Package 'hett'

May 4, 2005

Version 0.2

Date 2004-05-03

Title Heteroscedastic t regression

Author Julian Taylor <julian.taylor@adelaide.edu.au>

Maintainer Julian Taylor <julian.taylor@adelaide.edu.au>

Depends R (>= 1.8.0), MASS

Description Functions for the fitting and summarizing of heteroscedastic t-regression.

License GPL Version 2 or later.

URL http://www.biometricssa.adelaide.edu.au/staff/staff_jtaylor.shtml/hett

R topics documented:

`dof.profile` *Internal profile likelihood function*

Description

Internal profile log-likelihood function for `tlm` function

Usage

```
dof.profile(dof, n, sqResid, orthoI, X, Z)
```

Arguments

dof	degrees freedom value
n	number of values in the response vector
sqResid	squared residuals
orthoI	orthogonalized scale parameters
X	design matrix of explanatory variables for the location model
Z	design matrix of explanatory variables for the scale model

Details

This function is *not* intended to be directly called by users.

Value

a profile log-liklehood value

---

mm *Excess returns for Martin Marietta company*

Description

Data from the Martin Marietta company collected overa period of 5 years on a monthly basis

Usage

```
data(mm)
```

Format

A data frame with 60 observations on the following 4 variables.

date the month the data was collected

am.can a numeric vector

m.marietta excess returns from the Martin Marietta company

CRSP an index for the excess rate returns for the New York stock exchange

Source

Bulter et al (1990). Robust and partly adpative estimation of regression models. *Review of Economic Statistics*, **72**, 321-327.

Examples

```
data(mm, package = "hett")
attach(mm)
plot(CRSP, m.marietta)
lines(CRSP, fitted(lm(m.marietta ~ CRSP)), lty = 2)
```

rent *Rent for Land PLanted to Alfalfa*

Description

Dataset collected in 1977 from Minnesota to study the variation in land rented for growing alfalfa

Usage

```
data(rent)
```

Format

A data frame with 67 observations on the following 5 variables.

Rent a numeric vector average rent per acre.

AllRent a numeric vector describing average rent paid for all tillable land.

Cows a numeric vector describing the density of dairy cows (number per square mile).

Pasture a numeric vector describing the proportion of farmland used as pasture.

Liming a factor with levels `No` if no liming is required to grow alfalfa and `Yes` if it does.

Source

Weisberg, S (1985). *Applied Linear Regression* Wiley: New York

Examples

```
library(lattice)
data(rent, package = "hett")
attach(rent)
xyplot(log(Rent/AllRent) ~ sqrt(Cows), groups = Liming, panel = panel.superpose)
```

summary.tlm *summary method for class "tlm"*

Description

Summarizes the heteroscedastic *t* regression object

Usage

```
## S3 method for class 'tlm':
summary(object, correlation = FALSE, ...)
## S3 method for class 'summary.tlm':
print(x, ...)
```

Arguments

object	heteroscedastic t regression object called from tlm()
x	an object of class "summary.tlm" containing the values below
correlation	should the calaculation of the parameter correlation matrix be supressed. If the fit includes a location and a scale formula then both correlation matrices are printed. The default is FALSE.
...	arguments passed to or from other methods

Details

The table summary produced by this function should be used with caution. A more appropriate test between nested models is to use the score statistic function tscore.

Value

a list containing the following components:

loc.summary	an object containing a list of objects that summarize the location model
scale.summary	an object containing a list of objects that summarize the scale model
iter	the number of iterations of the algorithm
dof	value of the fixed or estimated degrees of freedom
dofse	the standard error associated with the degrees of freedom if estimated
logLik	the maximised log-likelihood
method	the method used to maximize the likelihood
endTime	the time taken for the algorithm to converge

See Also

tsum, tlm

Examples

```
data(mm, package = "hett")
attach(mm)

## fit a model with heteroscedasticity and estimating the degrees of freedom

tfit2 <- tlm(m.marietta ~ CRSP, ~ CRSP, data = mm, start = list(dof =
3), estDof = TRUE)
summary(tfit2)
```

`tlm.control` *Auxiliary for Controlling tlm Fitting*

Description

Auxiliary function for fitting `tlm` model. Generally only used when calling `tlm`

Usage

```
tlm.control(epsilon = 1e-07, maxit = 50, trace = FALSE, verboseLev = 1)
```

Arguments

`epsilon`	positive convergence tolerance value. The iterations converge when [newlik - oldlik] < epsilon/2
`maxit`	integer giving the maximum iterations allowable for the routine
`trace`	logical. If `TRUE` output is printted to the screen during each iteration
`verboseLev`	integer. If 1 then print accroding to `trace`. If 2 then print random scale effects also.

Details

Value

A list with the argument as values

See Also

`tlm`

Examples

```
data(mm, package = "hett")
attach(mm)

## change the maximum amount of iterations for the algorithm

fit1 <- tlm(m.marietta ~ CRSP, ~ 1, data = mm, start = list(dof = 3),
estDof = TRUE, control = tlm.control(maxit = 100))
```

`tlm` *Maximum likelihood estimation for heteroscedastic t regression*

Description

Fits a heteroscedastic t regression to given data for known and unknown degrees of freedom.

Usage

```
tlm(lform, sform =  ~ 1, data = sys.parent(), subset = NULL, contrasts =
    NULL, na.action = na.fail, start = NULL, control = tlm.control(...),
    obs = FALSE, estDof = FALSE, ... )

## S3 method for class 'tlm':
print(x, ...)
```

Arguments

`x`	an object of class "`tlm`"
`lform`	a formula of the type `response ~ terms`, where `terms` can be of the form, for example, `first + second` or `first*second`(see `lm` for details)
`sform`	a formula of the type `~ terms`, where `terms` can be of the form, for example, `first + second` or `first*second`(see `lm` for details).
`data`	the data in the form of a `data.frame` where the column names can be matched to the variable names supplied in `lform` and `sform`
`subset`	numerical vector to subset the `data` argument
`contrasts`	set of contrasts for the location model (see `contrasts.arg` for details)
`na.action`	the action to proceed with in the event of `NA`'s in the response. Currently `NA`'s are not allowed and therefore `na.fail` is the sole argument.
`start`	is a list of possibly four named components, ("`beta`", "`lambda`", "`dof`", "`omega`"), for the location, scale, degrees of freedom parameters and random scale effects respectively. Each component must be of the appropriate length.
`control`	is an argument to a function that maintains the control of the algorithm. The `tlm.control()`function contains the arguments, `epsilon` to determine how small the relative difference of likelihoods should be for convergence (default is 1e-07), `maxit` to determine the maximum iterations required (default = 50), `trace` if the user requires printing of estimates etc. as algorithm runs (default = `FALSE`), `verboseLev` to determine the amount of verbose printing to the screen as the algorithm runs (verboseLev = 1 displays location scale and dof estimates and the likelihood, verboseLev = 2 displays all of 1 plus the random scale effects)
`obs`	should the location parameters be calculated using the observed or expected information(default = `FALSE`). (Note: using the observed information does not calculate the appropriate standard errors, see DETAILS)
`estDof`	should the degrees of freedom parameter be estimated or not. If `FALSE` then the value given for `dof` in the `start` argument will be the fixed value used for the algorithm. If `TRUE` then the value given for `dof` in the `start` argument supplies an initial value only.
`...`	arguments passed to `tlm.control() or to the code{print} method`

Details

When the degrees of freedom is unknown the code uses the non-linear optimiser `nlm`. If the data is tending toward the Gaussian this optimisation will still converge but with with very high degrees of freedom.

To obtain the appropriate standard errors from `summary` the user must specify the argument `obs = F` to ensure that the location parameter is calculated using the expected information component.

Value

a list containing the following components:

`loc.fit`	an object containing the estimated location parameters and other elements associated with the location parameter model
`scale.fit`	an object containing the estimated scale parameters and other elements associated with the scale parameter model
`random`	the random scale effects
`dof`	fixed or estimated degrees of freedom
`dofse`	the standard error associated with the degrees of freedom
`iter`	the number of iterations of the algorithm
`logLik`	the maximised log-likelihood
`endTime`	the time taken for the algorithm to converge

Background

The theoretical background for this function can be found in Taylor and Verbyla (2004)

References

Taylor, J. D. & Verbyla, A. P (2004). Joint modelling of the location and scale parameters of the *t*-distribution. *Statistical Modelling* **4**, to appear.

See Also

`summary.tlm`

Examples

```
data(mm, package = "hett")
attach(mm)

## fit a model with no heteroscedasticity and fixed degrees of freedom

tfit <- tlm(m.marietta ~ CRSP, data = mm, start = list(dof = 3))

## fit a model with heteroscedasticity and fixed degrees of freedom

tfit1 <- tlm(m.marietta ~ CRSP, ~ CRSP, data = mm, start = list(dof = 3))

## fit a model with heteroscedasticity and estimating the degrees of freedom

tfit2 <- tlm(m.marietta ~ CRSP, ~ CRSP, data = mm,
start = list(dof = 3), estDof = TRUE)
```

`tscore` *Score test for heteroscedastic t models*

Description

Provides a score test for the location and scale parameters of the heteroscedastic t regression model.

Usage

```
tscore(..., data = NULL, scale = FALSE)
```

Arguments

`...`	Any number of arguments containing nested model fits from `tlm()` (see Details)
`data`	the data used to fit the models involved
`scale`	logical. If `TRUE` the scale model is tested

Details

The user must supply nested models that test, *either*, the scale or the location component of the model. The model objects *must* be nested from left to right. Currently there are no traps if the arguments are not given in this order.

The models must also have either, all fixed degrees of freedom or estimated degrees of freedom.

Value

Output containing the hypothesis, the score statistic, degrees of freedom for the test and the p-value are printed to the screen.

....

See Also

`tlm`

Examples

```
data(mm, package = "hett")
attach(mm)
tfit1 <- tlm(m.marietta ~ CRSP, ~ 1, data = mm, start = list(dof = 3),
estDof = TRUE)

tfit2 <- tlm(m.marietta ~ CRSP, ~ CRSP, data = mm, start = list(dof =
3), estDof = TRUE)

tscore(tfit1, tfit2, data = mm, scale = TRUE)
```

`tsum` *Summary function for the scale or location component of a heteroscedastic t model*

Description

Summarizes the location *or* scale components of a heteroscedastic *t* model

Usage

```
tsum(object, dispersion = NULL, correlation = FALSE,
             symbolic.cor = FALSE, ...)

## S3 method for class 'tsum':
print(x, digits = max(3, getOption("digits") - 3), symbolic.cor =
      x$symbolic.cor, signif.stars = getOption("show.signif.stars"),
      scale = TRUE, ...)
```

Arguments

`object`	either the location *or* scale object created by fitting a heteroscedastic t object with `tlm`
`x`	an object of class "`tsum`"
`dispersion`	1 if summarizing the location model; 2 if summarizing the scale model (see Details)
`correlation`	logical; if `TRUE`, the correlation matrix of the estimated parameters is returned and printed.
`digits`	the number of significant digits to be printed.
`symbolic.cor`	logical. If `TRUE`, print the correlations in a symbolic form (see 'symnum') rather than as numbers.
`signif.stars`	logical. if `TRUE`, "significance stars" are printed for each coefficient.
`scale`	logical. If `TRUE` then the dispersion is known in advance (2), and is printed accordingly.
`...`	further arguments passed to or from other methods.

Details

The argument supplied to `dispersion` must be either 1 (location model) or 2 (scale model). The reason for this is because the fitting of the model has already scaled the covariance matrix for the location coefficients. Hence the scaled and unscaled versions of covariance matrix for the location model are identical.

This function will not be generally called by the user as it will only summarize the location or scale model but not both. Instead the user should refer to `summary.tlm` to print a summary of both models.

Value

`tsum` returns an object of class "`tsum`", a list with components

`call`	the component from `object`
`df.residual`	the component from `object`
`coefficients`	the matrix of coefficients, standard errors, z-values and p-values
`dispersion`	the supplied dispersion argument
`df`	a 2-vector of the rank of the model and the number of residual degrees of freedom
`cov.unscaled`	the unscaled (`dispersion = 1`) estimated covariance matrix of the estimated coefficients
`cov.scaled`	ditto, scaled by `dispersion`
`correlation`	(only if `correlation` is true.) The estimated correlations of the estimated coefficients
`symbolic.cor`	(only if `correlation` is true.) The value of the argument `symbolic.cor`

See Also

`summary.tlm`, `tlm`

Examples

```
data(mm, package = "hett")
attach(mm)
tfit <- tlm(m.marietta ~ CRSP, ~ CRSP, data = mm, start = list(dof = 3),
estDof = TRUE)
tsum(tfit$loc.fit, dispersion = 1)
```

Index