, predictor variable two.

Details

Lakes data is collected from 29 lakes in Florida by United States Environmental Protection Agency (1978). Stromberg (1993) has identified observations 10 and 23 as outliers. The data presents the relationship between the mean annual total nitrogen concentration, TN, as the response variable and the average influence nitrogen concentration, NIN, and water retention time, TW, as predictors.

Source

United Stated Environmental Protection Agency (1978), "A Conpendium of Lake and Reservoir data Collected by the National Eutrophication Survey in Eastern, North Centeral and Southeastern United States," Working Paper #475, Corvallis Environmental Research Labratory, Corvallis, Oregon.

References

Stromberg, A. J. (1993). Computation of High Breakdown Nonlinear Regression Parameters, Journal of American Statistical Association 88(421): 237-244.

Examples

```
data(Lakes)
Lakes
```

`loss.chis` *Hetroscedastic chi-square loss function.*

Description

Compute the minus of chi-square pseoudo log likelihood,based on varaince model function. $\ell(\sigma^2, \lambda) = \sum\{w_i \log(\tilde{H}(x_i; \sigma^2, \lambda)) + z_i/\tilde{H}(x_i; \sigma^2, \lambda)\}$

Usage

```
loss.chis(formula, data, start, theta, varmodel, ...)
```

Arguments

`formula`	`nl.form` object of nonlinear function $f(x; \theta)$.
`data`	list of data include responce, predictor or possibly predictor of variance model function (t), if not represented then the predict of nonlinear model function will be replaced in predictor variable of nonlinear variance model function that is $Var(\varepsilon) = \sigma^2 H(f(\theta), \tau)$
`start`	list of parameter values of variance model function (τ in $H(t, \tau)$), initial value or increament during optimization procedure.
`theta`	list of model function parameter (θ in $f(x, \theta)$).
`varmodel`	`nl.form` object of heteroscedastic variance model function. `varmodel` must return variance function not standard deviation. Include gradient and hessian.
`...`	extra argument might pass to nonlinear regression or heteroscedastic functions.

Details

For estimating variance model parameter τ, chi-square pseudo chi square is used as classic estimate. Based on calculating the sample variances.

Value

list od los function values:

`value`	value of minus loglikelihood of chi-square, include attribute "gradient"" and "hessian". These values use in optimization functions.
`angvec`	angular vector for checking the convergence.
`angmat`	angular matrix for checking convergence in optimization procedure.
`refvar`	refvar, sample variance $\frac{\sum(wi*z)}{\sum(wi)}$
`fmod`	computed function model $f(x, \theta)$, include response, predictor and their gradient and hessian depends on the defined form of nonlinear function model.
`varcomp`	computed variance function model $H(t, \tau)$, include response or predictor and their gradient and hessian depends on the defined form of nonlinear function model.
`vcmdata`	list of data used in variance model function, that is varmodel$independent and varmodel$dependent typically is zi.
`sourcefnc`	source function from which this function is called. May be used in feature computing such as outlier detection measures.
`zi`	computed sample variance, which follows the chi-square distribution.

Note

This is used for classic estimates, for robust estimates see `loss.robchis` This is implemented for internal use, might not be called directly by user.

Author(s)

Bunke, O., Droge, B., Polzehl

References

Bunke, O., Droge, B., Polzehl, J. Splus tools for model selection in nonlinear regression (1998) Computational Statistics, 13 (2), pp. 257-281.

See Also

`loss.robchis`, `nlr`

Examples

```
## The function is currently defined as
"loss.chis"
```

`loss.hetroWM` *Weighted Robut loss function.*

Description

weighted loss function is used to estimate Weighted M-estimate. It is a robustified form of Likelihood function for heteroscedastic variance case. `loss.hetroWM` will be used in optimization function for estimating parameters of nonlinear function model and variance function model simultaneously.

Usage

```
loss.hetroWM(formula, data, start, varmodel, robfunc, ...)
```

Arguments

`formula`	`nl.form` object of nonlinear regression model.
`data`	list of data include responce, predictor or possibly predictor of variance model function (t), if not represented then the predict of nonlinear model function will be replaced in predictor variable of nonlinear variance model function that is $Var(\varepsilon) = \sigma^2 H(f(\theta), \tau)$
`start`	list of parameter values of nonlinear model function (θ in $f(x, \theta)$ and τ in $H(x, \tau)$), initial values or increament during optimization procedure.
`varmodel`	`nl.form` object of heteroscedastic variance model function. `varmodel` must return variance function not standard deviation. Include gradient and hessian.
`robfunc`	`nl.form` of rho function. It must include tuning constants k0 and k1.
`...`	extra argument might pass to nonlinear regression, heteroscedastic functions, or tuning constant and other parameters to robust loss rho function.

Details

Loss function in general form have robust rho function. l(tau,theta)=sum[log(h(mu;tau,sg))] + sum[rho(ri/h(mu;tau,sg))]

Value

result <- list (value=value,angvec=angvec,angmat=angmat, refvar=refvar,sourcefnc= match.call(), rho=robvalue,fmod=fmod,varcomp=varcomp,correlation =nlrho,ri=ri)

`value`	Robustified form of log-likelihood function, use in optimize function. Include gradient and hessian attributes.
`angvec`	angular vector for checking the convergence.
`angmat`	angular matrix for checking convergence in optimization procedure.
`refvar`	refvar, sample variance $\frac{\sum(wi*z)}{\sum(wi)}$

sourcefnc	Object of class "callorNULL" source function called for fitt.
rho	computed robust loss rho function. Include hessian and gradients.
fmod	computed function contains esponse and or its gradient and hessian predictor and or its gradient & hessian
varcomp	computed variance function model $H(t, \tau)$, include response or predictor and their gradient and hessian depends on the defined form of nonlinear function model.
correlation	correlation of fitt.
ri	residuals

Note

rho function can be square function, this will produce non robust MLE or LS. This is implemented for internal use, might not be called directly by user.

Author(s)

Lim, C., Sen, P. K., Peddada, S. D.

References

Lim, C., Sen, P. K., Peddada, S. D. (2010). Statistical inference in nonlinear regression under heteroscedasticity. Sankhya B 72:202-218.

See Also

nl.form

Examples

```
## The function is currently defined as
"loss.hetroWM"
```

---

loss.robchis *Hetroscedastic chi-square robust loss function.*

Description

Compute the robustified chi-square pseoudo log likelihood,based on varaince model function.

$$\ell(\boldsymbol{\theta}, \tau) = \sum \{ w_i \log(H(x_i; \tau) + \rho \left[\sqrt{z_i / H(x_i; \sigma^2, \lambda)} \right] \}$$

Usage

```
loss.robchis(formula, data, start, theta, varmodel, robfunc, ...)
```

Arguments

formula	nl.form object of nonlinear function $f(x;\theta)$.
data	list of data include responce, predictor or possibly predictor of variance model function (t), if not represented then the predict of nonlinear model function will be replaced in predictor variable of nonlinear variance model function that is $Var(\varepsilon) = \sigma^2 H(f(\theta), \tau)$
start	list of parameter values of variance model function (τ in $H(t,\tau)$), initial value or increament during optimization procedure.
theta	list of model function parameter (θ in $f(x,\theta)$).
varmodel	nl.form object of heteroscedastic variance model function. varmodel must return variance function not standard deviation. Include gradient and hessian.
robfunc	nl.form of rho function. It must include tuning constants k0 and k1.
...	extra argument might pass to nonlinear regression, heteroscedastic functions, or tuning constant and other parameters to robust loss rho function.

Details

For estimating variance model parameter τ, robustified form of chi-square pseudo likelihood is used as robust estimate. Based on calculating the sample variances.

Value

list od los function values:

value	value of minus loglikelihood of chi-square, include attribute "gradient"" and "hessian". These values use in optimization functions.
angvec	angular vector for checking the convergence.
angmat	angular matrix for checking convergence in optimization procedure.
refvar	refvar, sample variance $\frac{\sum(wi*z)}{\sum(wi)}$
fmod	computed function model $f(x,\theta)$, include response, predictor and their gradient and hessian depends on the defined form of nonlinear function model.
varcomp	computed variance function model $H(t,\tau)$, include response or predictor and their gradient and hessian depends on the defined form of nonlinear function model.
vcmdata	list of data used in variance model function, that is varmodel$independent and varmodel$dependent typically is zi.
sourcefnc	source function from which this function is called. May be used in feature computing such as outlier detection measures.
rho	computed robust loss rho function. Include hessian and gradients.
zi	computed sample variance, which follows the chi-square distribution.

Note

This is used for robust estimates, for classic see loss.chis This is implemented for internal use, might not be called directly by user.

Author(s)

Hossein Riazoshams, 08/01/2010. Email: <riazihosein@gmail.com> URL http://www.riazoshams.com/nlr/

References

Riazoshams H 2010 Outlier detection and robust estimation methods for nonlinear regression having autocorrelated and heteroscedastic errors.

See Also

nl.form, loss.chis

Examples

```
## The function is currently defined as
"loss.robchis"
```

---

loss.SSQ *Sum of squared loss function.*

Description

This function used in nlsnm function to compute the least square estimate using derivative free Nelder-Mead algorithm.

Usage

```
loss.SSQ(formula, data, start, vm = NULL, rm = NULL, ...)
```

Arguments

formula	nl.form object of nonlinear regression model.
data	list of data include responce and predictor.
start	list of parameter values of nonlinear model function (θ in $f(x, \theta)$), initial values or increament during optimization procedure.
vm	optional covariance matrix.
rm	optional cholesky decomposition of covariance matrix.
...	any other arguments might be used in formula, robfunc or tuning constants in rho function.

Details

loss.SSQ compute the sum of square of residuals, it is optimized to be used in nlsnm function, since optimization method Nelder-Mead is derivative free the result does not include derivatives.

Value

result <- list(value = value,correlation=correlation,fmod=fmod)

list values:

`value`	sum of squared residuals.
`correlation`	correlation of model
`fmod`	computed function (transformed by R) contains esponse and or its gradient and hessian predictor and or its gradient & hessian, transformed also by R.

Note

If required to compute square loss function include can use `nl.robfuncs[7]`, see `nl.robfuncs`. This is implemented for internal use, might not be called directly by user.

Author(s)

Hossein Riazoshams, May 2014. Email: <riazihosein@gmail.com> URL `http://www.riazoshams.com/nlr/`

References

Robust Nonlinear Regression, Theories and Methods with Practical Guides for R Packages. Riazoshams et al.

See Also

`nlsnm`, `nl.robfuncs`

Examples

```
## The function is currently defined as
"loss.SSQ"
```

---

lotsout *Artificially Contaminated Data from Logistic model.*

Description

lotsout is artificially Contaminated Data from Logistic model, using the computed parameter values from the estimates of fited chicken growth `Weights` data.

Usage

```
data("lotsout")
```

Format

The format is: chr "lotsout"

- lotsout[,1]: predictors repeated two times.
- lotsout[,2]: response simulated from the parameter estimates and logistic model.

Details

The predictor is similar to predictor of chicken growth `Weights` data but only repeated two times. The response is simulated from logistic model using the parameters from estimated fit for chicken growth data. This selection is used to mimic a real data example to have more natural behaviour.

Author(s)

Hossein Riazoshams, Jan 2010. Email: `<riazihosein@gmail.com>` URL `http://www.riazoshams.com/nlr/`

Source

Robust Nonlinear Regression, with Application Using R, Riazoshams et al, Wiley Inc.

References

Riazoshams H, Midi H, and Ghilagaber G, 2018,. Robust Nonlinear Regression, with Application using R, Joh Wiley and Sons.

Examples

```
lotsout
```

`methane` *Methane data.*

Description

Methane traped in iceberg during history.

Usage

```
data(methane)
```

Format

The format is: data.frame chr "methane"

- year: year of gas traped in iceberg.
- co2: measured Carbon Dioxide.

Details

UNEP (1989) presented the Methane Gas and Carbon Dioxide Gas collected from the Gas trapped in icebergs in south pole from 8000 years ago.

Source

UNEP (1989), Environmental data report / prepared for UNEP by the GEMS Monitoring and Assessment Research Centre, London, UK, in co-operation with the World Resources Institute, Washington, D.C.

References

Riazoshams, H., Midi, H., (2013) Application of Robust Nonlinear Regression, case study for modeling the greenhouse gases, Methane and Carbon Dioxide concentration in atmosphere. International Conference on Mathematical Science and Statistics (ICMSS 2013), Kula Lumpur, Malaysia.

Examples

```
data(methane)
methane
```

---

mplot *Multiple Plot*

Description

plot multiple models stored in `nl.fitt` and its child objects, in same graph.

Usage

```
mplot(mlist, case = 1, length.out = NULL, ...)
```

Arguments

`mlist`	list of object models
`case`	case=1, common x, case=2, different x
`length.out`	length of predictor to be increamented, if not given the original predictor data will be used.
`...`	extra option submit to plot.

Details

If different methods are used to estimate a model this function can be used to plot them all in same graph. Multiple output of estimates in `nlr` package should be stored in a list, then `mplot` plot data and predicted values for all methods over the data.

Value

Plot graph.

Note

A more common situation is when case=1 then common x-axis will be considered for all fited objects. That is the case when sommon data used with different method of fitt and once want to compare different methods.

Author(s)

Hossein Riazoshams, May 2014. Email: <riazihosein@gmail.com> URL http://www.riazoshams.com/nlr/

References

Rizo ML 2008 Statistical Computing with R The R Series. Chapman & Hall/CRC The R Series.

See Also

nl.fitt

Examples

```
datalist<-list(xr=trade.ir[,1],yr=trade.ir[,2])
a1<- nlr(nlrobj5[[18]],data=datalist,
control=nlr.control(trace=TRUE,derivfree = FALSE,tolerance=1e-4,singularCase=1,maxiter = 800
a2<- nlr(nlrobj5[[18]],data=datalist,
control=nlr.control(trace=TRUE,method="OLS",
derivfree = FALSE,tolerance=1e-4,maxiter = 4000))    # nelder mead, MM, selfstart

mlist=list(a1,a2)
mplot(mlist)
```

---

mscale *Scale M-estimate*

Description

Scale M-estimator with 50% breakdown

Usage

```
mscale(u)
```

Arguments

u	vector of values, in nonlinear regression residuals are used.

Details

Robust M-estimate of scale using rodust loss function rho, to acheive 50% breakdown. Tunning constant k1 should be stored in `nl.form` object of the robust function.

This is called from `nlmest.NLM` function to derive MM.estimate.

Value

Minimized Sum of rho function, by reweighting.

Note

General form is developed in `nlr` package to work for general robust function.

Author(s)

Stromberg (1993)

References

Yohai (1987) Annals, Stromberg (1993) JASA. GKS 2 June 99

See Also

`nl.form`, `nlmest.NLM`

Examples

```
x=c(2,3,10)
mscale(x)
```

---

net.ch *China Net Money Data*

Description

Net domestic credit (current LCU) china.

Usage

```
net.ch
```

Format

The format is: data.frame chr "net.ch"

- year: year
- net: net money amount.

Details

Net domestic credit (current LCU) of China.

Source

http://www.worldbank.org/

References

worldbank.com

Examples

```
data(net.ch)          # load data
net.ch                # access by variable name
```

---

net.ir *Iran Net Money Data.*

Description

Net domestic credit (current LCU) Iran.

Usage

```
net.ir
```

Format

The format is: data.frame chr "net.ir"

- year: year
- net: net money amount.

Details

Net domestic credit (current LCU) of Iran, from 1961.

Source

http://www.worldbank.org/

References

worldbank.com

Examples

```
data(net.ir)        # load data
net.ir              # access by variable name
```

net.kw *Kuwait Net Money Data.*

Description

Net domestic credit (current LCU) Kuwait.

Usage

```
net.kw
```

Format

The format is: data.frame chr "net.kw"

- year: year
- net: net money amount.

Details

Net domestic credit (current LCU) of Kuwait.

Source

http://www.worldbank.org/

References

worldbank.com

Examples

```
data(net.kw)
net.kw
```

net.sw *Sweden Net Money Data.*

Description

Net domestic credit (current LCU) Sweden.

Usage

```
net.sw
```

Format

The format is: data.frame chr "net.sw"

- year: year
- net: net money amount.

Details

Net domestic credit (current LCU) of Sweden.

Source

`http://www.worldbank.org/` obtained

References

worldbank.com

Examples

```
data(net.sw)
net.sw
```

---

nl.corrts *Autocorrelated two stage estimate*

Description

Two stage estimate for nonlinear regreession model with autocorrelated error.

Usage

```
nl.corrts(formula, data, start = getInitial(formula, data),
control = nlr.control(tolerance = 0.001, minlanda = 1/2^10,
maxiter = 25 * length(start)), correlation = NULL, ...)
```

Arguments

`formula`	nl.form object of the nonlinear function model. See `nl.form` object.
`data`	list of data with the response and predictor as name of variable.
`start`	list of starting value parameter, name of parameters must be represented as names of variable in the list.
`control`	nlr.control object, include tolerance, maxiter,... see `nlr.control`.
`correlation`	correlation structure, at the moment parameter of AR(p) process.
`...`	any argument pass to `formula`

Details

In first stage nonlinear regression parameter estimate and in second stage autocorrelation structure estimate and finally the generalized least square estimates the function model parameters.

Value

`fited`	`nl.fitt.gn` object generated by `nlsqr.gn` function.
`tm`	fitted time series model for residuals.

Note

This function currently run with AR process. The robust estimate is don by `nl.robcorrts` function. This function called from `nlr` function, since the correlation parameters have to be estimated it is more efficient to be called from `nlr` rather than directly by user.

Author(s)

Hossein Riazoshams, Jul 2009. Email: <riazihosein@gmail.com> URL http://www.riazoshams.com/nlr/

References

Riazoshams, H., Midi, H., Sharipov, O. S.H, (2010). The Performance of Robust Two Stage Estimator in Nonlinear Regression with autocorrelated Error, Communications in Statistics - Simulation and Computation, 39: 1251-1268.

See Also

`nl.robcorrts`, `nlsqr.gn`, `nl.fitt.gn`, `nlr.control`

Examples

```
library(nlme)
xr = tadr.ir[, 1]
yr = tadr.ir[, 2]
a1 <-  nl.corrts( nlrobj5[[18]],data=list(xr = xr, yr = yr),correlation=corAR1(0.8))
a1$parameters
```

---

nl.fitt-class *Class* "nl.fitt"

Description

Object for a fitted nonlinear regression model.

Objects from the Class

Objects can be created by calls of the form `new("nl.fitt",...)`.

Slots

`parameters`: Object of class `"list"`, estimate of nonlinear model θ.

`scale`: Object of class `"numericorNULL"`, standard deviation scale estimate σ.

`correlation`: Object of class `"numericorNULL"`, correlation structure of error.

`form`: Object of class `"nl.form"` of nonlinear model.

`response`: Object of class `"vectororMatrix"` response, left side of formula.

`predictor`: Object of class `"vectororMatrix"`, estimated predictor $\eta(\hat{\theta})$.

`curvature`: Object of class `"listorNULL"` of PE and IE curvatures.

`history`: Object of class `"matrixororNULL"` convergence computations in iteration procedures, include parameters, objective function and other parameters depends on the method.

`method`: Object of class `"fittmethodorNULL"` method of iteration used, contains main method, functions and sub methods. See `fittmethod`.

`data`: Object of class `"list"` data used in computation, including response and predictor variables.

`sourcefnc`: Object of class `"callorNULL"` source function called for fitt.

`Fault`: Object of class `"Fault"` of error or warnings if happened.

`others`: Object of class `"listorNULL"` of other computations, as an example the object of outlier detection measures will be saved in this slot later on.

Extends

Class `"nl.fittorNULL"`, directly.

Methods

$ `signature(x = "nl.fitt")`: access the slot values. Usage: `objectname$slotname`

atypicals `signature(nlfited = "nl.fitt")`: detect atypical points by calculating outlier detection measures. Usage: `atypicals(nlfited)`

hat `signature(x = "nl.fitt")`: generalized Hat matrix from linear regression to nonlinear regression using gradient. Usage: `hat(x="nl.fitt")`

JacobianLeverage `signature(nlfited = "nl.fitt")`: Jacobian-Leverage for nonlinear regression. Usage `JacobianLeverage(nlfited = "nl.fitt")`

parInfer `signature(object = "nl.fitt")`: parameter inference function, calculate covariance matrix of parameters and their confidence interval using gradient as design matrix. Usage: `parInfer(object,confidence = .95)`

plot `signature(x = "nl.fitt",y = "missing",control=nlr.control(),...)`: generic function extended to `nl.fitt` object. Plot the object. Usage. `plot(x,y="missing",control=nlr.control`
If history is TRUE the convergence of fitt will be ploted. length.out is length of incremented p[redictor to acheive smooter curve. singlePlot=F plot the model and residuals in two collumn.

predict `signature(object = "nl.fitt")`: generic function, predict nonlinear function model at estimated parameter values. Usage: `predict(object,...)`, dots argument can include the `newdata` which might be new list of new values for predictor variables, if not given the original data that used for fitt will be used to calculate prediction values.

predictionI `signature(nlfited = "nl.fitt.gn")`: prediction interval. Usage: `predictionI(nlfited,con` data is new data that will be predicting the values for them.

recalc `signature(object = "nl.fitt")`: generic function, recalculate the object with new arguments given in dots argument.

residuals `signature(object = "nl.fitt")`: residuals of fitt. Usage: `residuals((object,...))`, dots argument can include `data` list of predictor and response variables, if `data` is not given the residuals will calculate for original data used in estimation.

acf The function acf computes (and by default plots) estimates of the autocovariance or autocorrelation function of residuals. For argument details see stats `acf` general function.

Note

All information of a nonlinear fited model are saved in `nl.fitt`, thus it can be large variable of informations. The generalized form `nl.fitt.gn` and robust forms `nl.fitt.rob` and `nl.fitt.rgn` of a fitt is children of this object. Typically it used to save Least-Square estimation method. But it is extensively used to save other fited objects such as hetroscedastic variance parameter fitts.

Author(s)

Hossein Riazoshams, May 2014. Email: <riazihosein@gmail.com> URL `http://www.riazoshams.com/nlr/`

References

Riazoshams H, Midi H, and Ghilagaber G, 2018,. Robust Nonlinear Regression, with Application using R, Joh Wiley and Sons.

See Also

`nl.fitt.gn`, `fittmethod`, `nl.fitt.rob`, `nl.fitt.rgn`, `Fault`, `nl.form`, `acf`

Examples

```
showClass("nl.fitt")
```

---

nl.fitt.gn *Class* "nl.fitt.gn"

Description

Object for Generalized fitt of nonlinear regression, inherited from `nl.fitt` object. Classic estimates of Heterogeneouity, Autocorrelated, or Weighted fitt will be saved in this object.

Objects from the Class

Objects can be created by calls of the form `new("nl.fitt.gn",...)`. Include all slots of `nl.fitt` and another slots for saving heterosedastic variance fit or autocorrelated error parameters.

Slots

`parameters`: Object of class `"list"`, estimate of nonlinear model θ.

`scale`: Object of class `"numericorNULL"`, standard deviation scale estimate σ.

`correlation`: Object of class `"numericorNULL"`, correlation structure of error.

`form`: Object of class `"nl.form"` of nonlinear model.

`response`: Object of class `"vectororMatrix"` response, left side of formula.

`predictor`: Object of class `"vectororMatrix"`, estimated predictor $\eta(\hat{\theta})$.

`curvature`: Object of class `"listorNULL"` of PE and IE curvatures.

`history`: Object of class `"matrixororNULL"` convergence computations in iteration procedures, include parameters, objective function and other parameters depends on the method.

`method`: Object of class `"fittmethodorNULL"` method of iteration used, contains main method, functions and sub methods. See `fittmethod`.

`data`: Object of class `"list"` data used in computation, including response and predictor variables.

`sourcefnc`: Object of class `"callorNULL"` source function called for fitt.

`Fault`: Object of class `"Fault"` of error or warnings if happened.

`others`: Object of class `"listorNULL"` of other computations, as an example the object of outlier detection measures will be saved in this slot later on. So far was parent slot `nl.fitt`. Following slots are other that represent generalized nonlinear regression fitt.

`vm`: Object of class `"matrix"` of variance covariance matrix of error.

`rm`: Object of class `"matrix"` of correlated error.

`hetro`: Object of class `"nl.fittorNULL"`, include object `nl.fitt` of heteroscedastic error fit, or `NULL` for non hetroscedastic. It include parameter estimates of hetroscedastic variance τ and all other slots of `nl.fitt` object which represent the variance function fitt information.

`autcorr`: Object of class `"listorNULL"` of autocorrelated error.

`autpar`: Object of class `"listorNULL"` of aprameters for autocorrelated error.

`gresponse`: Object of class `"vectororMatrix"` generalized response, transformed response equal $R \times y$, for cholesky decomposition R of covariance matrix of error.

`gpredictor`: Object of class `"vectororMatrix"` generalized predictor, transformed of predictor equal $R \times \eta(\theta)$, for cholesky decomposition R of covariance matrix of error.

Extends

Class `"nl.fitt"`, directly. Class `"nl.fittorNULL"`, by class "nl.fitt", distance 2.

Methods

atypicals `signature(nlfited = "nl.fitt.gn")`: Compute statistical measures to identify outliers.

parInfer `signature(object = "nl.fitt.gn")`: parameter inference, covariance matrix of parameters.

predictionI `signature(nlfited = "nl.fitt.gn")`: prediction interval.

recalc `signature(object = "nl.fitt")`: generic function, recalculate the object with new arguments given in dots argument.

residuals `signature(object = "nl.fitt.gn")`: residuals of fitt.

Note

`nl.fitt.gn` inherit `nl.fitt` object. It include heterogeneous or autocorrelated fitt. The heteroscedastic fitt result stores in `hetro` slot, and autocorelation result stores in `correlation,autcor` slot. meanwhile the `vm,rm` include contains general of covariance and correlation matrix of both heteroscedastic and autocorrelated informations. See Seber and Wild (2003) for details in generalized Nonlinear model.

Author(s)

Hossein Riazoshams, May 2014. Email: <riazihosein@gmail.com> URL `http://www.riazoshams.com/nlr/`

References

Riazoshams H, Midi H, and Ghilagaber G, 2018,. Robust Nonlinear Regression, with Application using R, Joh Wiley and Sons. Seber, G., A. F. and Wild, C. J. (2003). Nonlinear Regression. New York: John Wiley & Sons, Inc.

See Also

`nl.fitt`, `fittmethod`.

Examples

```
showClass("nl.fitt.gn")
```

---

nl.fitt.rgn *Class* "nl.fitt.rgn"

Description

Object of generalized robust estimates of nonlinear regression model.

Objects from the Class

Objects can be created by calls of the form `new("nl.fitt.rgn",...)`.

Slots

Robust generalized slots:

Object of class "matrix" of variance covariance matrix of error.

vmc: Object of class "matrix" of correlated error.

hetro: Object of class "nl.fittorNULL", include object nl.fitt of heteroscedastic error fit, or NULL for non hetroscedastic. It include parameter estimates of hetroscedastic variance τ and all other slots of nl.fitt object which represent the variance function fitt information.

autcorr: Object of class "listorNULL" of autocorrelated error.

autpar: Object of class "listorNULL" of aprameters for autocorrelated error.

gresponse: Object of class "vectororMatrix" generalized response, transformed response equal $R \times y$, for cholesky decomposition R of covariance matrix of error.

gpredictor: Object of class "vectororMatrix" generalized predictor, transformed of predictor equal $R \times \eta(\theta)$, for cholesky decomposition R of covariance matrix of error.

Robust estimate slots:

htheta: Object of class "vectororNULL" optimized objective loss function is equal sum of rho function, with gradient and hessian as attribute.

rho: Object of class "vectororNULL" computed robust ρ function, including gradient and hessian as attribute.

ri: Object of class "vectororNULL" residuals equal predictor values minus predicted values, with gradient and hessian as attribute.

curvrob: Object of class "listorNULL" robust Object of class "listorNULL" of PE and IE curvatures. Is not operational at the moment.

robform: Object of class "nl.formorNULL", robust ρ function of object type "nl.form".

Nonlinear model estimates, inherited slots from nl.form object follows.

parameters: Object of class "list", estimate of nonlinear model θ.

scale: Object of class "numericorNULL", standard deviation scale estimate σ.

correlation: Object of class "numericorNULL", correlation structure of error.

form: Object of class "nl.form" of nonlinear model.

response: Object of class "vectororMatrix" response, left side of formula.

predictor: Object of class "vectororMatrix", estimated predictor $\eta(\hat{\theta})$.

curvature: Object of class "listorNULL" of PE and IE curvatures.

history: Object of class "matrixororNULL" convergence computations in iteration procedures, include parameters, objective function and other parameters depends on the method.

method: Object of class "fittmethodorNULL" method of iteration used, contains main method, functions and sub methods. See fittmethod.

data: Object of class "list" data used in computation, including response and predictor variables.

sourcefnc: Object of class "callorNULL" source function called for fitt.

Fault: Object of class "Fault" of error or warnings if happened.

others: Object of class "listorNULL" of other computations, as an example the object of outlier detection measures will be saved in this slot later on.

Extends

Class "nl.fitt.rob", directly. Class "nl.fitt", by class "nl.fitt.rob", distance 2. Class "nl.fitt.roborNULL", by class "nl.fitt.rob", distance 2. Class "nl.fittorNULL", by class "nl.fitt.rob", distance 3.

Methods

parInfer signature(object = "nl.fitt"): parameter inference function, calculate covariance matrix of parameters and their confidence interval. Usage: parInfer(object,confidence = .95)

predictionI signature(nlfited = "nl.fitt.gn"): prediction interval. Usage: predictionI(nlfited,con data is new data that will be predicting the values for them.

residuals signature(object = "nl.fitt.gn"): residuals of fitt.

atypicals signature(nlfited = "nl.fitt"): detect atypical points by calculating outlier detection measures. Usage: atypicals(nlfited)

Note

All information of a generalized nonlinear robust fited model are saved in nl.fitt.rgn, thus it can be large variable of informations. It is inheritance of nl.fitt.rob, and robust form of nl.fitt.gn. It include heterogeneous or autocorrelated fitt. The heteroscedastic fitt result stores in hetro slot, and autocorelation result stores in correlation,autcor slot. meanwhile the vm,rm include contains general of covariance and correlation matrix of both heteroscedastic and autocorrelated informations.

Author(s)

Hossein Riazoshams, May 2014. Email: <riazihosein@gmail.com> URL http://www.riazoshams.com/nlr/

References

Riazoshams H, Midi H, and Ghilagaber G, 2018,. Robust Nonlinear Regression, with Application using R, Joh Wiley and Sons.

See Also

nl.fitt, nl.fitt.gn, fittmethod, nl.fitt.rob, Fault, nl.form

Examples

```
showClass("nl.fitt.rgn")
```

`nl.fitt.rob-class` *Class* `"nl.fitt.rob"`

Description

Object of robust estimates of nonlinear regression model.

Objects from the Class

Objects can be created by calls of the form `new("nl.fitt.rob",...)`.

Slots

`htheta`: Object of class `"vectororNULL"` optimized objective loss function is equal sum of rho function, with gradient and hessian as attribute.

`rho`: Object of class `"vectororNULL"` computed robust ρ function, including gradient and hessian as attribute.

`ri`: Object of class `"vectororNULL"` residuals equal predictor values minus predicted values, with gradient and hessian as attribute.

`curvrob`: Object of class `"listorNULL"` robust Object of class `"listorNULL"` of PE and IE curvatures. Is not operational at the moment.

`robform`: Object of class `"nl.formorNULL"`, robust ρ function of object type `"nl.form"`. Nonlinear model estimates, inherited slots from `nl.form` object follows.

`parameters`: Object of class `"list"`, estimate of nonlinear model θ.

`scale`: Object of class `"numericorNULL"`, standard deviation scale estimate σ.

`correlation`: Object of class `"numericorNULL"`, correlation structure of error.

`form`: Object of class `"nl.form"` of nonlinear model.

`response`: Object of class `"vectororMatrix"` response, left side of formula.

`predictor`: Object of class `"vectororMatrix"`, estimated predictor $\eta(\hat{\theta})$.

`curvature`: Object of class `"listorNULL"` of PE and IE curvatures.

`history`: Object of class `"matrixororNULL"` convergence computations in iteration procedures, include parameters, objective function and other parameters depends on the method.

`method`: Object of class `"fittmethodorNULL"` method of iteration used, contains main method, functions and sub methods. See `fittmethod`.

`data`: Object of class `"list"` data used in computation, including response and predictor variables.

`sourcefnc`: Object of class `"callorNULL"` source function called for fitt.

`Fault`: Object of class `"Fault"` of error or warnings if happened.

`others`: Object of class `"listorNULL"` of other computations, as an example the object of outlier detection measures will be saved in this slot later on.

Extends

Class "nl.fitt", directly. Class "nl.fitt.roborNULL", directly. Class "nl.fittorNULL", by class "nl.fitt", distance 2.

Methods

dlev signature(nlfited = "nl.fitt.rob"): DLEV Difference in LEverage measure.

JacobianLeverage signature(nlfited = "nl.fitt.rob"): Jacobian-Leverage for non-linear regression. Usage JacobianLeverage(nlfited = "nl.fitt.rob")

parInfer signature(object = "nl.fitt"): parameter inference function, calculate covariance matrix of parameters and their confidence interval. Usage: parInfer(object,confidence = .95)

plot signature(x = "nl.fitt",y = "missing",control=nlr.control(history=F,length.out=NU generic function extended to nl.fitt object. Plot the object. Usage. plot(x,y="missing",control=nlr.c If history is TRUE the convergence of fitt will be ploted.length.out is length of incremented p[redictor to acheive smooter curve. singlePlot=F plot the model and residuals in two collumn. If the estimate be Least MEdian Square, the plotlms function is used to plot the object.

predictionI signature(nlfited = "nl.fitt.gn"): prediction interval. Usage: predictionI(nlfited,con data is new data that will be predicting the values for them.

recalc signature(object = "nl.fitt.rob"): recalculate the original call of the fited model by some extra options. It is created for usage in atyp function.

Note

All information of a nonlinear robust fited model are saved in nl.fitt.rob, thus it can be large variable of informations. The generalized form nl.fitt.rgn of a fitt is children of this object. Typically it used to save robust MM-estimation method. But it is extensively used to save other fited objects such as hetroscedastic variance parameter fitts.

Author(s)

Hossein Riazoshams, May 2014. Email: <riazihosein@gmail.com> URL http://www.riazoshams.com/nlr/

References

Riazoshams H, Midi H, and Ghilagaber G, 2018,. Robust Nonlinear Regression, with Application using R, Joh Wiley and Sons..

See Also

nl.fitt, nl.fitt.gn, fittmethod, nl.fitt.rgn, Fault, nl.form

Examples

```
showClass("nl.fitt.rob")
```

`nl.fittorNULL` *"OR" Class*

Description

Set of "or" classes are created for compatibility reasons.

- nl.fittorNULL `nl.fitt` object or `NULL`
- functionorNULL `function` object or `NULL`
- expressionorNULL `expression` object or `NULL`
- callorNULL`call` object or `NULL`
- integerorNULL`integer` object or `NULL`
- numericorNULL`numeric` object or `NULL`
- characterorNULL`character` object or `NULL`
- logicalorNULL`logical` object or `NULL`
- listorNULL`list` object or `NULL`
- nl.fitt.roborNULL`nl.fitt.rob` object or `NULL`
- fittmethodorNULL`fittmethod` object or `NULL`
- vectororNULL`vector` object or `NULL`
- matrixororNULL`matrix` object or `NULL`
- vectororMatrix`vector` object or `matrix`

Objects from the Class

A virtual Class: No objects may be created from it.

Methods

No methods defined with class "nl.fittorNULL" in the signature.

Note

These classes used in several slots of objects. User might not use them directly.

Author(s)

Hossein Riazoshams, May 2014. Email: <riazihosein@gmail.com> URL `http://www.riazoshams.com/nlr/`

References

Riazoshams H, Midi H, and Ghilagaber G, 2018,. Robust Nonlinear Regression, with Application using R, Joh Wiley and Sons.

See Also

nl.fitt, nl.form, nl.fitt.rob, fittmethod

Examples

```
showClass("nl.fittorNULL")
```

---

nl.form-class *Class* "nl.form"

Description

It is a class of nonlinear regression model function, can embed expression formula and function. It is a more general object such that the heterogeneous variance model and robust loss function in nlr is saved as nl.form object.

Objects from the Class

Objects can be created by calls of the form:

```
new("nl.form",formula,fnc,formtype,p,inv=NULL,name=name,par,arguments=list(...),dep
```

,

or constructor:

```
nl.form(form,p=NULL,inv=NULL,name,par=NULL,dependent=NULL,independent=NULL,origin=N
```

Slots

formula: Object of class "call or NULL" it can be 1- a two sided formula with response (or a function of response) in left of ~ and nonlinear function model is a function of predictors and parameters, or 2- a one sided formula with ~nonlinear model in right, is again a function of predictors and unknown parameters. If Null then the nonlinear model is a R function stored in fnc slot as bellow. Each of right side or left side formula can return "gradient" and "hessian" as attributed value.

fnc: Object of class "function or NULL" is nonlinear model stored as R function.

dependent: Object of class "characterorNULL" of predictor variable, null value means the formula slot is one sided.

independent: Object of class "characterorNULL" caharacter vector name of predictor variables, which can be more than one predictor.

formtype: Object of class "character" character type of nl.form. Do not insert this slot, it will be set automatically by creater, it use internaly for evaluation.

p: Object of class "numericorNULL" number of parameters.

inv: Object of class "callorNULL" if nonlinear model is function of one predictor the inverse function define in this slot. Still not functioning, designed for feature extention, so it can be ignored at the moment, but it is better to define.

name: Object of class "character" a character name for the nonlinear function model.

`par`: Object of class `"list"` of parameters, assigned value to parameters will be used in worst case that initial values can not be computed.

`arguments`: Object of class `"list"` list of extra arguments use in `formula` or `fnc` slot.

`origin`: Object of class `"callorNULL"` is original one sided or two sided expression of non-linear model withought gradient and hessian.

`selfStart`: Object of class `"functionorNULL"` selfstart function defined for initial values guiss. Is same as nls or nlme::nlme functions.

Extends

Class `"nl.formorNULL"`, directly.

Methods

$ `signature(x = "nl.form")`: return slots.

all.vars `signature(expr = "nl.form")`: Return a character vector containing all the names which occur in `"formula"` or or `"fnc"`.

eval `signature(expr = "nl.form")`: evaluate `nl.form` object in the environment include parameters and predictor variables.

evald `signature(expr = "nl.form")`: same functionality as eval but extended for compatibility. `eval.nl.form` has same functionality created for compatibility purpose.

getInitial `signature(object = "nl.form")`: get initial value from selfstart or par slot.

selfStart `signature(model = "nl.form")`: self computing initial value, if not given `"getInitial"` method return initial values from environment or "par" slot.

eval.nl.form `signature(expr = "eval.nl.form")`: `eval.nl.form` has same functionality as `eval`, created for compatibility purpose.

Note

this object typicaly implemented to store a nonlinear regression model function informations. But extensively used in `nlr` package to save heteroscedastic variances and robust loss functions.

Author(s)

Hossein Riazoshams, 2013. Email: <riazihosein@gmail.com> URL `http://www.riazoshams.com/nlr/`

References

Riazoshams H, Midi H, and Ghilagaber G, 2018,. Robust Nonlinear Regression, with Application using R, Joh Wiley and Sons.

See Also

`nl.robfuncs`

Examples

```
# define hampel robust loss as a function
hampel <- nl.form(
  form = function(t,
                  a = 1.345,
                  k0 = 3.73677,
                  k1 = 4,
                  maxrho5 = 1.345,
                  ...) {
    U <- abs(t)
    Ugrta <- (U > abs(a))
    .rho <- .grad <- .hess <- .weight <- NULL
    .rho[Ugrta] <- 2. * abs(a) * U[Ugrta] - a * a
    .rho[!Ugrta] <- t[!Ugrta] ^ 2
    .grad[Ugrta] <- 2. * abs(a) * sign(t[Ugrta])
    .grad[!Ugrta] <- 2. * t[!Ugrta]
    .hess[Ugrta] <- 0.
    .hess[!Ugrta] <- 2.
    .weight[Ugrta] <- 2. * abs(a) / U[Ugrta]
    .weight[!Ugrta] <- 2.
    attr(.rho, "gradient") <- .grad
    attr(.rho, "hessian") <- .hess
    attr(.rho, "weight") <- .weight
    return(.rho)
  },
  name = "huber",
  independent = "t",
  a = 1.345,
  k0 = 3.73677,
  k1 = 4,
  maxrho5 = 1.345
)
```

---

nl.formorNULL *Class* "nl.formorNULL"

Description

nl.formorNULL, Union class that can have `nl.form` value or NULL value.

Objects from the Class

A virtual Class: No objects may be created from it.

Methods

Union class used in assignment of slots, in nl.fitt.gn, and others. No methods defined with class "nl.formorNULL" in the signature.

Note

For internal usage, it might not call directly by user.

Author(s)

Hossein Riazoshams, Apr 2014. Email: <riazihosein@gmail.com> URL http://www.riazoshams.com/nlr/

References

Robust Nonlinear Regression, Theories and Methods with Practical Guides for R Packages. Riazoshams et al.

See Also

nl.form

Examples

```
showClass("nl.formorNULL")
```

---

nl.hetro *Classic Multi Stage Estimate (CME).*

Description

(CME) Classic multi stage estimate for nonlinear regression with heteroscedastic error, when variance is function of unkown parameters. The variance function model parameter estimate using pseudo chi-square likelihood of computed sample variance.

Usage

```
nl.hetro(formula, data, start = getInitial(formula, data),
control = nlr.control(tolerance = 1e-05, minlanda = 1/2^10,
maxiter = 25 * length(start)), varmodel, tau = NULL, ...)
```

Arguments

formula	nl.form object of the nonlinear function model.
data	list of data include responce and predictor.
start	list of parameter values of nonlinear model function (θ.
control	list of nlr.control for controling convergence criterions.
varmodel	nl.fomr object of variance function model for heteroscedastic variance.
tau	list of initial values for variance model function varmodel argument.
...	extra arguments to nonlinear regression model, heteroscedastic variance function, robust loss function or its tuning constants.

Details

In stage 1 the nonlinear model parameter estimates by Classic OLS, Stage 2 compute sample variance of data, Stage 3 estimate the parameter of variance function model by maximizing the chi-square pseudo-likelihood function. Stage 4 estimate the final value of function model parameter by generalized least square.

Value

generalized fitt object `nl.fitt.gn`. The `hetro` slot include parameter estimate and other information of fitt for heteroscedastic variance model.

`parameters`	nonlinear regression parameter estimate of θ.
`correlation`	of fited model.
`form`	`nl.form` object of called nonlinear regression model.
`response`	computed response.
`predictor`	computed (right side of formula) at estimated parameter with gradient and hessian attributes.
`curvature`	list of curvatures, see `curvature` function.
`history`	matrix of convergence history, collumns include: convergence index, parameters, minimized objective function, convergence criterion values, or other values. These values will be used in `plot` function in ploting history.
`method`	`fittmethod` object of method used for fitt.
`data`	list of called data.
`sourcefnc`	Object of class `"callorNULL"` source function called for fitt.
`Fault`	`Fault` object of error, if no error Fault number = 0 will return back.
`vm`	covariance matrix, diagonal of variance model predicted values.
`rm`	cholesky decomposition of vm.
`gresponse`	transformed of response by rm, include gradinet and hessian attributes.
`gpredictor`	transformed of predictor by rm, include gradinet and hessian attributes.
`hetro`	`nl.fitt` object of fited variance odel:

- parametersestimate of variance parameter τ
- form`nl.form` object of called `varmodel`.
- predictorvariance model computed at estimated parameter, $H(x; \hat{\tau})$
- responsesample variance computed used as response variable.
- historymatrix of convergence history, collumns include: convergence index, parameters, minimized objective function, convergence criterion values, or other values.
- method`fittmethod` object of method used for fitt.
- dataresponse (z_i) and predictor `t` variable values, used to computing the variance model.
- sourcefncObject of class `"callorNULL"` source function called for fitt.
- Fault`Fault` object of error, if no error Fault number = 0 will return back.

Note

Heteroscedastic variance can have several cases, this function assume variance is parameteric function of predictor ($H(t;\tau)$). If data does not include the predictor variable of `varmodel` (t), the predicted of function model $f(x;\hat{\theta})$ will replace for (t), otherwise user have to defin (t) or (x) as predictor variable of (H).

Author(s)

Hossein Riazoshams, May 2014. Email: <riazihosein@gmail.com> URL `http://www.riazoshams.com/nlr/`

References

Riazoshams, H,. 2010. Outlier detection and robust estimation methods for nonlinear regression having autocorrelated and heteroscedastic errors. PhD thesis disertation, University Putra Malaysia.

Riazoshams H, Midi H, and Ghilagaber G, 2018,. Robust Nonlinear Regression, with Application using R, Joh Wiley and Sons.

See Also

`fittmethod`, `nl.form`, `nl.fitt`, `nl.fitt.gn`

Examples

```
chkft<- nl.hetro(formula=nlrobj1[[14]], data=list(xr=Weights$Date,yr=Weights$Weight),
start=list(p1=2200,p2=38,p3=.11), tau=list(sg=.09,landa=2),varmodel=nlrobjvarmdls1[[1]])
chkft$parameters
```

`nl.hetroLS` *CLSME estimate.*

Description

Classic Least square based Multi Stage Estimate (CLSME) for heteroscedastic error case.

Usage

```
nl.hetroLS(formula, data, start = getInitial(formula, data),
control = nlr.control(tolerance = 0.001, minlanda = 1/2^10,
maxiter = 25 * length(start)), varmodel, tau = getInitial(varmodel, vdata), ...)
```

Arguments

formula	nl.form object of the nonlinear function model.
data	list of data include responce and predictor.
start	list of parameter values of nonlinear model function (θ. in $f(x, \theta)$).
control	list of nlr.control for controling convergence criterions.
varmodel	nl.fomr object of variance function model for heteroscedastic variance.
tau	list of initial values for variance model function varmodel argument.
...	extra arguments to nonlinear regression model, heteroscedastic variance function, robust loss function or its tuning constants.

Details

Least square based estimate for nonlinear regression with hetroscedastic error when variance is a general function of unkown parameters.

Value

generalized fitt object nl.fitt.gn. The hetro slot include parameter estimate and other information of fitt for heteroscedastic variance model.

(parameters	nonlinear regression parameter estimate of θ.
correlation	of fited model.
form	nl.form object of called nonlinear regression model.
response	computed response.
predictor	computed (right side of formula) at estimated parameter with gradient and hessian attributes.
curvature	list of curvatures, see curvature function.
history	matrix of convergence history, collumns include: convergence index, parameters, minimized objective function, convergence criterion values, or other values. These values will be used in plot function in ploting history.
method	fittmethod object of method used for fitt.
data	list of called data.
sourcefnc	Object of class "callorNULL" source function called for fitt.
Fault	Fault object of error, if no error Fault number = 0 will return back.
vm	covariance matrix, diagonal of variance model predicted values.
rm	cholesky decomposition of vm.
gresponse	transformed of response by rm, include gradinet and hessian attributes.
gpredictor	transformed of predictor by rm, include gradinet and hessian attributes.
hetro	nl.fitt object of fited variance odel:

- parametersestimate of variance parameter τ
- formnl.form object of called varmodel.
- predictorvariance model computed at estimated parameter, $H(x; \hat{\tau})$

- responsesample variance computed used as response variable.
- historymatrix of convergence history, collumns include: convergence index, parameters, minimized objective function, convergence criterion values, or other values.
- methodfittmethod object of method used for fitt.
- dataresponse (z_i) and predictor t variable values, used to computing the variance model.
- sourcefncObject of class "callorNULL" source function called for fitt.
- FaultFault object of error, if no error Fault number = 0 will return back.

Note

Heteroscedastic variance can have several cases, this function assume variance is parameteric function of predictor ($H(t;\tau)$). If data does not include the predictor variable of varmodel (t), the predicted of function model $f(x;\hat{\theta})$ will replace for (t), otherwise user have to defin (t) or (x) as predictor variable of (H).

Author(s)

Hossein Riazoshams, May 2016, ongoing book. Email: <riazihosein@gmail.com> URL http://www.riazoshams.com/nlr/

References

Riazoshams, H. (2012), Robustifying the Least Squares estimate of parameters of variance model function in nonlinear regression with heteroscedastic variance, Poster Presentation, Royal Statistical Society Conference (RSS) 2012, Telford, UK.

See Also

fittmethod, nl.form, nl.fitt, nl.fitt.gn

Examples

```
chkft<- nl.hetroLS(formula=nlrobj1[[14]], data=list(xr=Weights$Date,yr=Weights$Weight),
start=list(p1=2200,p2=38,p3=.11), tau=list(sg=.09,landa=2),varmodel=nlrobjvarmdls1[[1]])
chkft$parameters
```

---

nl.lmsGA

Fitt a nonlinear regression model by least median of squares. The Optimization is done by genetic algorithm.

Description

Least Median of square estimate is robust fitt by minimizing the median of squared residuals. This function use the "ga" function,from "GA" package, which minimize using genetic algorithm method.

Usage

```
nl.lmsGA(formula, data, start,min=NULL,max=NULL,type="real-valued")
```

Arguments

formula	nl.form object of the nonlinear function model.
data	list of data include responce and predictor.
start	list of parameter values of nonlinear model function (θ. in $f(x, \theta)$).
min	vector of minimum values of parameters, which passes to "ga" function.
max	vector of maximum values of parameters, which passes to "ga" function.
type	the type of genetic algorithm to be run, which passes to "ga" function.

Details

Note that due to using genetic algorithm method, this function is mor efficient than nl.lmsNM, but estimator due to non uniqueness of minimum of objective function is not efficient, but is high breakdown estimate.

Value

list of parameter estimates.

Note

When that function is not working properly, nl.lmsNM function can be used, but requires more precise initial values. These function uses by nlr for initial purposes. User can request least median square as initial value from nlr.control argument by initials="lms" method, nlr use nl.lmsGA not nl.lmsNM.

Author(s)

Hossein Riazoshams, May 2014. Email: <riazihosein@gmail.com> URL http://www.riazoshams.com/nlr/

References

Riazoshams H, Midi H, and Ghilagaber G, 2018,. Robust Nonlinear Regression, with Application using R, Joh Wiley and Sons.

See Also

nl.lmsNM, nlr.control, nlr

Examples

```
# chicken data fitt example
data=list(xr=Weights$Date,yr=Weights$Weight)
fit<- nl.lmsGA(nlrobj1[[14]],data=data,start=list(p1=100,p2=42,p3=.11))
```

`nl.lmsNM` *Fitt a nonlinear regression model by least median of squares. The Optimization is done by golden section method.*

Description

Least Median of square estimate is robust fitt by minimizing the median of squared residuals. This function use the "optim" function which minimize using Golden section method.

Usage

```
nl.lmsNM(formula, data, start)
```

Arguments

`formula`	`nl.form` object of the nonlinear function model.
`data`	list of data include responce and predictor.
`start`	list of parameter values of nonlinear model function ($\theta.$ in $f(x, \theta)$).

Details

Note that due to using classic optimization method, and due to non uniqueness of minimum of objective function this estimate is not efficient, but is high breakdown estimate.

Value

list of parameter estimates.

Note

A more efficient function is recomended is `nl.lmsGA`. When that function is not working properly this function can be used, but requires more precise initial values. These function uses by `nlr` for initial purposes. User can request least median square as initial value from `nlr.control` argument by `initials="lms"` method, nlr use `nl.lmsGA` not `nl.lmsNM`.

Author(s)

Hossein Riazoshams, May 2014. Email: <riazihosein@gmail.com> URL `http://www.riazoshams.com/nlr/`

References

Riazoshams H, Midi H, and Ghilagaber G, 2018,. Robust Nonlinear Regression, with Application using R, Joh Wiley and Sons.

See Also

`nl.lmsGA`, `nlr.control`, `nlr`

Examples

```
# chicken data fitt example
data=list(xr=Weights$Date,yr=Weights$Weight)
fit<- nl.lmsNM(nlrobj1[[14]],data=data,start=list(p1=1000,p2=42,p3=.11))
fit
```

---

nl.lts *Compute (LTS) Least Trimmed Square Estimate.*

Description

LTS is minimizing trimmed sum of squares.

Usage

```
nl.lts(formula, data, start, h = NULL, control = nlr.control())
```

Arguments

formula	nl.form object of the nonlinear model function.
data	List of data used in predictor and response.
start	List of parameter starting value.
h	Percentage of trimming the residuals, if omited the default 25% will be used.
control	nlr.control options, will be submited to least square nlsqr function.

Details

LTS trimme h percent of residuals first then compute the least square estimate, and final parameter estimate is the one minimize the sum of squares of errors.

Value

nl.fitt object of fitted model parameter.

Note

The result data returnd in fitted object is trimmed data.

Author(s)

Hossein Riazoshams, May 2014. Email: <riazihosein@gmail.com> URL http://www.riazoshams.com/nlr/

References

Riazoshams H, Midi H, and Ghilagaber G, 2018,. Robust Nonlinear Regression, with Application using R, Joh Wiley and Sons.

See Also

`nl.form`, `nl.fitt`

Examples

```
data=list(xr=Weights$Date,yr=Weights$Weight)
fit<- nl.lts(nlrobj1[[14]],data=data,start=list(p1=1000,p2=42,p3=.11))
fit$parameters
```

---

nl.MLE *Nonlinear MLE*

Description

MLE estimate of a nonlinear function. with hetro variance model function, and weights.

Usage

```
nl.MLE(formula, data, start = getInitial(formula, data), vm = NULL,
rm = solve(t(chol(vm))),
control =nlr.control(derivfree = T),
varmodel = NULL, tau = varmodel$par, ...)
```

Arguments

`formula`	`nl.form` object of the nonlinear function model.
`data`	list of data include responce and predictor.
`start`	list of parameter values of nonlinear model function (θ.
`vm`	optional covariance matrix.
`rm`	optional cholesky decomposition of covariance matrix.
`control`	list of `nlr.control` for controling convergence criterions. Defaul value of `derivfree` is "True", force function to use derivative free methods. But it can be "False" to use derivative based, has faster convergence.
`varmodel`	`nl.fomr` object of variance function model for heteroscedastic variance.
`tau`	list of initial values for variance model function `varmodel` argument.
`...`	extra arguments to nonlinear regression model, heteroscedastic variance function, robust loss function or its tuning constants.

Details

Calculate Maximum Likelihood estimate in several sitautions, if `varmodel` is given the heteroscedastic variance consider. If `vm` or `rm` is given, weighted is computing.

Value

Depending given options different fitt object will result as follow

- if vm=NULL and varmodel=NULLrepresent homogeneous and uncorrelated erro, output is `nl.fitt` object generated by `nlsqr` or `nlsnm` for derivative based and derivative free method respectivly given by `derivfree` option.
- if vm=NULL and varmodel is given represent heteroscedastic variance case, output is `nl.fitt.gn` generated by `nl.robhetroWM`, depends on using derivative free method or no.
- if vm is given represent general covariance matrix as weight, ouput is `nl.fitt.gn` generated by `nlsqr.gn`.

Note

The objective function used in `nl.robhetroWM` is general form of Likelihood, thus it can generate Least Square estimate using quadratic function, which can be accessed in nl.robfuncs[["least square"]] variable. It can include parameteric variance function also. Due to compatibility it is beter to be called from `nlr` function rather than direct call by user.

Author(s)

Hossein Riazoshams, May 2014. Email: <riazihosein@gmail.com> URL `http://www.riazoshams.com/nlr/`

References

Riazoshams, H,. 2010. Outlier detection and robust estimation methods for nonlinear regression having autocorrelated and heteroscedastic errors. PhD thesis disertation, University Putra Malaysia.

Riazoshams H, Midi H, and Ghilagaber G, 2018,. Robust Nonlinear Regression, with Application using R, Joh Wiley and Sons.

See Also

`nlsqr.gn`, `nl.robhetroWM`,`nl.fitt`, `nl.fitt.gn`, `nlsnm`, `nlsqr`, `nlr.control`

Examples

```
## The function is currently defined as
"nl.MLE"
```

`nl.mscale` *Scale M-estimator with 50% breakdown.*

Description

Compute High Breakdown point M-estimate of scale σ.

Usage

```
nl.mscale(u, robfunc, ...)
```

Arguments

`u`	Residuals $r_i = y_i - f(x_i; \theta)$.
`robfunc`	Robust ρ_2 function used to compute M-estimate of scale/
`...`	Ane other parameter passed to ρ function, and others.

Details

This estimate is used in MM-estimate procedure of location parameter θ.

Value

Single Numeric value of σ estimate.

Note

Its value is used inside MM-estimate procedure of parameter θ. Currently work with Hampel ρ function, not all of ρ defined functions. It is called by `nlmest.NLM` in scale estimation steps, might not be called directly by user.

Author(s)

Hossein Riazoshams, May 2014. Email: <riazihosein@gmail.com> URL `http://www.riazoshams.com/nlr/`

References

Yohai (1987) Annals, Stromberg (1993) JASA.

See Also

`nlmest.NLM`

Examples

```
## The function is currently defined as
"nl.mscale"
```

`nl.robcorrts` *Robust two stage estimate*

Description

Robust Two stage estimate for nonlinear regreession model with autocorrelated error. `dfr.robcorrts` is derivative free version.

Usage

```
nl.robcorrts(formula, data, start = getInitial(formula, data),
control = nlr.control(tolerance = 0.001, minlanda = 1/2^10,
maxiter = 25 * length(start)), correlation =  list(StructName = "NAN",
manualcorr = NULL), robfunc, ...)
```

Arguments

`formula`	nl.form object of the nonlinear function model. See `nl.form` object.
`data`	list of data with the response and predictor as name of variable.
`start`	list of starting value parameter, name of parameters must be represented as names of variable in the list.
`control`	nlr.control object, include tolerance, maxiter,... see `nlr.control`.
`correlation`	correlation structure, at the moment parameter of AR(p) process.
`robfunc`	`nl.form` object of robust function.
`...`	any other argument pass to `formula` or `robfunc`.

Details

In first stage nonlinear regression parameter estimate by robust MM method, and in second stage autocorrelation structure estimate and finally the generalized MM-estimates the function model parameters.

Value

t2st <- nlmest.NLM(formula, data=data, start=st,robfunc=robfunc,vm=vmat,rm=rmat,control=control,...)
result <- list(fited=t2st,tm=tm)

`fited`	`nl.fitt.rgn` object generated by `nlmest.NLM` function.
`tm`	fitted time series model for residuals.

Note

This function currently run with AR process. The classic estimate is don by `nl.corrts` function. This function call from `nlr`, due to compatibility it is more efficient to call `nlr` by user rather than this function explicitly.

Author(s)

Hossein Riazoshams, Jul 2009. Email: <riazihosein@gmail.com> URL http://www.riazoshams.com/nlr/

References

Riazoshams, H., Midi, H., Sharipov, O. S.H, (2010). The Performance of Robust Two Stage Estimator in Nonlinear Regression with autocorrelated Error, Communications in Statistics - Simulation and Computation, 39: 1251-1268.

See Also

nl.corrts, nlmest.NLM, nl.robcorrts, nlsqr.gn, nl.fitt.gn, nlr.control

Examples

```
xr = trade.ir[, 1]
yr = trade.ir[, 2]
a1 <-  nl.robcorrts( nlrobj5[[18]],data=list(xr = xr, yr = yr),
correlation=list(StructName="corAR1"),
robfunc = nl.robfuncs[["hampel"]])
a1$parameters
```

---

nl.robfuncs *Robust Loss functions provided for* nlr.

Description

List of nl.form objects of 7 pre defined robust loss functions. Robust loss functions used for robust estimating parameters. They can be used in several part of the package functions.

Usage

```
nl.robfuncs
```

Format

The format is: list nl.robfuncs[index]

- nl.robfuncs[1]: huber function.
- nl.robfuncs[2]: hampel function.
- nl.robfuncs[3]: bisquare function.
- nl.robfuncs[4]: andrew function.
- nl.robfuncs[5]: halph huber function.
- nl.robfuncs[6]: hampel 2 function.
- nl.robfuncs[7]: least square (quadratic) function.

Details

Each of loss functions include tuning parameters as extra argument to `nl.form`. The result is the computed loss function with attributes of "gradient", "hessian", "weights". the object are mostly stored in fnc slot which is function, it can be called directly as a function.

Source

Riazoshams H, Midi H, and Ghilagaber G, 2018,. Robust Nonlinear Regression, with Application using R, Joh Wiley and Sons.

References

Riazoshams H, Midi H, and Ghilagaber G, 2018,. Robust Nonlinear Regression, with Application using R, Joh Wiley and Sons.

Examples

```
data(nl.robfuncs)
## maybe str(nl.robfuncs) ; plot(nl.robfuncs) ...
## the object are mostly stored in fnc slot which is function, it can be called
## directly as a function.
plot(seq(-6,6,length.out=30),nl.robfuncs[[1]]$fnc(seq(-6,6,length.out=30)),type="l",
xlab="t",ylab="rho",main=nl.robfuncs[[1]]$name)
```

`nl.robhetro` *Robust Multi Stage Estimate.*

Description

(RME) for nonlinear regression with heteroscedastic variance, when the variance of error is general parameteric function of unkown parameters. Robust form of CME (See `nl.hetro`).

Usage

```
nl.robhetro(formula, data, start = getInitial(formula, data),
control = nlr.control(tolerance = 1e-05, minlanda = 1/2^10,
maxiter = 25 * length(start)), robfunc, varmodel, tau = NULL, ...)
```

Arguments

`formula`	`nl.form` object of the nonlinear function model.
`data`	list of data include responce and predictor.
`start`	list of parameter values of nonlinear model function (θ.
`control`	list of `nlr.control` for controling convergence criterions.
`robfunc`	nl.form object of robust function used for downgrading.
`varmodel`	`nl.fomr` object of variance function model for heteroscedastic variance.

tau	list of initial values for variance model function varmodel argument.
...	extra arguments to nonlinear regression model, heteroscedastic variance function, robust loss function or its tuning constants.

Details

In stage 1 the nonlinear model parameter estimates by robust MM-estimate, Stage 2 compute robust sample variance of data, Stage 3 estimate the parameter of variance function model by maximizing the robustified form of chi-square pseudo-likelihood function. Stage 4 estimate the final value of function model parameter by generalized robust MM-estimate.

Value

nl.fitt.rgn for heterogeneous and autocorrelated error (nonlinear fitt robust generalized) will return.

parameters	nonlinear regression parameter estimate of θ.
correlation	of fited model.
form	nl.form object of called nonlinear regression model.
response	computed response.
predictor	computed (right side of formula) at estimated parameter with gradient and hessian attributes.
curvature	list of curvatures, see curvature function.
history	matrix of convergence history, collumns include: convergence index, parameters, minimized objective function, convergence criterion values, or other values. These values will be used in plot function in ploting history.
method	fittmethod object of method used for fitt.
data	list of called data.
sourcefnc	Object of class "callorNULL" source function called for fitt.
Fault	Fault object of error, if no error Fault number = 0 will return back.
htheta	robust loss value including gradient and hessain attributes.
rho	computed robust rho function, including gradient and hessain attributes.
ri	estimated residuals, including gradient and hessain attributes.
curvrob	curvature
robform	nl.form object of robust loss rho function.
vm	covariance matrix, diagonal of variance model predicted values.
rm	cholesky decomposition of vm.
gresponse	transformed of response by rm, include gradinet and hessian attributes.
gpredictor	transformed of predictor by rm, include gradinet and hessian attributes.
hetro	nl.fitt.rob object of fited variance odel:

- parametersestimate of variance parameter τ
- formnl.form object of called varmodel.

- predictorvariance model computed at estimated parameter, $H(x;\hat{\tau})$
- responsesample variance computed used as response variable.
- historymatrix of convergence history, collumns include: convergence index, parameters, minimized objective function, convergence criterion values, or other values.
- method`fittmethod` object of method used for fitt.
- dataresponse (z_i) and predictor `t` variable values, used to computing the variance model.
- sourcefncObject of class `"callorNULL"` source function called for fitt.
- Fault`Fault` object of error, if no error Fault number = 0 will return back.
- hthetarobust loss value including gradient and hessain attributes, for variance model. In fact is loglikelihood values.
- rhocomputed robust rho function, including gradient and hessain attributes.

`others` $refvar reference variance. variance of zi's.

Note

Heteroscedastic variance can have several cases, this function assume variance is parameteric function of predictor ($H(t;\tau)$). If data does not include the predictor variable of `varmodel` (t), the predicted of function model $f(x;\hat{\theta})$ will replace for (t), otherwise user have to defin (t) or (x) as predictor variable of (H).

Author(s)

Hossein Riazoshams

References

Riazoshams, H,. 2010. Outlier detection and robust estimation methods for nonlinear regression having autocorrelated and heteroscedastic errors. PhD thesis disertation, University Putra Malaysia.

See Also

`fittmethod`, `nl.form`, `nl.fitt.rob`, `nl.fitt.rgn`

Examples

```
# ntp data fitt
# tolerance is set as 1e-3 for testing purposes
# is not accurate enough, user can increase it.
ntpstart=list(p1=.12,p2=6,p3=1,p4=33)
ntpstarttau=list(tau1=-.66,tau2=2,tau3=.04)
datalist=list(xr=ntp$dm.k,yr=ntp$cm.k)
datalist[[nlrobjvarmdls3[[2]]$independent]]<-ntp$dm.k
aa1 <- nl.robhetro(formula=nlrobj1[[16]],data=datalist,
start=ntpstart,robfunc=nl.robfuncs[["hampel"]],
tau=ntpstarttau,varmodel=nlrobjvarmdls3[[2]],robscale=T,method="NM",
control=nlr.control(tolerance=1e-4))
aa1$parameters
```

`nl.robhetroLS` *Robust Generalized Multistage Estimate (RGME).*

Description

(RGME) for heteroscedastic error case, robust form of CLsME (See `nl.hetroLS`)

Usage

```
nl.robhetroLS(formula, data, start = getInitial(formula, data),
control = nlr.control(tolerance = 1e-05, minlanda = 1/2^10,
maxiter = 30 * length(start), robscale = T), robfunc, varmodel, tau = varmodel$pa
```

Arguments

`formula`	`nl.form` object of the nonlinear function model.
`data`	list of data include responce and predictor.
`start`	list of parameter values of nonlinear model function (θ. in $f(x, \theta)$).
`control`	list of `nlr.control` for controling convergence criterions.
`robfunc`	nl.form object of robust function used for downgrading.
`varmodel`	`nl.fomr` object of variance function model for heteroscedastic variance.
`tau`	list of initial values for variance model function `varmodel` argument.
`...`	extra arguments to nonlinear regression model, heteroscedastic variance function, robust loss function or its tuning constants.

Details

Robustified form of Least square based estimate for nonlinear regression with hetroscedastic error when variance is a general function of unkown parameters.

Value

return object `nl.fitt.rgn` for nonlinear regression wuth heterogeneous error.

`parameters`	nonlinear regression parameter estimate of θ.
`correlation`	of fited model.
`form`	`nl.form` object of called nonlinear regression model.
`response`	computed response.
`predictor`	computed (right side of formula) at estimated parameter with gradient and hessian attributes.
`curvature`	list of curvatures, see `curvature` function.
`history`	matrix of convergence history, collumns include: convergence index, parameters, minimized objective function, convergence criterion values, or other values. These values will be used in `plot` function in ploting history.

method	fittmethod object of method used for fitt.
data	list of called data.
sourcefnc	Object of class "callorNULL" source function called for fitt.
Fault	Fault object of error, if no error Fault number = 0 will return back.
htheta	robust loss value including gradient and hessain attributes.
rho	computed robust rho function, including gradient and hessain attributes.
ri	estimated residuals, including gradient and hessain attributes.
curvrob	curvature
robform	nl.form object of robust loss rho function.
vm	covariance matrix, diagonal of variance model predicted values.
rm	cholesky decomposition of vm.
gresponse	transformed of response by rm, include gradinet and hessian attributes.
gpredictor	transformed of predictor by rm, include gradinet and hessian attributes.
hetro	nl.fitt.rob object of fited variance odel:

- parametersestimate of variance parameter τ
- formnl.form object of called varmodel.
- predictorvariance model computed at estimated parameter, $H(x;\hat{\tau})$
- responsesample variance computed used as response variable.
- historymatrix of convergence history, collumns include: convergence index, parameters, minimized objective function, convergence criterion values, or other values.
- methodfittmethod object of method used for fitt.
- dataresponse (z_i) and predictor t variable values, used to computing the variance model.
- sourcefncObject of class "callorNULL" source function called for fitt.
- FaultFault object of error, if no error Fault number = 0 will return back.
- hthetarobust loss value including gradient and hessain attributes, for variance model. In fact is loglikelihood values.
- rhocomputed robust rho function, including gradient and hessain attributes.

Note

Heteroscedastic variance can have several cases, this function assume variance is parameteric function of predictor ($H(t;\tau)$). If data does not include the predictor variable of varmodel (t), the predicted of function model $f(x;\hat{\theta})$ will replace for (t), otherwise user have to defin (t) or (x) as predictor variable of (H).

This function is called from nlr, for compatibility it is more efficient to be called by nlr than callind directly.

Author(s)

Hossein Riazoshams, May 2014. Email: <riazihosein@gmail.com> URL http://www.riazoshams.com/nlr/

References

Riazoshams, H. (2012), Robustifying the Least Squares estimate of parameters of variance model function in nonlinear regression with heteroscedastic variance, Poster Presentation, Royal Statistical Society Conference (RSS) 2012, Telford, UK.

See Also

`fittmethod`, `nl.form`, `nl.fitt.rob`, `nl.fitt.rgn`

Examples

```
# function defined as
"nl.robhetroLS"
```

`nl.robhetroWM` *Weighted M-estimate.*

Description

Weighted M-estimate is robustified form of MLE, for nonlinear regression with heteroscedastic error, when variance is parameteric fucntion form. Both nonliner regression model parameter and variance function parameters compute simultanously by minizing the robustified objective function form.

Usage

```
nl.robhetroWM(formula, data, start = getInitial(formula, data),
control = nlr.control(tolerance = 1e-04, minlanda = 1/2^10,
maxiter = 50 * length(start), derivfree = T), robfunc, varmodel, tau = varmodel$par
```

Arguments

`formula`	`nl.form` object of the nonlinear function model.
`data`	list of data include responce and predictor.
`start`	list of parameter values of nonlinear model function ($\theta.$ in $f(x, \theta)$).
`control`	list of `nlr.control` for controling convergence criterions.
`robfunc`	nl.form object of robust function used for downgrading.
`varmodel`	`nl.fomr` object of variance function model for heteroscedastic variance.
`tau`	list of initial values for variance model function `varmodel` argument.
`...`	extra arguments to nonlinear regression model, heteroscedastic variance function, robust loss function or its tuning constants.

Details

For minimizing the objective function simultaneously for theta and tau, derivative free method Nelder-Mead is used.

Value

return object `nl.fitt.rgn` for nonlienar regression with heterogeneous error.

`parameters`	nonlinear regression parameter estimate of θ.
`correlation`	of fited model.
`form`	`nl.form` object of called nonlinear regression model.
`response`	computed response.
`predictor`	computed (right side of formula) at estimated parameter with gradient and hessian attributes.
`history`	matrix of convergence history, collumns include: convergence index, parameters, minimized objective function, convergence criterion values, or other values. These values will be used in `plot` function in ploting history.
`method`	`fittmethod` object of method used for fitt.
`data`	list of called data.
`sourcefnc`	Object of class `"callorNULL"` source function called for fitt.
`Fault`	`Fault` object of error, if no error Fault number = 0 will return back.
`htheta`	robust loss likelihood value including gradient and hessain attributes.
`rho`	computed robust rho function, including gradient and hessain attributes.
`ri`	estimated residuals, including gradient and hessain attributes.
`robform`	`nl.form` object of robust loss rho function.
`vm`	covariance matrix, diagonal of variance model predicted values.
`rm`	cholesky decomposition of vm.
`hetro`	`nl.fitt.rob` object of fited variance odel: • parametersestimate of variance parameter τ • form`nl.form` object of called `varmodel`. • predictorvariance model computed at estimated parameter, $H(x;\hat{\tau})$ • responsesample variance computed used as response variable.
`others`	$refvar reference variance. variance of zi's.

Note

Heteroscedastic variance can have several cases, this function assume variance is parameteric function of predictor ($H(t;\tau)$). If data does not include the predictor variable of `varmodel` (t), the predicted of function model $f(x;\hat{\theta})$ will replace for (t), otherwise user have to defin (t) or (x) as predictor variable of (H).

Author(s)

Lim, C., Sen, P. K., Peddada, S. D.

References

Lim, C., Sen, P. K., Peddada, S. D. (2010). Statistical inference in nonlinear regression under heteroscedasticity. Sankhya B 72:202-218.

See Also

fittmethod, nl.form, nl.fitt.rob, nl.fitt.rgn

Examples

```
ntpstart=list(p1=.12,p2=6,p3=1,p4=33)
ntpstarttau=list(tau1=-.66,tau2=2,tau3=.04)
datalist=list(xr=ntp$dm.k,yr=ntp$cm.k)
datalist[[nlrobjvarmdls3[[2]]$independent]]<-ntp$dm.k
# ntp data fitt
# tolerance is set as 1e-3 for testing purposes
# is not accurate enough, user can increase it.
bb1 <- nl.robhetroWM(formula=nlrobj1[[15]],data=datalist,
start=ntpstart,robfunc=nl.robfuncs[["least square"]],
tau=ntpstarttau,varmodel=nlrobjvarmdls3[[2]],control=nlr.control(tolerance=1e-3,maxiter=1500
bb1$parameters
#---------------- hampel -----------------
aa1 <- nl.robhetroWM(formula=nlrobj1[[15]],data=datalist,start=ntpstart,
robfunc=nl.robfuncs[["hampel"]],derivfree=T,
tau=ntpstarttau,varmodel=nlrobjvarmdls3[[2]],
control=nlr.control(tolerance=1e-3,maxiter=1500))#,delta=c(0.2,1,1,160,.2,1,.03))
aa1$parameters
```

---

nl.robmeas *Class* "nl.robmeas"

Description

Outlier detection measure object.

Objects from the Class

Objects can be created by calls of the form new("nl.robmeas",...).

Slots

measure: Object of class "numeric" vector of statistics measure for each data points.

cutofpoint: Object of class "numeric" cut of point for the measure.

name: Object of class "character" name of the measure.

Methods

$ signature(x = "nl.robmeas"): access the slots.

plot signature(x = "nl.robmeas",y = "ANY"): plot sequence of measure and cut of point line. Usually if the value is more than cut of point the data point might be outlier or influence observation. In contrast plot.atyps function is to plot "nl.robmeas" object for multiple purpose. If more than such objects are stored in a list the plot will iterate on all cells and draw all. It can be called as:

plot.atyps(x,...) #where x is lis that include more than one "nl.robmeas" object.

Note

nl.robmeas usually generate from atypical method to calculate several measures for identifying outlier or influential observations.

Author(s)

Hossein Riazoshams, May 2014. Email: <riazihosein@gmail.com> URL http://www.riazoshams.com/nlr/

References

Riazoshams H, Midi H, and Ghilagaber G, 2018,. Robust Nonlinear Regression, with Application using R, Joh Wiley and Sons.

See Also

nlout, nl.fitt

Examples

```
showClass("nl.robmeas")
```

---

nlmest.LM *Nonlinear MM-estimate using Levenberg-Marquardt algorithm.*

Description

Parameters estimates by robust MM-estimate by minimizing the sum of robust rho function, using Levenberg-Marquardt algorithm.

Usage

```
nlmest.LM(formula, data, start = getInitial(formula, data), robfunc,
control = nlr.control(tolerance = 0.001, minlanda = 1/2^10,
maxiter = 25 * length(start), robscale = T), vm = NULL, rm = eiginv(t(chol(vm))), .
```

Arguments

formula	nl.form object of the nonlinear function model. See nl.form object.
data	list of data with the response and predictor as name of variable. In heterogeneous case if it include response variable values of heterogenous variance function it asume variance function is function of predictor $H(x_i, \tau)$, otherwise it assume is a function of predictor $H(f(x_i, \theta), \tau)$.
start	list of starting value parameter, name of parameters must be represented as names of variable in the list.
robfunc	nl.form object of robust function used for downgrading.

control	nlr.control object, include tolerance, maxiter,... see nlr.control.
vm	optional covariance matrix of residuals, used for nonlinear generalized M-estimate.
rm	optional correlation matrix, used for nonlinear generalized M-estimate. rm is correlation matrix of vm, thus only vm is enough to be given. It can be given by user also but not necessary automatically will be calculated by argument eiginv(t(chol(vm))).
...	any other argument passed to formula, robfnc, or optimization function.

Details

This function is mixture of Levenberg Marquardt, Newton and Steepest descent, is derivative base optimization method. It is used to minimize the robust loss function using ρ function. This method is very fast and used for when the gradient of colaborating functions exists.

Due to wrong effect of outlier in creating singularity in hessian matrix the levenberg Marquardt is used to remedy the effect.

Value

result is object of nl.fitt.rob (nonlinear fitt robust) for homogeneous variance, and nl.fitt.rgn for heterogeneous and autocorrelated error (nonlinear fitt robust generalized), see nl.fitt.rgn object detail.

parameters	nonlinear regression parameter estimate of θ.
correlation	of fited model.
form	nl.form object of called nonlinear regression model.
response	computed response.
predictor	computed (right side of formula) at estimated parameter with gradient and hessian attributes.
curvature	list of curvatures, see curvature function.
history	matrix of convergence history, collumns include: convergence index, parameters, minimized objective function, convergence criterion values, or other values. These values will be used in plot function in ploting history.
method	fittmethod object of method used for fitt.
data	list of called data.
sourcefnc	Object of class "callorNULL" source function called for fitt.
Fault	Fault object of error, if no error Fault number = 0 will return back.
htheta	robust loss value including gradient and hessain attributes.
rho	computed robust rho function, including gradient and hessain attributes.
ri	estimated residuals, including gradient and hessain attributes.
curvrob	curvature
robform	nl.form object of robust loss rho function.

if vm is not NULL the nl.fitt.rgn include following extra slots:

vm	covariance matrix, diagonal of variance model predicted values.
rm	cholesky decomposition of vm.
gresponse	transformed of response by rm, include gradinet and hessian attributes.
gpredictor	transformed of predictor by rm, include gradinet and hessian attributes.

Note

starting values `start` it must contains initial value for 'sigma'.

The `nlmest.NLM` function is more developed form of this function, but still there are some situations that current function might be used.

`nlr` package for estimating robust MM, try to acheive the optimization convergence using `nlmest.NLM` or `nlmest.WF` or `nlmest.LM`, either of them that does not attain convergence other method might be used.

This function is called from `nlr`, for compatibility it is more efficient to be called by `nlr` than callind directly.

Author(s)

Hossein Riazoshams, Jan 2010. Email: <riazihosein@gmail.com> URL `http://www.riazoshams.com/nlr/`

References

Riazoshams H, Midi H, and Ghilagaber G, 2018,. Robust Nonlinear Regression, with Application using R, Joh Wiley and Sons.

See Also

`nlmest.WF`, `nlmest.NLM`, `nl.form`, `nl.fitt.rob`, `nl.fitt.rgn`, `nlr.control`

Examples

```
## The function is currently defined as
"nlmest.LM"
```

---

nlmest.NLM *Nonlinear MM-estimate.*

Description

MM-estimate of a nonlinear function, Using Mixture of Newton and Levenberg-Marquardt method. Parameters estimates by robust MM-estimate by minimizing the sum of robust rho function.

Usage

```
nlmest.NLM(formula, data, start = getInitial(formula, data), robfunc,
control = nlr.control(tolerance = 1e-04,
minlanda = 1/2^25, maxiter = 25 * length(start)), vm = NULL, rm = NULL, ...)
```

Arguments

formula	nl.form object of the nonlinear function model. See nl.form object.
data	list of data with the response and predictor as name of variable. In heterogeneous case if it include response variable values of heterogenous variance function it asume variance function is function of predictor $H(x_i, \tau)$, otherwise it assume is a function of predictor $H(f(x_i, \theta), \tau)$.
start	list of starting value parameter, name of parameters must be represented as names of variable in the list.
robfunc	nl.form object of robust function used for downgrading.
control	nlr.control object, include tolerance, maxiter,... see nlr.control.
vm	optional covariance matrix of residuals, used for nonlinear generalized M-estimate.
rm	optional correlation matrix, used for nonlinear generalized M-estimate. rm is correlation matrix of vm, thus only vm is enough to be given. It can be given by user also but not necessary automatically will be calculated by argument eiginv(t(chol(vm))).
...	any other argument passed to formula, robfnc, or optimization function.

Details

This function is mixture of Levenberg Marquardt, Newton and Steepest descent, is derivative base optimization method. It is used to minimize the robust loss function using ρ function. This method is very fast and used for when the gradient of colaborating functions exists.

Due to wrong effect of outlier in creating singularity in hessian matrix the levenberg Marquardt is used to remedy the effect. Morover for fast convergence when hessian is non singular Newton with Steepest descent is applied.

Value

result is object of nl.fitt.rob (nonlinear fitt robust) for homogeneous variance, and nl.fitt.rgn for heterogeneous and autocorrelated error (nonlinear fitt robust generalized), see nl.fitt.rgn object detail.

parameters	nonlinear regression parameter estimate of θ.
correlation	of fited model.
form	nl.form object of called nonlinear regression model.
response	computed response.
predictor	computed (right side of formula) at estimated parameter with gradient and hessian attributes.
curvature	list of curvatures, see curvature function.
history	matrix of convergence history, collumns include: convergence index, parameters, minimized objective function, convergence criterion values, or other values. These values will be used in plot function in ploting history.
method	fittmethod object of method used for fitt.
data	list of called data.

sourcefnc	Object of class "callorNULL" source function called for fitt.
Fault	Fault object of error, if no error Fault number = 0 will return back.
htheta	robust loss value including gradient and hessain attributes.
rho	computed robust rho function, including gradient and hessain attributes.
ri	estimated residuals, including gradient and hessain attributes.
curvrob	curvature
robform	nl.form object of robust loss rho function.

if vm is not NULL the nl.fitt.rgn include following extra slots:

vm	covariance matrix, diagonal of variance model predicted values.
rm	cholesky decomposition of vm.
gresponse	transformed of response by rm, include gradinet and hessian attributes.
gpredictor	transformed of predictor by rm, include gradinet and hessian attributes.

Note

starting values start it must contains initial value for 'sigma'.

nlr package for estimating robust MM, try to acheive the optimization convergence using nlmest.NLM or nlmest.WF or nlmest.LM, either of them that does not attain convergence other method might be used.

nlmest.NLM.sCase2 compute singularity case as optim.NLM optimization. Can be computed this case by using singularCase=2 argument in nlr.control.

nlmest.NLMf Compute the procedure with fixed scale value, the result is not MM-estimate.

This function is called from nlr, for compatibility it is more efficient to be called by nlr than callind directly.

Author(s)

Hossein Riazoshams, Jan 2010. Email: <riazihosein@gmail.com> URL http://www.riazoshams.com/nlr/

References

Riazoshams H, Midi H, and Ghilagaber G, 2018,. Robust Nonlinear Regression, with Application using R, Joh Wiley and Sons.

See Also

nlmest.LM, nlmest.WF, nl.form, nl.fitt.rob, nl.fitt.rgn, nlr.control

Examples

```
# functioned defined as
"nlmest.NLM"
```

`nlmest.NM` *Nonlinear MM-estimate, Nelder-Mead.*

Description

MM-estimate of a nonlinear function, Using Nelder Mead derivative free optimization method.

Usage

```
nlmest.NM(formula, data, start = getInitial(formula, data), robfunc,
control = nlr.control(tolerance = 1e-04,
minlanda = 1/2^25, maxiter = 100 * length(start), robscale = T),
vm = NULL, rm = eiginv(t(chol(vm))), delta = NULL, ...)
```

Arguments

`formula`	nl.form object of the nonlinear function model. See `nl.form` object.
`data`	list of data with the response and predictor as name of variable. In heterogeneous case if it include response variable values of heterogenous variance function it asume variance function is function of predictor $H(x_i, \tau)$, otherwise it assume is a function of predictor $H(f(x_i, \theta), \tau)$.
`start`	list of starting value parameter, name of parameters must be represented as names of variable in the list.
`robfunc`	nl.form object of robust function used for downgrading.
`control`	nlr.control object, include tolerance, maxiter,... see `nlr.control`.
`vm`	NULL, optional covariance matrix of residuals, used for nonlinear generalized M-estimate.
`rm`	optional correlation matrix, used for nonlinear generalized M-estimate. rm is correlation matrix of vm, thus only vm is enough to be given. It can be given by user also but not necessary automatically will be calculated by argument eiginv(t(chol(vm))).
`delta`	increament of Nelder Mead method, default will be calculated 10% of parameter values, in the case of nonconvergence it can be modified manually to acheive convergence.
`...`	any other argument passed to formula, robfnc, or optimization function.

Details

Nelder Mead is derivative free optimization method. It is used to minimize the robust loss function using ρ function. This method is very slow and sugest to use with a large maximum number of iterations.

The function `smptry2` Find next minimum point in Nelder-Mead algorithm. It used for internal usage might not be called by user directly.

Value

result is object of nl.fitt.rob (nonlinear fitt robust) for homogeneous variance, and nl.fitt.rgn for heterogeneous and autocorrelated error (nonlinear fitt robust generalized), see nl.fitt.rgn object detail.

`parameters`	nonlinear regression parameter estimate of θ.
`correlation`	of fited model.
`form`	`nl.form` object of called nonlinear regression model.
`response`	computed response.
`predictor`	computed (right side of formula) at estimated parameter with gradient and hessian attributes.
`curvature`	list of curvatures, see `curvature` function.
`history`	matrix of convergence history, collumns include: convergence index, parameters, minimized objective function, convergence criterion values, or other values. These values will be used in `plot` function in ploting history.
`method`	`fittmethod` object of method used for fitt.
`data`	list of called data.
`sourcefnc`	Object of class `"callorNULL"` source function called for fitt.
`Fault`	`Fault` object of error, if no error Fault number = 0 will return back.
`htheta`	robust loss value including gradient and hessain attributes.
`rho`	computed robust rho function, including gradient and hessain attributes.
`ri`	estimated residuals, including gradient and hessain attributes.
`curvrob`	curvature
`robform`	`nl.form` object of robust loss rho function.

if `vm` is not NULL the nl.fitt.rgn include following extra slots:

`vm`	covariance matrix, diagonal of variance model predicted values.
`rm`	cholesky decomposition of vm.
`gresponse`	transformed of response by rm, include gradinet and hessian attributes.
`gpredictor`	transformed of predictor by rm, include gradinet and hessian attributes.

Note

This is a slow algorithm, since "nlr" is designed for derivative based, when the gradient does not exist recomend to use this function. When the gradient exists it is strongly recomend to use derivative base methods.

This function is called from `nlr`, for compatibility it is more efficient to be called by `nlr` than callind directly.

Author(s)

Maria L. Rizzo

References

Statistical Computing with R, Maria L. Rizzo, 2008, Chopman & Hall/CRC

See Also

`nlmest.NLM`, `nl.form`

Examples

```
ntpstart=list(p1=.12,p2=6,p3=1,p4=33)
ntpstarttau=list(tau1=-.66,tau2=2,tau3=.04)
datalist=list(xr=ntp$dm.k,yr=ntp$cm.k)
datalist[[nlrobjvarmdls3[[2]]$independent]]<-ntp$dm.k
fittnml <- nlmest.NM(formula=nlrobj1[[15]], data = list(xr=ntp$dm.k,yr=ntp$cm.k), start=ntps
robscale = TRUE, robfunc = nl.robfuncs[["hampel"]],control=nlr.control(tolerance=1e-8,trace=
fittnml$parameters
```

---

nlmest.RWT *Nonlinear MM-estimate using reweighting method.*

Description

Compute MM-estimate using reweighting method developed by Stromberg.

Usage

```
nlmest.RWT(formula, data, start = getInitial(formula, data), robfunc,
control = nlr.control(tolerance = 0.001, minlanda = 1/2^25,
maxiter = 25 * length(start), trace = F), vm = NULL, rm = eiginv(t(chol(vm))), ...)
```

Arguments

`formula`	nl.form object of the nonlinear function model. See `nl.form` object.
`data`	list of data with the response and predictor as name of variable. In heterogeneous case if it include response variable values of heterogenous variance function it asume variance function is function of predictor $H(x_i, \tau)$, otherwise it assume is a function of predictor $H(f(x_i, \theta), \tau)$.
`start`	list of starting value parameter, name of parameters must be represented as names of variable in the list.
`robfunc`	nl.form object of robust function used for downgrading.
`vm`	optional covariance matrix of residuals, used for nonlinear generalized M-estimate.
`rm`	optional correlation matrix, used for nonlinear generalized M-estimate. rm is correlation matrix of vm, thus only vm is enough to be given. It can be given by user also but not necessary automatically will be calculated by argument eiginv(t(chol(vm))).
`...`	any other argument passed to formula, robfnc, or optimization function.
`control`	`nlr.control` option variables.

Details

Compute MM-estimate using reweighting method developed by Stromberg.

Value

result is object of nl.fitt.rob (nonlinear fitt robust) for homogeneous variance, and nl.fitt.rgn for heterogeneous (not developed yet) and autocorrelated error (nonlinear fitt robust generalized), see nl.fitt.rgn object detail.

Note

It is similar to `nlrob` function in `robustbase` package, it is not completly operational in `nlr`.

This function is called from `nlr`, for compatibility it is more efficient to be called by `nlr` than callind directly.

Author(s)

Hossein Riazoshams, Jan 2010. Email: <riazihosein@gmail.com> URL http://www.riazoshams.com/nlr/

References

Stromberg, A. J. (1993). Computation of High Breakdown Nonlinear Regression Parameters, Journal of American Statistical Association 88(421): 237-244.

See Also

`nlmest.NLM`

Examples

```
## The function is currently defined as
"nlmest.RWT"
```

---

nlmest.WF *Nonlinear MM-estimate using wolf conditions.*

Description

Parameters estimates by robust MM-estimate by minimizing the sum of robust rho function, Choosing a Step-Length using Wolfe Conditions.

Usage

```
nlmest.WF(formula, data, start = getInitial(formula, data), robfunc,
control = nlr.control(tolerance = 1e-04,
maxiter = 25 * length(start), robscale = T), vm = NULL, rm = eiginv(t(chol(vm))), .
# zoom2 is sub function
#zoom2(a1,a2,p1,p2,pd1,pd2,ht,phi0,phid0,theta1,
#delta1,sigma,objfnc,data,start,control,...)
```

Arguments

formula	nl.form object of the nonlinear function model. See `nl.form` object.
data	list of data with the response and predictor as name of variable. In heterogeneous case if it include response variable values of heterogenous variance function it asume variance function is function of predictor $H(x_i, \tau)$, otherwise it assume is a function of predictor $H(f(x_i, \theta), \tau)$.
start	list of starting value parameter, name of parameters must be represented as names of variable in the list.
robfunc	nl.form object of robust function used for downgrading.
control	nlr.control object, include tolerance, maxiter,... see `nlr.control`.
vm	optional covariance matrix of residuals, used for nonlinear generalized M-estimate.
rm	optional correlation matrix, used for nonlinear generalized M-estimate. rm is correlation matrix of vm, thus only vm is enough to be given. It can be given by user also but not necessary automatically will be calculated by argument eiginv(t(chol(vm))).
...	any other argument passed to formula, robfnc, or optimization function.

Details

This function Choosing a Step-Length using Wolfe Conditions in direct search optimization. Sum of robust loss function ρ is minimized in order to get the robust MM-estimates. This method is very fast and used for when the gradient of colaborating functions exists.

Value

result is object of nl.fitt.rob (nonlinear fitt robust) for homogeneous variance, and nl.fitt.rgn for heterogeneous and autocorrelated error (nonlinear fitt robust generalized), see nl.fitt.rgn object detail.

parameters	nonlinear regression parameter estimate of θ.
correlation	of fited model.
form	`nl.form` object of called nonlinear regression model.
response	computed response.
predictor	computed (right side of formula) at estimated parameter with gradient and hessian attributes.
curvature	list of curvatures, see `curvature` function.

history	matrix of convergence history, collumns include: convergence index, parameters, minimized objective function, convergence criterion values, or other values. These values will be used in plot function in ploting history.
method	fittmethod object of method used for fitt.
data	list of called data.
sourcefnc	Object of class "callorNULL" source function called for fitt.
Fault	Fault object of error, if no error Fault number = 0 will return back.
htheta	robust loss value including gradient and hessain attributes.
rho	computed robust rho function, including gradient and hessain attributes.
ri	estimated residuals, including gradient and hessain attributes.
curvrob	curvature
robform	nl.form object of robust loss rho function.

if vm is not NULL the nl.fitt.rgn include following extra slots:

vm	covariance matrix, diagonal of variance model predicted values.
rm	cholesky decomposition of vm.
gresponse	transformed of response by rm, include gradinet and hessian attributes.
gpredictor	transformed of predictor by rm, include gradinet and hessian attributes.

Note

The nlmest.NLM function is more developed form of this function, but still there are some situations that current function might be used. nlr package for estimating robust MM, try to acheive the optimization convergence using nlmest.NLM or nlmest.WF or nlmest.LM, either of them that does not attain convergence other method might be used.

Author(s)

Jorge Nocedal Stephen J. Wright

References

Numerical Optimization, Jorge Nocedal Stephen J. Wright, Springer 2006.

See Also

nlmest.LM, nlmest.NLM, nlmest.LM, nlmest.WF, nl.form, nl.fitt.rob, nl.fitt.rgn, nlr.control

Examples

```
## The function is currently defined as
"nlmest.WF"
```

`nlout` *Nonlinear outlier detection.*

Description

Detecting outlier for nonlinear regression, is based on mixing robust estimates and statsitics measures.

Usage

```
nlout(nlfited)
```

Arguments

`nlfited`	Object of type `nl.fitt` or `nl.fitt.gn` for classic estimators, `nl.fitt.rob` or `nl.fitt.rgn` for robust estimators.

Details

The outlier detection measutred used in this function are studentized residuals and Cook Distance. They are mixture of estimators and Jacobians. They are successful for detecting outlier only if combine with robust fits, eventhough the function can work with classic fits but it is not recomended.

Value

Result is list of `nl.robmeas` objects for each statistics.

`"vmat"`	variance covariance matrix of parameters $\sigma^2(\nabla f(\theta)'\nabla f(\theta))^{-1})$
`"d.yhat"`	predicted values after rremoving a point $\hat{y}_{(-i)}$
`"studres"`	`nl.robmeas` object Studentized residuals.
`"cook"`	`nl.robmeas` object od Elliptic Norm (Cook Dist)
`"mahd.v"`	`nl.robmeas` object of Regression Mahalanobis Distance.
`"mahd.dt"`	`nl.robmeas` object of Mahalanobis MVE, data.
`"mahd.xs"`	`nl.robmeas` object of Mahalanobis MVE, xs.
`"hadi"`	`nl.robmeas` object of Hadi potential.
`"potmah"`	`nl.robmeas` object of Potential mahalanobis.
`"delstud"`	`nl.robmeas` object of Deletion Studentized.
`"dffits"`	`nl.robmeas` object of DFFITS.
`"atk"`	`nl.robmeas` object of Atkinson Distance.
`"mvedta"`	`nl.robmeas` object of MVE data.
`"mvex"`	`nl.robmeas` object of MVE x.
`"dfbetas"`	`nl.robmeas` object of DFBETAS.

Note

This function return back all resutls and statistics but, Riazoshams (2009) showed studentized residuals and Cook distance when combine with robust estimators can detect outliers correctly. Thus to identify outlier correctly first estimate the parameters bu robust options of `nlr` function then call `nlout`, finally look at the list values `"studres"` and `"cook"` from the result list. The `plot` and other methods of `nl.robmeas` display the results visually.

Author(s)

Hossein Riazoshams, Dec 2008 Email: `<riazihosein@gmail.com>` URL `http://www.riazoshams.com/nlr/`

References

Riazoshams H, Habshah M and Adam MB 2009 On the outlier detection in nonlinear regression. 3(12), 243-250.

See Also

`nl.fitt`, `nl.fitt.gn`, `nl.fitt`, `nl.fitt.gn`, `nl.fitt.rob`, `nl.fitt.rgn`, `nl.robmeas`, `nlr`, `nlout.JL`

Examples

```
d<-list(xr=Weights$Date, yr=Weights$Weight)
wmodel <- nlr(nlrobj1[[2]],data=d,control=nlr.control(method = "OLS",trace=TRUE))
a=nlout(wmodel)
## Run the command as bellow
## nlout(wmodel)
```

`nlout.JL` *Nonlinear outlier detection.*

Description

Detecting outlier for nonlinear regression, is based on mixing statsitics measures and robust estimates through their covariance matrices (hat matrix). The covariance matrix in nonlinear is based on the gradient of nonlinear regression model, but it based on linear approximation of the model, instead Jacobian Leverage is used in this function.

Usage

```
nlout.JL(nlfited)
```

Arguments

`nlfited`	Object of type `nl.fitt` or `nl.fitt.gn` for classic estimators, `nl.fitt.rob` or `nl.fitt.rgn` for robust estimators.

Details

The outlier detection measutred used in this function are studentized residuals and Cook Distance. They are mixture of estimators and Jacobians. They are successful for detecting outlier only if combine with robust fits, eventhough the function can work with classic fits but it is not recomended. Riazoshams and Midi (2014)

Value

Result is list of `nl.robmeas` objects for each statistics.

`"jl.vmat"`	Jacobian-leverage matrix.
`"jl.studres"`	`nl.robmeas` object of Jacobian Leverage Studentised Residuals.
`"jl.cook"`	`nl.robmeas` object of Jacobian Leverage Elliptic Norm (Cook Dist).
`"jl.hadi"`	`nl.robmeas` object of Jacobian Leverage Hadi potential.
`"jl.delstud"`	`nl.robmeas` object of Jacobian Leverage Deletion Studentized.
`"jl.dffits"`	`nl.robmeas` object of Jacobian Leverage DFFITS.
`"jl.atk"`	`nl.robmeas` object ofJacobian Leverage Atkinson Distance.

Note

This function return back all resutls and statistics based on Jacobian leverage, but Riazoshams (2014) showed studentized residuals when combine with robust estimators can detect outliers correctly. Thus to identify outlier correctly first estimate the parameters bu robust options of `nlr` function then call `nlout`, finally look at the list values `"jl.delstud"` from the result list. The `plot` and other methods of `nl.robmeas` display the results visually.

Author(s)

Hossein Riazoshams, Jan 2010. Email: `<riazihosein@gmail.com>` URL `http://www.riazoshams.com/nlr/`

References

Riazoshams H, Habshah M and Adam MB 2009 On the outlier detection in nonlinear regression. 3(12), 243-250.

Riazoshams H and Midi H 2014 Robust Leverage and outlier detection measures in nonlienar regression, 2014 (Unpublished manuscript).

See Also

`nl.fitt`, `nl.fitt.gn`, `nl.fitt`, `nl.fitt.gn`, `nl.fitt.rob`, `nl.fitt.rgn`, `nl.robmeas`, `nlr`, `nlout`

Examples

```
d<-list(xr=Weights$Date, yr=Weights$Weight)
wmodel <- nlr(nlrobj1[[2]],data=d,control=nlr.control(method = "OLS",trace=TRUE))
a=nlout.JL(wmodel)
plot(a[[2]])
```

nlr *Non-Linear Robust fitt.*

Description

This is generic function fits a nonlinear mixed-effects model using robust methods described in Riazoshams et al (2015), allowed to be correlated and/or have unequal variances.

Usage

```
nlr(formula, data = parent.frame(), start = getInitial(formula, data),
control = nlr.control(minlanda=1 / 2 ^ 10,
maxiter=25 * length(start)), weights = NULL, robustobj = NULL,
robustform = c("hampel", "huber", "bisquare",
"andrew", "halph huber", "hampel 2", "least square"),
varianceform = NULL, tau = NULL, correlation = NULL,
covariance = NULL, ...)
```

Arguments

formula	can be a nl.form object of the nonlinear function model, or a model formula with the response on the left of a ~ operator and an expression involving parameters and covariates on the right. See nl.form object.
data	an optional data frame or list of data with the response and predictor as name of variable. In heterogeneous case if it include response variable values of heterogenous variance function it asume variance function is function of predictor $H(x_i, \tau)$, otherwise it assume is a function of predictor $H(f(x_i, \theta), \tau)$.
start	list of starting value parameter, name of parameters must be represented as names of variable in the list.
control	nlr.control, include controles for nlr, see nlr.control.
weights	User optional matrix of variance covariance matrix of error, a general weight that can be used by user, but if correlation, covariace or varianceform argument given, then will be ignored.
robustobj	=NULL, optional nl.form object of robust loss function defined by user. User can define his/her own function, to be nl.form object, Or use defined robloss function in robustform.
robustform	nl.form object of robust function used for downgrading. functions("hampel","huber","bisquare","andrew" huber","hampel 2","least square").
varianceform	NULL, nl.form object if given heterogeneous variance will be fited.
tau	NULL, list or data.frame of initial values for heterogeneous variance function parameter. the stored value in vardnc object of nl.form will be stored.
correlation	autocorrelated error, form of corStruct but not in nlme, is extra arguments to any of "nl.forms".

covariance	optional covariance matrix of errors. If given Generalized estimates will be calculated.
...	Any extra arguments to any function sources, such as nonlinear regression model, heteroscedastic variance function, robust loss function or optimization object function.

Details

nlr is non-linear robust inference. It is mostly gradient and hessian based. The classic estimates also is avilable.

Value

Output depends on the method called as:

- Nonlinear Least Square Estimate(NLLS): `nl.fitt` object.
- Generalized NLLS: `nl.fitt.gn` object.
- Robust Estimate: `nl.fitt.rob` object.
- Generalized Robust Estimate: `nl.fitt.rgn` object.

Note

Thic is the global function of `nlr` package. You can fitt models both with classic and robust methods. Besides homogeneous variance of error, heteroscedastic and autocorrelated error can be fited by this function.

`nlr` is optimized for derivative based computation, but deribative free methods are provided for nonlinear regressions that derivative does not exist. In contrast derivative free method in this package are slow.

This package requires `tseries` package to be installed, under ubuntu if facing with error that can not install "curl" ot "libcurl" package, run these command in terminal: sudo apt-get install libcurl4-openssl-dev

deb http://security.ubuntu.com/ubuntu/ precise-security restricted main multiverse universe deb http://us.archive.ubuntu.com/
precise-updates restricted main multiverse universe

sudo apt-get update sudo apt-get install libcurl4-gnutls-dev

after all we can install tseries in R.

Author(s)

Hossein Riazoshams, May 2014. Email: <riazihosein@gmail.com> URL `http://www.riazoshams.com/nlr/`

References

Riazoshams H, Midi H, and Ghilagaber G, 2018,. Robust Nonlinear Regression, with Application using R, Joh Wiley and Sons.

See Also

nl.form, nl.fitt, nl.fitt.gn, nl.fitt.rob, nl.fitt.rgn

Examples

```
## Chicken fitt without hetroscedasticity
 d<-list(xr=Weights$Date, yr=Weights$Weight)
 fitt.chicken1 <- nlr(nlrobj1[[2]],data=d,control=nlr.control(method = "OLS",trace=TRUE))
fitt.chicken1$parameters
plot(fitt.chicken1)
## Chicken fitt with hetroscedasticity
# RME by default
fitt.chicken2<-nlr(formula=nlrobj1
                   [[14]],data=d,start=list(p1=2300,p2=42,p3=.11),
                   robustform ="hampel",
                   tau=list
                   (sg=.09,landa=1),varianceform=nlrobjvarmdls1[[1]],
                   control=nlr.control(tolerance=1e-3))
fitt.chicken2$parameters
fitt.chicken2$hetro$parameters
plot(fitt.chicken2)
# autocorrelated case
xr = trade.ir[, 1]
yr = trade.ir[, 2]
a1 <-  nlr( nlrobj5[[18]],data=list(xr = xr, yr = yr),
correlation=list(StructName="corAR1"))
a1$parameters
plot(a1)
```

---

nlr.control *list of nlr package controls.*

Description

create nlr.control options for using in diferent part of the package.

Usage

```
nlr.control(maxiter = 50, tolerance = 0.0001, minscale = 0.001, trace = F,
minlanda=1e-16,derivfree=F,robscale=T,
algorithm = c("Levenberg-Marquardt", "Nelder-Mead","Gauss Newton"),
method=c("default","RME","CME","CLSME","RGME","WME","MLE","OLS","TS","RTS","lms"),
initials=c("manuall","lms","OLS","quantile"),history=F,length.out=NULL,singlePlot=F
singularCase=1,
JacobianLeverage = c("default", "classic", "robust"))
```

Arguments

`maxiter`	maximum number of iteration. To be used optimization or fitt procedures.
`tolerance`	Tolerance of convergence.
`minscale`	Minimum of scale value.
`trace`	Default is False, if True draw the graph of convergence in iterations, depends on availability for that cases.
`minlanda`	Minimum of landa coeficient values in stepest descend or levenberg marquardt. When the step length does not reduce the objective function this value decrease.
`derivfree`	Default value is False. If True the implicitly program use derivative free optimization methods based on Nelder-Mead method. This is very slow convergence in that case the number of maximum iteration should be increased, at least maxiter=500 is sugested. nlr is derivative based but in the cases where problem happenes in derivative computation Nelder-Mead will be used.
`robscale`	Default is True, when False the clasic standard error of residuals will be used in computations. It is not recomneded to use this option because reduce the robustness of estimators.
`algorithm`	of optimization method, default is "Levenberg-Marquardt", is robust in computation when outlier happense. "Nelder-Mead" is derivative free method, and "Gauss Newton" method is used for fast computation but might face with some singularity in hessian when outlier happense.
`singularCase`	Select how to solve the singular gradient matrix case in MM-estimate procedure of levenberg marquard method. The efault value equal 1 add up a value to diagonal elements, and 2 add up proportion to size of diagonal values. They are same but Generally case 1 works in most of cases, but in case of divergence or singularity problem can use 2.
`JacobianLeverage`	character name of jacobian leverage to be used incomputation. • "default":default value assigned by any function • "classic":cllasic nonrobust value • "robust":robust jacobian leverage value
`method`	of computation using several type of estimators. • "RME": Heteroscedastic error Robust Multi Stage Estimate. • "CME": Heteroscedastic error Classic Multi Stage Estimate. • "CLSME": Heteroscedastic error Classic Least Square based Estimate. • "RGME": Heteroscedastic error Robust Generalized Estimate. • "WME": Heteroscedastic error Weighted M-Estimate, See Lim 2010. • "MLE": Maximum Likelihood Estimate. • "OLS": Ordinary Least square, constant and uncorrelated error. • "TS": Autocorelated error, classic Two Stage Estimate. • "RTS": Autocorelated error, Robust Two Stage Estimate. • "lms": Least median of squared residuals estimate. (Non efficient)
`initials`	is used to define initializing parameters using a specific estimatro. Might be used when initial values are dificult to find manually, or getInitial function.

- "manuall" default value is manually provided initial values by user.
- "lms" robust high breakdown point least median of squares of errors estimator.
- "ols" ordinary least square estimator.
- "quantile" robust least quantile regression.

`length.out` Length for incrementing independent variable to be used in ploting commands to have a more smoother curve.

`history` Default=F, Used in plot command, to draw the history of convergence, default is F, the TRUE value force the function to draw the history.

`singlePlot` Default=F, By default the plot function draw the fited model (in one dimentional case), and residuals in second collumn, singlePlot=T causes the plot to draw the two graph in two windows.

Details

nlr.control function create nlr.control variable and mostly used as control argument to the functions. It define convergence and many other parameters depends on the function operation.

Value

list of controls.

Note

"nlr" package is derivative based, the default value for algorithm for optimization is derivative based methods such as Levenberg-Marquardt, then nlr check use optional user request. Beside some options for ploting and further more can be defined by `nlr.control`. This control passe to functions for controling computations, by users.

Author(s)

Hossein Riazoshams, Apr 2014. Email: <riazihosein@gmail.com> URL `http://www.riazoshams.com/nlr/`

References

Riazoshams H, Midi H, and Ghilagaber G, 2018,. Robust Nonlinear Regression, with Application using R, Joh Wiley and Sons.

See Also

`nlr`

Examples

```
## The function is currently defined as
"nlr.control"
nlr.control()
```

`nlrobj1` *Nonlinear model objects*

Description

List of `nl.form` objects of nonlinear regression models.

- nlrobj1: has 16 model.
- nlrobj3: has 18 model.
- nlrobj4: has 11 model.
- nlrobj5: has 19 model.
- nlrobj6: has 7 model.
- nlrobj7: has 23 model.

Usage

```
nlrobj1
```

Format

The format is: chr "nlrobj1"

Elements are list of model, each cell is `nl.form` object.

Details

by index can access any of the nodels, for example nlrobj1[[14]] with name nlrobj1[[14]]$name="Logistic without intercept" is logestic model to fitt the chicken `Weights` data.

Note

The variables with prefix `fkt` are created by Bunke et al. They will not be used in `nlr` directly or by user.

Source

("nlrobj1", "nlrobj3", "nlrobj4", "nlrobj5") are generalization of objects developed by Bunke et al. 1998.

References

Bunke, O., Droge, B., Polzehl, J. Splus tools for model selection in nonlinear regression (1998) Computational Statistics, 13 (2), pp. 257-281.

See Also

`nl.form`, `Weights`

Examples

```
data(nlrobj1)
nlrobj1
```

`nlrobjvarmdls1` *Variance model objects.*

Description

List of `nl.form` objects of pre defined variance models. They use in heteroscedastic variance cases.

- nlrobjvarmdls1 has 7 nonlinear variance model function, with σ^2 as variance.
- nlrobjvarmdls1 has 7 nonlinear variance model function exactly same as `nlrobjvarmdls1` but the σ as variance. In fact it can be standard deviation, square roots. But `nlr` work with variance functions.
- nlrobjvarmdls1 has 3 nonlinear variance model function, in general case they dont include constant variance σ. Variance is general parameteric form $Var(error) = H(x, \tau)$

Usage

```
data(nlrobjvarmdls1)

#nlrobjvarmdls1[[1]] access first element.
```

Format

The format is: chr "nlrobjvarmdls1"

Elements are list of model, each cell is `nl.form` object.

Details

by index can access any of the nodels, for example nlrobj1[[14]] with name nlrobjvarmdls1[[1]]$name="power" is power model used to fitt the chicken `Weights` data.

Source

Riazoshams 2015

References

Robust Nonlinear Regression, Theories and Methods with Practical Guides for R Packages. Riazoshams et al.

See Also

`nl.form`, `Weights`

Examples

```
data(nlrobjvarmdls1)
nlrobjvarmdls1
```

`nlsnm` *Least Square estimate.*

Description

Least Square estimate of a nonlinear function, Using Nelder Mead derivative free optimization method.

Usage

```
nlsnm(formula, data, start = getInitial(formula, data), delta = NULL,
control = nlr.control(tolerance = 1e-04,
maxiter = 100 * length(start)), vm = NULL, rm = NULL, ...)
```

Arguments

`formula`	nl.form object of the nonlinear function model. See `nl.form` object.
`data`	list of data with the response and predictor as name of variable.
`start`	list of starting value parameter, name of parameters must be represented as names of variable in the list.
`control`	nlr.control object, include tolerance, maxiter,... see `nlr.control`.
`vm`	NULL, optional covariance matrix of residuals, used for nonlinear generalized M-estimate.
`rm`	optional correlation matrix, used for nonlinear generalized M-estimate. rm is correlation matrix of vm, thus only vm is enough to be given. It can be given by user also but not necessary automatically will be calculated by argument eiginv(t(chol(vm))).
`delta`	increament of Nelder Mead method, default will be calculated 10% of parameter values, in the case of nonconvergence it can be modified manually to acheive convergence.
`...`	any other argument passed to formula, robfnc, or optimization function.

Details

Nelder Mead is derivative free optimization method. It is used to minimize the square loss function. This method is very slow and sugest to use with a large maximum number of iterations.

Value

result is object of `nl.fitt` (nonlinear fitt robust) for homogeneous variance, and `nl.fitt.gn` for generalized fitt when covariance or correlation matrix is given.

`parameters`	nonlinear regression parameter estimate of θ.
`correlation`	of fited model.
`form`	`nl.form` object of called nonlinear regression model.
`response`	computed response.
`predictor`	computed (right side of formula) at estimated parameter with gradient and hessian attributes.
`curvature`	list of curvatures, see `curvature` function.
`history`	matrix of convergence history, collumns include: convergence index, parameters, minimized objective function, convergence criterion values, or other values. These values will be used in `plot` function in ploting history.
`method`	`fittmethod` object of method used for fitt.
`data`	list of called data.
`sourcefnc`	Object of class `"callorNULL"` source function called for fitt.
`Fault`	`Fault` object of error, if no error Fault number = 0 will return back.

if `vm` is not NULL the nl.fitt.rgn include following extra slots:

`vm`	covariance matrix, diagonal of variance model predicted values.
`rm`	cholesky decomposition of vm.
`gresponse`	transformed of response by rm, include gradinet and hessian attributes.
`gpredictor`	transformed of predictor by rm, include gradinet and hessian attributes.

Note

This is a slow algorithm, since "nlr" is designed for derivative based, when the gradient does not exist recomend to use this function. When the gradient exists it is strongly recomend to use derivative base methods.

This function call by `nlr`, for compatibility it is better to call from `nlr` rather than directly by user.

Author(s)

Maria L. Rizzo

References

Statistical Computing with R, Maria L. Rizzo, 2008, Chopman & Hall/CRC

See Also

`nlmest.NLM`, `nl.form`, `nlsqr`

Examples

```
## The function is currently defined as
"nlsnm"
```

`nlsqr` *Least Square estimate.*

Description

Least Square estimate of a nonlinear function, Using QR-decomposition of Gradient matrix.

Usage

```
nlsqr(formula, data, start = getInitial(formula, data),
control = nlr.control(tolerance = 1e-04, minlanda = 1/2^10,
maxiter = 25 * length(start)))
```

Arguments

`formula`	nl.form object of the nonlinear function model. See `nl.form` object.
`data`	list of data with the response and predictor as name of variable.
`start`	list of starting value parameter, name of parameters must be represented as names of variable in the list.
`control`	nlr.control object, include tolerance, maxiter,... see `nlr.control`.

Details

It is used to minimize the square loss function, using QR-decomposition of gradient matrix, thus the nonlinear function model `formula` must return back Gradient.

Value

result is object of `nl.fitt` (nonlinear fitt robust) for homogeneous and uncorrelated variance.

`parameters`	nonlinear regression parameter estimate of θ.
`correlation`	of fited model.
`form`	`nl.form` object of called nonlinear regression model.
`response`	computed response.
`predictor`	computed (right side of formula) at estimated parameter with gradient and hessian attributes.
`curvature`	list of curvatures, see `curvature` function.
`history`	matrix of convergence history, collumns include: convergence index, parameters, minimized objective function, convergence criterion values, or other values. These values will be used in `plot` function in ploting history.

method	fittmethod object of method used for fitt.
data	list of called data.
sourcefnc	Object of class "callorNULL" source function called for fitt.
Fault	Fault object of error, if no error Fault number = 0 will return back.

Note

This function is fast algorithm based on gradient. If gradient does not exist one can use nlsnm function.

This function call by nlr, for compatibility it is better to call from nlr rather than directly by user.

Author(s)

Hossein Riazoshams, Jan 2010. Email: <riazihosein@gmail.com> URL http://www.riazoshams.com/nlr/

References

Bates, D. M., and Watts, D. G. (1988). Nonlinear regression analysis and its applications. New York: John Wiley & Sons.

See Also

nl.form, nlsnm, nlr.control, nl.fitt, curvature, Fault

Examples

```
## The function is currently defined as
"nlsqr"
```

---

nlsqr.gn *Generalized Least Square estimate.*

Description

Generalized Least Square estimate of a nonlinear function, Using QR-decomposition of Gradient matrix.

Usage

```
nlsqr.gn(formula, data, start = getInitial(formula, data),
control = nlr.control(tolerance = 0.001,
minlanda = 1/2^10, maxiter = 25 * length(start)), vm, rm = eiginv(t(chol(vm))))
```

Arguments

formula	nl.form object of the nonlinear function model. See nl.form object.
data	list of data with the response and predictor as name of variable.
start	list of starting value parameter, name of parameters must be represented as names of variable in the list.
control	nlr.control object, include tolerance, maxiter,... see nlr.control.
vm	Covariance matrix of residuals, used for nonlinear generalized M-estimate.
rm	optional correlation matrix, used for nonlinear generalized M-estimate. rm is correlation matrix of vm, thus only vm is enough to be given. It can be given by user also but not necessary automatically will be calculated by argument eiginv(t(chol(vm))).

Details

It is used to minimize the square loss function, using QR-decomposition of gradient matrix, thus the nonlinear function model formula must return back Gradient. nlsqr.gn work with a general variance covariance matrix, such as heteroscedastic or weights in variance, and partially autocorrelated with any general format.

Value

result is object of nl.fitt.gn for generalized fitt when covariance or correlation matrix is given.

parameters	nonlinear regression parameter estimate of θ.
correlation	of fited model.
form	nl.form object of called nonlinear regression model.
response	computed response.
predictor	computed (right side of formula) at estimated parameter with gradient and hessian attributes.
curvature	list of curvatures, see curvature function.
history	matrix of convergence history, collumns include: convergence index, parameters, minimized objective function, convergence criterion values, or other values. These values will be used in plot function in ploting history.
method	fittmethod object of method used for fitt.
data	list of called data.
sourcefnc	Object of class "callorNULL" source function called for fitt.
Fault	Fault object of error, if no error Fault number = 0 will return back.
vm	covariance matrix, diagonal of variance model predicted values.
rm	cholesky decomposition of vm.
gresponse	transformed of response by rm, include gradinet and hessian attributes.
gpredictor	transformed of predictor by rm, include gradinet and hessian attributes.

Note

This is a generalized form of `nlsqr` function. It is fast algorithm based on gradient. If gradient does not exist one can use `nlsnm` function.

This function call by `nlr`, for compatibility it is better to call from `nlr` rather than directly by user.

Author(s)

Hossein Riazoshams, Jan 2010. Email: <riazihosein@gmail.com> URL http://www.riazoshams.com/nlr/

References

Seber, G., A. F. and Wild, C. J. (2003). Nonlinear Regression. New York: John Wiley & Sons, Inc.

See Also

`nl.form`, `nlsnm`, `nlr.control`, `nl.fitt`, `curvature`, `Fault`,`nlsqr`

Examples

```
## The function is currently defined as
"nlsqr.gn"
```

---

nonrepl *Sample variance of response.*

Description

Compute the sample standard deviation of response data$y at repeated predictor data$x.

Usage

```
nonrepl(data)
```

Arguments

`data`	list of response data$y and predictor data$x.

Details

If predictor $x(i)$ repeated n_i times, the sample variance of response is compute for $y(i,j), j = 1, ..., n_i$.

Value

list of result include standard deviations and information about repeated data as:

x	ordered data predictor.
y	ordered data response by predictor.
xk	nonrepeated data of x's.
ni	number of repeated of each xk's.
xm	position of each xk's in ordered x.
k	length of non repeated data, xk.
xo	
yq	
ys	

Author(s)

Bunke et al. 1998.

References

This function provided by Bunke, O., Droge, B., Polzehl, J. Splus tools for model selection in nonlinear regression (1998) Computational Statistics, 13 (2), pp. 257-281.

See Also

zvalues, rzvalues

Examples

```
## The function is currently defined as
"nonrepl"
```

---

ntp *ntp data*

Description

Cromium concentration in blood and kidney of Mouse, Rat, and Guinea pig.

Usage

```
data(ntp)
```

Format

The format is:data.frame chr "ntp"

- dm.k, cm.k, Dose (x) and Chromium Concentration (y) in Mouse kidney.
- dr.b, cr.b, Dose (x) and Chromium Concentration (y) Rat blood.
- dr.k, cr.k, Dose (x) and Chromium Concentration (y) Rat kidney.
- dp.b, cp.b, Dose (x) and Chromium Concentration (y) Guinea pig blood.
- dp.k, cp.k, Dose (x) and Chromium Concentration (y) in Guinea pig kidney.

Details

Lim et al. (2010) illustrated the Weighted M-Estimate (WME) methodology with real data from National Toxicology study Program (NTP 2007. pp. 11-12). NTP (2007) conducted 3-month and 2-year studies, where rodents were exposed to CrVI administered in drinking water as sodium dichromate dihydrate.The dose concentrations were 0, 2.87, 8.62, 28.7, 86.2, 287, and 862mg sodium dichromate dihyrate/L (to yield 0, 1, 3, 10, 30, 100, and 300 mg chromium/L). When animals were sacrificed, total chromium concentrations in blood, kidneys, and femurs were measured. Lim et al. (2010) proposed the hill model for the data

$$\theta_0 + \frac{\theta_1 x_i^{\theta_2}}{\theta_3^{\theta_2} + x_i^{\theta_2}}$$

with heterogeneous standard deviation

$$\tau_0 + \frac{\tau_1}{1 + e^{-\tau_2 x_i}}$$

Source

National Toxicology Program, 2007. NTP Toxicity Studies of Sodium Dichromate Dihydrate (CAS No. 7789-12-0) Administered in Drinking Water to Male and Female F344/N Rats and B6C3F1 Mice and Male BALB/c and am3-C57BL/6 Mice. Toxicity Report Series 72, 1-G4, U.S. Department of Health and Human Services, Public Health Service, National Institutes of Health, RTP, North Carolina, U.S.A.

References

Lim, C., Sen, P. K., Peddada, S. D. (2010). Statistical inference in nonlinear regression under heteroscedasticity. Sankhya B 72:202-218.

Examples

```
data(ntp)
## maybe strnt ntp
```

`optim.NLM` *NLM optimization.*

Description

Modified Newton-Levenberg-Marquardt optimization. It is derivative based optimization method, designed to be robust against sigularity problem due to outliers.

Usage

```
optim.NLM(objfnc, data, start = getInitial(objfnc, data),
control = nlr.control(tolerance = 0.001, minlanda = 1/2^10,
maxiter = 25 * length(start)), ...)
```

Arguments

`objfnc`	any objective function for minimizing, it must contains accept formula, data and start as argument, extra argument can be passed by (...). The output of objfnc must be a list contains: $value(attr,gradient,hessian), $angmat (angular matrix),$angvec (angular vector) to check convergence. Usually it might have `nl.form` object as entry.
`data`	list of the data, that might have predictor and response variables with names.
`start`	list of initial values with names as parameters.
`control`	nlr.control options to control the optimization iterations.
`...`	any external parameters passe to `objfnc`.

Details

Optimize objective function `objfnc` with respect to parameters `start`. The mothod is gradient base combines Newton, Stepest descend and levenberg-Marquardt.

Value

result is a list of:

`parameters`	list of estimated parameters wit hsame names as `start`
`objfnc`	computed object function returned back by `objfnc`
`history`	history of fitt, include parameters and objective values, other level of iteration is presented for which in each iteration some more steps is done to rectify the singularity of hessian.

Note

User can define his own optimization function `objfnc` for any purpose, but this function designed eficiently for robust estimates. It is applied for minimizing several kind of objective functions such as heteroscedastic chi-square likelihood, robust loss, but for other general problem usage is not tested.

This function call by `nlr`, for compatibility it is better to call from `nlr` rather than directly by user. User can use it for optimization purposes.

Author(s)

Hossein Riazoshams, May 2014. Email: <riazihosein@gmail.com> URL `http://www.riazoshams.com/nlr/`

References

Riazoshams H, Midi H, and Ghilagaber G, 2018,. Robust Nonlinear Regression, with Application using R, Joh Wiley and Sons. Seber, G., A. F. and Wild, C. J. (2003). Nonlinear Regression. New York: John Wiley & Sons, Inc.

See Also

`nl.form`

Examples

```
## The function is currently defined as
"optim.NLM"
```

`optim.NM` *NM optimization*

Description

Nelder-Mead derivative free optimization. Since it is derivative free have slow convergence.

Usage

```
optim.NM(objfnc, data, start = getInitial(objfnc, data), delta = NULL,deltar=.1,
control=nlr.control(tolerance=1e-4,
maxiter=250 * length(start)), ...)
```

Arguments

objfnc	any objective function for minimizing, it must contains accept formula, data and start as argument, extra argument can be passed by (...). The output of objfnc must be a list contains: $value(attr,gradient,hessian), $angmat (angular matrix),$angvec (angular vector) to check convergence. Usually it might have nl.form object as entry.
data	list of the data, that might have predictor and response variables with names.
deltar	=0.1. Ratio of delta value. "nlr" compute the increament interval of golden section by start +- delta, the delta by default computed by deltar*start. The user can give direct value for deltar or value of delta to be unequally ratio movement.
start	list of initial values with names as parameters.
delta	vector with same size of parameters named as parameter names, is increament for each parameter at the begining. Each parameter will be moved by delta.
control	nlr.control options to control the optimization iterations.
...	any external parameters passe to objfnc.

Details

Optimize objective function objfnc with respect to parameters start. The mothod is derivative free using Nelder-Mead method.

The function smptry Find next minimum point in Nelder-Mead algorithm. It used for internal usage might not be called by user directly.

Value

result is a list of:

parameters	list of estimated parameters wit hsame names as start
objfnc	computed object function returned back by objfnc
history	history of fitt, include parameters and objective values, other level of iteration is presented for which in each iteration some more steps is done to rectify the singularity of hessian.

Note

User can define his own optimization function objfnc for any purpose, but this function designed eficiently for robust estimates. It is applied for minimizing several kind of objective functions such as heteroscedastic, chi-square likelihood, robust loss, but for other general problem usage is not tested. Since Nelder-Mead is derivative free its convergence is slow, so it is suggested to use more maximum number of iteration option in nlr.control. optim.NLM is more efficient method use derivative values, but when the derivative does not exist nelder-Mead can be used.

This function call by nlr, for compatibility it is better to call from nlr rather than directly by user. User can use it for optimization purposes.

Author(s)

Hossein Riazoshams, May 2014. Email: <riazihosein@gmail.com> URL http://www.riazoshams.com/nlr/

References

Rizo ML 2008 Statistical Computing with R The R Series. Chapman & Hall/CRC The R Series.

See Also

nlr.control, nl.form, optim.NLM

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--or do  help(data=index)  for the standard data sets.

## The function is currently defined as
"optim.NM"
```

---

optim.WF *WF optimization*

Description

Optimization using Wolf conditions.

Usage

```
optim.WF(objfnc, data, start = getInitial(objfnc, data),
control = nlr.control(tolerance = 0.001, minlanda = 1/2^10,
maxiter = 25 * length(start)), ...)
```

Arguments

objfnc	any objective function for minimizing, it must contains accept formula, data and start as argument, extra argument can be passed by (...). The output of objfnc must be a list contains: $value(attr,gradient,hessian), $angmat (angular matrix),$angvec (angular vector) to check convergence. Usually it might have nl.form object as entry.
data	list of the data, that might have predictor and response variables with names.
start	list of initial values with names as parameters.
control	nlr.control options to control the optimization iterations.
...	any external parameters passe to objfnc.

Details

Optimize objective function `objfnc` with respect to parameters `start`. The mothod is gradient base using Wolf condition for rectifying the negative definit hessian problems.

The following function are called from `optim.WF`.

`CubInrep`: Cubic interpolation,

Value

result is a list of:

`parameters`	list of estimated parameters wit hsame names as `start`
`objfnc`	computed object function returned back by `objfnc`
`history`	history of fitt, include parameters and objective values, other level of iteration is presented for which in each iteration some more steps is done to rectify the singularity of hessian.

Note

User can define his own optimization function `objfnc` for any purpose, but this function designed eficiently for robust estimates. It is applied for minimizing several kind of objective functions such as heteroscedastic chi-square likelihood, robust loss, but for other general problem usage is not tested.

Together with `optim.NLM` are used in `nlr` package to acheive optimization result in failure of one another method.

This function call by `nlr`, for compatibility it is better to call from `nlr` rather than directly by user. User can use it for optimization purposes.

Author(s)

Hossein Riazoshams, May 2014. Email: `<riazihosein@gmail.com>` URL `http://www.riazoshams.com/nlr/`

References

Nocedal J and Wright SJ 2006 Numerical optimization.. New York, NY.

See Also

`nl.form`, `optim.NLM`

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--or do  help(data=index)  for the standard data sets.

## The function is currently defined as
"optim.WF"
```

`parameter.names` *Support for Functions* `nlr()`

Description

This is support for the functions ms() and nls(). It is not intended to be called directly by users.

Usage

```
parameter.names(formula, data)
```

Arguments

`formula`	Expression include variables.
`data`	data might have parameters attribute.

Details

In several functions, If start is not given the name of data and variables have to be gathered from formula and environment variables, which use this function appropriately.

Value

list of parameters.

Note

For internal use, muight not be called by user.

Author(s)

Venables, W.N., and Ripley

References

Venables, W.N., and Ripley, B.D. (1999). Modern Applied Statistics with S-PLUS. New York: Springer-Verlag

See Also

`nlmest.NLM`

Examples

```
## The function is currently defined as
"parameter.names"
```

`parInfer.WM` *WM-estimate Inference*

Description

Parameter inference for weighted M-estimate. WM-estimate is based on minimizing the robustified form of likelihood, simultanously over nonlinear function parameter and variance model parameters the the covariance of parameter, the estimate is assymptotically normal (See Lim et al.2010) with given covariance matric which compute for sample by the function.

Usage

```
parInfer.WM(object, confidence = 0.95)
```

Arguments

`object`	`nl.fitt.rgn` object of WM-fitt generated by `nl.robhetroWM` function.
`confidence`	Confidence probability.

Details

Compute covariance matrix and confidence interval for nonlinear model function parameter and nonlinear variance model parameters.

Value

`covmat:`	Covariance matrix of nonlinear model function parameters.
`covtau`	Covariance matrix of nonlinear variance model parameters.
`parstdev`	Standard deviation of nonlinear model function parameter. It is square root of diagonal of `covmat`.
`CI`	Confidence interval for nonlinear model function parameter.

Note

`ParInfer` method of `nl.fitt.rgn` call this function automatically, so user might not call it directly.

This function call by `nlr`, for compatibility it is better to call from `nlr` rather than directly by user.

Author(s)

Hossein Riazoshams, May 2014. Email: <riazihosein@gmail.com> URL `http://www.riazoshams.com/nlr/`

References

Lim, C., Sen, P. K., Peddada, S. D. (2010). Statistical inference in nonlinear regression under heteroscedasticity. Sankhya B 72:202-218.

See Also

nl.fitt.rgn, nl.robhetroWM

Examples

```
## The function is currently defined as
"parInfer.WM"
```

---

pInf *Parameter Inference for classic nonlinear regression.*

Description

Parameter inference for classic nonliner regression. It work same as parInfer method of nl.fitt, calculate covariance matrix of parameters and their confidence interval using gradient as design matrix.

Usage

```
pInf(object, confidence = 0.95)
```

Arguments

object	Object type nl.fitt or any other of its child objects such as nl.fitt.gn, nl.fitt.rob, nl.fitt.rgn.
confidence	Confidence probability.

Details

For computing the covariance matrix of a nonlinear regression parameter, the gradient of function with respect to parameters is consider as design matrix and linear regression formulas apply for computing covariances and confidence intervals.

Value

covmat:	Covariance matrix of nonlinear model function parameters.
corrmat	Correlation matrix of nonlinear model function parameters.
parstdev:	Standard deviation of nonlinear model function parameter. It is square root of diagonal of covmat.
CI:	Confidence interval for nonlinear model function parameter.

Note

This function implemented for calling for non object purpose, for example computing covarianc matrix for Weighted M-estimate stored as nl.fitt.rgn but using classic covariance computation using gradinet, instead parInfer which use convergence properties (Lim et al. 2010)

This function call by nlr, for compatibility it is better to call from nlr rather than directly by user.

Author(s)

Hossein Riazoshams, Jan 2010. Email: <riazihosein@gmail.com> URL http://www.riazoshams.com/nlr/

References

Seber, G., A. F. and Wild, C. J. (2003). Nonlinear Regression. New York: John Wiley & Sons, Inc.

Lim, C., Sen, P. K., Peddada, S. D. (2010). Statistical inference in nonlinear regression under heteroscedasticity. Sankhya B 72:202-218.

See Also

nl.fitt, nl.fitt.gn, nl.fitt.rob, nl.fitt.rgn

Examples

```
## The function is currently defined as
"pInf"
```

---

plotinitial *Initial Values plot.*

Description

Plot the fited curve and data, at a certain point of parameters, a given user parameter value (initial values mostly), or from selfStart slot.

Usage

```
plotinitial(form, data, start = getInitial(form, data), length.out = 100,...)
```

Arguments

form	nl.form object of the nonlinear regression model function.
data	List of data set.
start	NULL or list of starting value, include the parameter names.
length.out	Length of the incremented predictor values, to acheive a more smooth curve.
...	extra argument to be passed to plot function, for further developement purposes.

Details

Plot initial can plot any fited curve at any parameter value. Technically it is implemented to test initial values computing by selfStart slot of nl.form object. But user can insert hi/her own starting value or even the final fited value.

Value

Two dimensional Fitted curve and points.

Note

Although the final parameter estimates can be used to fit the curve, but it has less options than `plot` function.

Author(s)

Hossein Riazoshams, May 2014. Email: <riazihosein@gmail.com> URL `http://www.riazoshams.com/nlr/`

References

Riazoshams H, Midi H, and Ghilagaber G, 2018,. Robust Nonlinear Regression, with Application using R, Joh Wiley and Sons.

Seber, G., A. F. and Wild, C. J. (2003). Nonlinear Regression. New York: John Wiley & Sons, Inc.

See Also

`nl.form`

Examples

```
##
## Plot initial values from selfStart of Scaled Exponential Convex for carbon data
##
  crbdt<-list(xr=nlr::carbon$year,yr=nlr::carbon$co2)
   plotinitial(form = nlrobj5[[8]],data = crbdt)
```

---

plotlist *plot a list of objects.*

Description

Plot each item of objects stored in a list. This is not extention of list object, just to help plot many objects stored in a list. Typically `nlr` fited objects such as `nl.fitt`, `nl.fitt.gn`, `nl.fitt.rob`, `nl.fitt.rgn` has plot method and the result of many model fitts can be stored in a list, then `plotlist` function plot all of them.

Usage

```
plotlist(listobj,...)
```

Arguments

listobj	An nl.fitt, nl.fitt.gn, nl.fitt.rob, nl.fitt.rgn obejct after a nonlinear model fitted before.
...	An ..., argument passe to plot function.

Details

plot items in a least one by one.

Value

Plot of objects.

Note

Used mostly for internal purposes.

Author(s)

Hossein Riazoshams, Jan 2010. Email: <riazihosein@gmail.com> URL http://www.riazoshams.com/nlr/

References

Riazoshams H, Midi H, and Ghilagaber G, 2018,. Robust Nonlinear Regression, with Application using R, Joh Wiley and Sons.

See Also

nl.fitt, nl.fitt.gn, nl.fitt.rob, nl.fitt.rgn

Examples

```
nn<-length(nlrobj1)
d<-list(xr=nlr::Weights$Date, yr=nlr::Weights$Weight)
wmodel.nlrob <-list( NULL)
j<-1
for(i in 1:3){

  start<-as.list(nlrobj1[[i]]$par)
  #wmodel.nlrob[i] <-list( nlmest(nlrobj1[[i]],data=d,start=start,robfunc=nl.robfuncs[[1]]))
  a1 <- nlr(nlrobj1[[i]],data=d,robustform = "hampel")
  if (! is.Fault(a1)){
    wmodel.nlrob[j]<-a1
    j<-j+1
  }
}
plotlist(wmodel.nlrob)
```

prodAV *Compute product of Array into a Vector.*

Description

function: prodAV product array to matrix

Usage

```
prodAV(ary, vector)
```

Arguments

ary	is (p*p*n) array.
vector	is a vector with dimension (p)

Details

function: prodAV product array to matri. compute: 'array * vector', which is (n,p) vector.

Value

'array * vector', which is (n,p) vector.

Note

Three dimentional array multiple matrix have several possibilities, see refeerence. prodVAV is another case of vector multiple array multiple vector. Is for feature purposes.

Author(s)

Hossein Riazoshams, Jan 2010. Email: <riazihosein@gmail.com> URL http://www.riazoshams.com/nlr/

References

Riazoshams H, Midi H, and Ghilagaber G, 2018,. Robust Nonlinear Regression, with Application using R, Joh Wiley and Sons.

See Also

%3d*m%, %m3d% prodVA

Examples

```
## The function is currently defined as
"prodAV"
```

prodVA *Product of three dimentional array in vector.*

Description

Array(n*p*p)*Vector(n*1) is (p*p) matrix equal the sum of vector multiple first dimention of array.

Usage

```
prodVA(ary, vector)
```

Arguments

ary	(n*p*P) numeric array.
vector	(p*p) matrix.

Details

It can be used to multiple a (n*p*p) Hessian into (n*1) vector of residuals.

Value

(p*p) matrix.

Note

It is used in optimization and loss functions, may not explicitly called by user. May be used in defining loss functions by user.

Author(s)

Hossein Riazoshams, May 2014. Email: <riazihosein@gmail.com> URL http://www.riazoshams.com/nlr/

References

Riazoshams H, Midi H, and Ghilagaber G, 2018,. Robust Nonlinear Regression, with Application using R, Joh Wiley and Sons.

See Also

%3d*m%, %m3d%

Examples

```
## The function is currently defined as
"prodVA"
a=array(1:8,c(2,2,2))
b=c(2,3)
d=prodVA(a,b)
```

`psi.hampel` *hampel redescending function*

Description

- psi.hampel: compute hampel psi function.
- rho.hampel: compute hampel rho function.

Usage

```
psi.hampel(u, a = 1.5, b = 3.5, c = 8)
rho.hampel(u, a = 1.5, b = 3.5, c = 8)
```

Arguments

`u`	vector of values.
`a`	tuning constant, for wich less than that the quadratic function compute.
`b`	tuning constant, for wich less than that the linear function compute.
`c`	tuning constant, for wich less than that the constant function compute.

Details

psi and rho function used to in MM-estimate. The tuning constants together with k0,k1 used to give high breakdown point. This is function in MASS library, will be developed in nlr latter on.

Value

vector of psi and who function values.

Note

This function is created at begining for simplicity but in feature will not used by `nlr` any more.

Author(s)

Stromberg, A. J. (1993)

References

Stromberg, A. J. (1993). Computation of High Breakdown Nonlinear Regression Parameters, Journal of American Statistical Association 88(421): 237-244. Stromberg, J., and Ruppert, D. (1992). Breakdown in Nonlinear Regression, Journal of American Statistical Association 87: 991-997.

See Also

`rho.hampel`

Examples

```
## The function is currently defined as
"psi.hampel"
```

---

robloss *Robut loss function.*

Description

Resturn robust loss function for minimization purpose to find the M-estimate.

Usage

```
robloss(formula, data, start, robfunc, control = nlr.control(robscale = T), ...)
```

Arguments

`formula`	`nl.form` object of nonlinear regression model.
`data`	list of data include responce and predictor.
`start`	list of parameter values of nonlinear model function (θ in $f(x, \theta)$), initial values or increament during optimization procedure. It must include scale sigma (standard deviation), if not included Fault(9) will be returned.
`robfunc`	`nl.form` of rho function. It must include tuning constants k0 and k1.
`control`	list of `nlr.control` for controling convergence criterions.
`...`	any other arguments might be used in formula, robfunc or tuning constants in rho function.

Details

Compute Loss function, sum of robust rho function to compute the M-estimate.

$$\ell(\theta) = \sum \rho\left(\frac{r_i}{\sigma}\right)$$

Standard deviation σ must be included in `start` argument list with the name `sigma`.

Value

list of output:

htheta	sum of rho function, include attribute "gradient" and "hessian"
rho	computed rho function and attributes of "gradient" and "hessian"
ri	residuals
hessh.p1	hessian of loss function part1
hessh.p2	hessian of loss function part2, in clasic this part removed but in robust statistics values are significant and can not be omited, See Riazoshams et al 1014
dtilda	D(thilda) part of hessian
fmod	computed function contains esponse and or its gradient and hessian predictor and or its gradient & hessian
Fault	Fault object of error, if no error Fault number = 0 will return back.

Note

All functions should have gradient and hessian in attributes. For derivative free purpose the dfr.robloss can be used.

This function call by nlr, for compatibility it is better to call from nlr rather than directly by user.

Author(s)

Hossein Riazoshams, May 2014. Email: <riazihosein@gmail.com> URL http://www.riazoshams.com/nlr/

References

Riazoshams H, Midi H, and Ghilagaber G, 2018,. Robust Nonlinear Regression, with Application using R, Joh Wiley and Sons.

See Also

nl.form, nlr.control, nlmest.NLM, dfr.robloss

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--or do  help(data=index)  for the standard data sets.

## The function is currently defined as
"robloss"
```

`robloss.gn` *Generalized Robut loss function.*

Description

Resturn Generalized robust loss function for minimization purpose to find the Generalized M-estimate. Generalized M-estimate required correlation or covariance matrix of data, then the model transform and estimated.

Usage

```
robloss.gn(formula, data, start, robfunc, rmat, control = nlr.control(robscale = T)
```

Arguments

`formula`	`nl.form` object of nonlinear regression model.
`data`	list of data include responce and predictor.
`start`	list of parameter values of nonlinear model function (θ in $f(x, \theta)$), initial values or increament during optimization procedure. It must include scale sigma (standard deviation), if not included Fault(9) will be returned.
`robfunc`	`nl.form` of rho function. It must include tuning constants k0 and k1.
`rmat`	R matrix, is cholesky decomposition of covariance matrix, the model transform by multiplying by R matrix.
`control`	list of `nlr.control` for controling convergence criterions.
`...`	any other arguments might be used in formula, robfunc or tuning constants in rho function.

Details

Compute Loss function, sum of robust rho function to compute the M-estimate.

$$\ell(\theta) = \sum \rho\left(\frac{R \times r_i}{\sigma}\right)$$

Standard deviation σ must be included in `start` argument list with the name `sigma`.

The R matrix is `rmat` argument.

Value

list of output:

`htheta`	sum of rho function, include attribute `"gradient"` and `"hessian"`
`rho`	computed rho function and attributes of `"gradient"` and `"hessian"`
`ri`	residuals, transformed by R.
`hessh.p1`	hessian of loss function part1

hessh.p2	hessian of loss function part2, in clasic this part removed but in robust statistics values are significant and can not be omited, See Riazoshams et al 1014
dtilda	D(thilda) part of hessian
fmod	computed function (transformed by R) contains response and or its gradient and hessian predictor, transformed also by R.
Fault	Fault object of error, if no error Fault number = 0 will return back.

Note

This function use in optimization functions, specially from nlmest.NLM, for where the covariance matrix or R matrix given, may not be called explicitly by user.

Generalized M-estimate might reperesent the autocorrelated or heteroscedastic variance case.

This function call by nlr, for compatibility it is better to call from nlr rather than directly by user.

Author(s)

Hossein Riazoshams, May 2014. Email: <riazihosein@gmail.com> URL http://www.riazoshams.com/nlr/

References

Riazoshams H, Midi H, and Ghilagaber G, 2018,. Robust Nonlinear Regression, with Application using R, Joh Wiley and Sons.

See Also

nl.form, nlr.control, nlmest.NLM

Examples

```
## The function is currently defined as
"robloss.gn"
```

---

rzvalues *Robust sample variance*

Description

Compute Robust Sample variance for dependent variable. For repeated response the sample Mead Absolute Deviance (MAD) of predictor compute over a cross section of predictor, for non repeated data sample variance of response computed by difference of residuals of consequetive values.

Usage

```
rzvalues(res, ni, xo)
```

Arguments

res	The data to compute variance for. In most of application it is residuals of fitt.
ni	vector of bumber of repeated data. It can be output from nonrepl function.
xo	Position of the repeated data in original vector.

Details

Typically it is used to compute the robust variance of residuals output from nonrepl function.

Value

vector of robust sample variance.

Note

This is robustified form of sample variance function zvalues.

Author(s)

Hossein Riazoshams, Jan 2010. Email: <riazihosein@gmail.com> URL http://www.riazoshams.com/nlr/

References

Riazoshams, H. (2012), Robustifying the Least Squares estimate of parameters of variance model function in nonlinear regression with heteroscedastic variance, Poster Presentation, Royal Statistical Society Conference (RSS) 2012, Telford, UK.

Riazoshams H, Midi H, and Ghilagaber G, 2018,. Robust Nonlinear Regression, with Application using R, Joh Wiley and Sons.

See Also

zvalues

Examples

```
## The function is currently defined as
"rzvalues"
```

`sqrtvat` *Compute square root of vairiance attribute.*

Description

Transform variance to standard deviation with all its gradient and hessian.

Usage

```
sqrtvat(varcomp)
```

Arguments

`varcomp`	Is (n*1) vector of some variance, transform to $\sqrt(vc)$, with attributes attr(vc,"gradient"), $n \times p$ gradient. And attr(vc,"hessian"), $n \times p\, timesp$ hessian.

Details

For computation purpose to transform variance function values to standard deviation function value is used.

Value

Standard deviation is equal the square root of variance, with Gradient equal to:

$$Gradient(sdev) = \frac{1}{2}\sqrt{Var} \times Gradient(Var)$$

and hessian is equal

$$hessian(sdev) = \frac{1}{2}\sqrt{vc} \times hesian(vc) - (\frac{1}{4}\sigma^3) grad(vc)^T \%m3d\% grad(vc)$$

Note

Is used for when standard deviation of a heterogeneous variance function model is needed.

This function call by `nlr`, for compatibility it is better to call from `nlr` rather than directly by user.

Author(s)

Hossein Riazoshams, May 2014. Email: <riazihosein@gmail.com> URL `http://www.riazoshams.com/nlr/`

References

Riazoshams H, Midi H, and Ghilagaber G, 2018,. Robust Nonlinear Regression, with Application using R, Joh Wiley and Sons.

See Also

```
nlr
```

Examples

```
## The function is currently defined as
"sqrtvat"
```

`tadr.ir` *Iran Birth Rate.*

Description

Birth rate, crude (per 1,000 people).

Usage

```
tadr.ir
```

Format

The format is: chr "tadr.ir"

- year: year
- tadr: birth rate per

Details

Birth rate, crude (per 1,000 people). From 1960 to 2011.

Source

http://www.worldbank.org/

References

worldbank.com

Examples

```
data(tadr.ir)
tadr.ir
```

`theophil` *Iran Trade Data.*

Description

Trademark applications, direct resident.

Usage

```
theophil
```

Format

The format is: chr "theophil"

- time: time
- concenteration: theophil concenteration

Details

Theophil concentration, the two consecutive first order reaction fitts to data, for which the model stored in `fktlistex6` object.

Source

Unknown web, 2007.

References

Riazoshams H, Midi H, and Ghilagaber G, 2018,. Robust Nonlinear Regression, with Application using R, Joh Wiley and Sons.

Examples

```
data(theophil)
theophil
```

trade.ir *Iran Trade Data.*

Description

Trademark applications, direct resident.

Usage

```
trade.ir
```

Format

The format is: chr "trade.ir"

- year: year
- trade: trademark

Details

Trademark applications, direct resident from 1960 to 2006.

Source

http://www.worldbank.org/

References

worldbank.com

Examples

```
data(trade.ir)
trade.ir
```

transform *Transform by R matrix.*

Description

Transform $R \times value$ and its gradient and hessian.

Usage

```
transform(value,rm)
```

Arguments

value	A vector to be transformed by R-matrix. It must have gradient and hessian as attributes.
rm	The R-matrix, from QR decomposition.

Details

It is used to transfomr a nonlinear regression model by R matrix from QR decomposition of variance covariance matrix.

Value

Is a transformed vector $R \times V$, and hessian and gradient as its attributes. Gradient is equal:

$$gradient = rm \times g$$

and hessian equal the three dimentional product:

$$hessian = h\%3d * m\%rm$$

.

Note

It is used in generalized model. In both heteroscedastic and autocorrelated model will be applied.

This function call by `nlr`, it might not be called directly by user.

Author(s)

Hossein Riazoshams, May 2014. Email: <riazihosein@gmail.com> URL http://www.riazoshams.com/nlr/

References

Robust Nonlinear Regression, Theories and Methods with Practical Guides for R Packages. Riazoshams et al.

See Also

`nlmest.NLM`, `nlmest.NLMf`, `nlmest.WF`

Examples

```
## The function is currently defined as
"transform"
```

`transformNR` *Transform nonlinear regression model*

Description

Transform both side a nonlinear regression model by cholesky decomposition of covariance matrix of errors. The new transformed model has constant variance and uncorrelated errors.

- `transform`: Transform using `rm` matrix
- `transforminv`: Inverse transform using inverse of `rm` matrix

Usage

```
transformNR(value, rm)
```

Arguments

`value`	Numeric value with (n by p) `"gradient"` and (three dimentional n by p by p) `"hessian"` attributes. Usually it is right side (predictors) or left side (response) of a nonlinear regression model.
`rm`	Matrix for transforming. In nonlinear regression if covaraince matrix of error represent by $\sigma^2 V$, `rm` is cholesky decompostion of `V`.

Details

If errors of a nonlinear regression is Autocorrelated or heteroscedastic, the model can be transformed to uncorrelated and homoscedastic using cholesky decomposition.

- `transformNR`: Multiply by `rm` matrix, create standardized model.
- `transforminv`: Multiply by inverse of `rm` matrix, create heteroscedastic or autocorrelated model.

Value

Transformed values including (n by p) `"gradient"` and (three dimentional n by p by p) `"hessian"` attributes.

Note

For transfering a nonlinear regression model both side should be transformed. In `nlr` package generalized nonlinear model use this transformation.

This function call by `nlr`, for compatibility it is better to call from `nlr` rather than directly by user.

Author(s)

Hossein Riazoshams, May 2014. Email: <riazihosein@gmail.com> URL `http://www.riazoshams.com/nlr/`

References

Riazoshams H, Midi H, and Ghilagaber G, 2018,. Robust Nonlinear Regression, with Application using R, Joh Wiley and Sons.

Seber, G., A. F. and Wild, C. J. (2003). Nonlinear Regression. New York: John Wiley & Sons, Inc.

See Also

`transforminv`

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--or do  help(data=index)  for the standard data sets.

## The function is currently defined as
"transformNR"
```

`transfquad` *Variance to standard deviation transform.*

Description

This function compute the standard deviation of an output from a parameteric nonlinear variance function object, usualy stored as `nl.form`. The gradient and hessian attributes also will be transformed. `transfquadvec` transform a vector similarly.

Usage

```
transfquad(varcomp)
```

Arguments

`varcomp`	Is vector of variances, with (n by p) `"gradient"` and (three dimentional n by p by p) `"hessian"` attributes.

Details

The standard deviation is simply square root of variance. The gradinet is transformed of square root of gradient. Hessian is transformed of second derivative of square root function as well.

Value

Vector of transformed standard deviation, including (n by p) `"gradient"` and (three dimentional n by p by p) `"hessian"` attributes.

Note

In `nlr` variance function is stored in `nl.form` for computations.

This function call by `nlr`, for compatibility it is better to call from `nlr` rather than directly by user.

Author(s)

Hossein Riazoshams, May 2014. Email: <riazihosein@gmail.com> URL `http://www.riazoshams.com/nlr/`

References

Riazoshams H, Midi H, and Ghilagaber G, 2018,. Robust Nonlinear Regression, with Application using R, Joh Wiley and Sons.

See Also

`nl.form`

Examples

```
## The function is currently defined as
"transfquad"
```

Tumor *Tumor metastasis data.*

Description

The response variable is the fraction of breast cancer patients with metastases and the predictor variable is the tumor size.

Usage

```
data("Tumor")
```

Format

The format is: chr "Tumor"

- tumorSize predictor variable is the tumor size.
- metastasized response variable is the fraction of breast cancer patients with metastases.

Details

The data consist of 12 observations. The response variable is the fraction of breast cancer patients with metastases and the predictor variable is the tumor size.

Source

Michaelson JS, Halpern E, Kopans D. Breast cancer: Computer simulation methodfor estimating optimal intervals for screening. Radiology. 1999;21:551-560.

References

Michaelson JS, Halpern E, Kopans D. Breast cancer: Computer simulation methodfor estimating optimal intervals for screening. Radiology. 1999;215:51-560.

Examples

```
data(Tumor)
Tumor
```

---

var1 *Compute Variance.*

Description

Compute variance of a vector.

Usage

```
var1(x)
```

Arguments

x	Vector of data.

Details

Only variance, for internal use.

Value

Numeric Variance of a vector.

Note

For internal use. might not be directly called by user.

Author(s)

Hossein Riazoshams, 2013. Email: <riazihosein@gmail.com> URL http://www.riazoshams.com/nlr/

References

Robust Nonlinear Regression, Theories and Methods with Practical Guides for R Packages. Riazoshams et al.

See Also

`nlr`

Examples

```
## The function is currently defined as
"var1"
```

`Weights` *Chicken growth data.*

Description

weights of chicken growth in 51 day of chicken meet production.

Usage

```
data(Weights)
```

Format

The format is: data.frame chr "Weights"

- Date: Date of sampling.
- Weight: weight of each selected randomply chicken.

Details

Weights of chicken randomly selected during the all production period in a chicken breeding farms. Data collected from a chicken breeder chamber, at a local area in Marvdasht at Fars Province of Iran. the beginning population of Flocks was 7300 chicks, at he first 25th dates we have weighted 10 chicken in each date, after that due to wider spread of weight in time, we have collected 20 chicken in each two date. In collecting chickens at each date, we tried to collect completely random data from different part of the chamber.

Source

Riazoshams, H., Miri. H. (2005), Investigating growth models using nonlinear regression models, Research project presented to abade islamic azad university, Abade/Iran.

References

Riazoshams, H., Midi, H. (2009), A Nonlinear regression model for chickens' growth data. European Journal of Scientific Research, 35(3):393-404.

Examples

```
data(Weights)
Weights
```

`zvalues` *Sample Variance*

Description

Compute Sample variance for dependent variable. For repeated response the sample variance of predictor compute over a cross section of predictor, for non repeated data sample variance of response computed by difference of residuals of consequetive values.

Usage

```
zvalues(res, ni, xo)
```

Arguments

`res`	The data to compute variance for. In most of application it is residuals of fitt.
`ni`	vector of bumber of repeated data. It can be output from `nonrepl` function.
`xo`	Position of the repeated data in original vector.

Details

Typically it is used to compute the variance of residuals output from `nonrepl` function.

Value

vector of classic variance.

Note

The robustified form of this function is `rzvalues`. This function call by `nlr`, it might not be called directly by user.

Author(s)

Bunke, O., Droge, B., Polzehl.

References

Bunke, O., Droge, B., Polzehl, J. Splus tools for model selection in nonlinear regression (1998) Computational Statistics, 13 (2), pp. 257-281.

See Also

rzvalues

Examples

```
## The function is currently defined as
"zvalues"
```

`%3d*m%` *Product array to matrix*

Description

Binary operator, compute multiplication of three dimentional array to a vector, sum over first dimention of array to the vector. Technically cross section values in first dimention are equal. It use for computation purposes in optimizations.

Usage

```
ary %3d*m% vector
```

Arguments

ary	(n*p*p) numeric array.
vector	(n*1) numeric vector.

Details

Three dimentional array multiple by vector, first dimention of array and vector related to sample data, multiple of them and sum is a fixed value will be returned back.

Value

result(:,i,j) = sum (ary[,i,j]*vector), but repeat in all cross section, thus the cross section values are equal.

Note

It use for computation purposes in optimization, might not be called directly by user.

Author(s)

Hossein Riazoshams, May 2013. Email: <riazihosein@gmail.com> URL http://www.riazoshams.com/nlr/

References

Riazoshams H, Midi H, and Ghilagaber G, 2018,. Robust Nonlinear Regression, with Application using R, Joh Wiley and Sons.

See Also

`%m3d%`, `prodVA`

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--or do  help(data=index)  for the standard data sets.
## The function is currently defined as
"%3d*m%"
a=array(1:8,c(2,2,2))
b=3:4
d=a %3d*m% b
```

%c% *cross product binary operator.*

Description

Binary operator, given matrices x and y as arguments, return a matrix cross-product.

Usage

```
x %c%  y
```

Arguments

`x`	(m*n) matrix.
`y`	(m*q) matrix.

Details

result is (n*m) matrix = t(x) %*% y, sum(x[,i]*y[,j])

Value

A double or complex matrix, with appropriate dimnames taken from x and y.

Note

This is binary operator of `crossprod` adjusted in `nlr` for more streamlined and readable. Mostly used in optimization functions, might not be called by user explicitly.

Author(s)

Hossein Riazoshams, May 2014. Email: `<riazihosein@gmail.com>` URL `http://www.riazoshams.com/nlr/`

References

Becker, R. A., Chambers, J. M. and Wilks, A. R. (1988) The New S Language. Wadsworth & Brooks/Cole.

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--or do  help(data=index)  for the standard data sets.

## The function is currently defined as
(1:4) %c% (1:4) ## result is sum(1^2+2^2+3^2+4^2)
 m1 <- matrix(c(1:4),nrow=2)
 m2 <- matrix(c(1:8),nrow=2)
 m1 %c% m2
```

`%m3d%` *Matrix 3 dimentional product.*

Description

Binary Operator, for sample size (n), vector with size p of gradient stored in (n*p) matrix, transpose of gradient multiple by gradient for (n) points can be obtained by (gradient %m3d% gradient). Result save in (n*p*p) three dimentional array.

Usage

```
mat1 %m3d%  mat2
```

Arguments

`mat1`	n*p matrix, in gradient example n is sample size, p is number of parameters, mat1[i,] is (p) vector of gradient of ith sample.
`mat2`	p*n matrix.

Details

It is Used to implement gradient product for n sample

Value

$$Array_{n*p*p} = \nabla_\theta^T f_i(\theta) \nabla_\theta f_i(\theta), i = 1, \ldots, n$$

Author(s)

Hossein Riazoshams, May 2014. Email: <riazihosein@gmail.com> URL http://www.riazoshams.com/nlr/

References

Riazoshams H, Midi H, and Ghilagaber G, 2018,. Robust Nonlinear Regression, with Application using R, Joh Wiley and Sons.

See Also

`%3d*m%`, `prodVA`

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--or do  help(data=index)  for the standard data sets.

## The function is currently defined as
"%m3d%"
m1 <- matrix(c(1:4),nrow=2)
m2 <- matrix(c(1:4),nrow=2)
d=m1 %m3d% m2
```