Package ‘lmtest’

May 28, 2009

Title Testing Linear Regression Models

Version 0.9-24

Date 2009-05-28

Author Torsten Hothorn, Achim Zeileis, Giovanni Millo, David Mitchell

Maintainer Achim Zeileis <Achim.Zeileis@R-project.org>

Description A collection of tests, data sets and examples for diagnostic checking in linear regression models.

LazyLoad yes

LazyData yes

Depends R (>= 2.1.0), stats, zoo

Suggests car, strucchange, sandwich, dynlm, survival

Imports stats, graphics

License GPL-2

Repository CRAN

Date/Publication 2009-05-28 15:46:12

R topics documented:

`bgtest` *Breusch-Godfrey Test*

Description

`bgtest` performs the Breusch-Godfrey test for higher-order serial correlation.

Usage

```
bgtest(formula, order = 1, order.by = NULL, type = c("Chisq", "F"), data = list())
```

Arguments

`formula`	a symbolic description for the model to be tested (or a fitted `"lm"` object).
`order`	integer. maximal order of serial correlation to be tested.
`order.by`	Either a vector `z` or a formula with a single explanatory variable like `~ z`. The observations in the model are ordered by the size of `z`. If set to `NULL` (the default) the observations are assumed to be ordered (e.g., a time series).
`type`	the type of test statistic to be returned. Either `"Chisq"` for the Chi-squared test statistic or `"F"` for the F test statistic.
`data`	an optional data frame containing the variables in the model. By default the variables are taken from the environment which `bgtest` is called from.

Details

Under H_0 the test statistic is asymptotically Chi-squared with degrees of freedom as given in `parameter`. If `type` is set to `"F"` the function returns the exact F statistic which, under H_0, follows an F distribution with degrees of freedom as given in `parameter`.

The starting values for the lagged residuals in the supplementary regression are chosen to be 0.

Value

A list with class `"htest"` containing the following components:

`statistic`	the value of the test statistic.
`p.value`	the p-value of the test.
`parameter`	degrees of freedom.
`method`	a character string indicating what type of test was performed.
`data.name`	a character string giving the name(s) of the data.

Author(s)

David Mitchell <david.mitchell@dotars.gov.au>, Achim Zeileis

References

Johnston, J. (1984): *Econometric Methods*, Third Edition, McGraw Hill Inc.

Godfrey, L.G. (1978): 'Testing Against General Autoregressive and Moving Average Error Models when the Regressors Include Lagged Dependent Variables', *Econometrica*, 46, 1293-1302.

Breusch, T.S. (1979): 'Testing for Autocorrelation in Dynamic Linear Models', *Australian Economic Papers*, 17, 334-355.

See Also

`dwtest`

Examples

```
## Generate a stationary and an AR(1) series
x <- rep(c(1, -1), 50)

y1 <- 1 + x + rnorm(100)

## Perform Breusch-Godfrey test for first-order serial correlation:
bgtest(y1 ~ x)
## or for fourth-order serial correlation
bgtest(y1 ~ x, order = 4)
## Compare with Durbin-Watson test results:
dwtest(y1 ~ x)

y2 <- filter(y1, 0.5, method = "recursive")
```

```
bgtest(y2 ~ x)
```

`bondyield` *Bond Yield*

Description

Bond Yield Data.

Usage

```
data(bondyield)
```

Format

A multivariate quarterly time series from 1961(1) to 1975(4) with variables

RAARUS difference of interest rate on government and corporate bonds,

MOOD measure of consumer sentiment,

EPI index of employment pressure,

EXP interest rate expectations,

Y artifical time series based on RAARUS,

K artifical time series based on RAARUS.

Source

The data was originally studied by Cook and Hendershott (1978) and Yawitz and Marshall (1981), the data set is given in Krämer and Sonnberger (1986). Below we replicate a few examples given in their book. Some of these results differ more or less seriously and are sometimes parameterized differently.

References

T.Q. Cook & P.H. Hendershott (1978), The Impact of Taxes, Risk and Relative Security Supplies of Interest Rate Differentials. *The Journal of Finance* **33**, 1173–1186

J.B. Yawitz & W. J. Marshall (1981), Measuring the Effect of Callability on Bond Yields. *Journal of Money, Credit and Banking* **13**, 60–71

W. Krämer & H. Sonnberger (1986), *The Linear Regression Model under Test*. Heidelberg: Physica

Examples

```
data(bondyield)

## page 134, fit Cook-Hendershott OLS model and Yawitz-Marshall OLS model
## third and last line in Table 6.5

modelCH <- RAARUS ~ MOOD + EPI + EXP + RUS
lm(modelCH, data=bondyield)
dwtest(modelCH, data=bondyield)
## wrong sign of RUS coefficient

modelYM <- RAARUS ~ MOOD + Y + K
lm(modelYM, data=bondyield)
dwtest(modelYM, data=bondyield)
## coefficient of Y and K differ by factor 100

## page 135, fit test statistics in Table 6.6 b)
################################################

## Chow 1971(1)
if(require(strucchange, quietly = TRUE)) {
sctest(modelCH, point=c(1971,1), data=bondyield, type="Chow") }

## Breusch-Pagan
bptest(modelCH, data=bondyield, studentize=FALSE)
bptest(modelCH, data=bondyield)

## Fluctuation test
if(require(strucchange, quietly = TRUE)) {
sctest(modelCH, type="fluctuation", data=bondyield, rescale=FALSE)}

## RESET
reset(modelCH, data=bondyield)
reset(modelCH, power=2, type="regressor", data=bondyield)
reset(modelCH, type="princomp", data=bondyield)

## Harvey-Collier
harvtest(modelCH, order.by= ~ MOOD, data=bondyield)
harvtest(modelCH, order.by= ~ EPI, data=bondyield)
harvtest(modelCH, order.by= ~ EXP, data=bondyield)
harvtest(modelCH, order.by= ~ RUS, data=bondyield)

## Rainbow
raintest(modelCH, order.by = "mahalanobis", data=bondyield)

## page 136, fit test statistics in Table 6.6 d)
################################################

## Chow 1966(1)
if(require(strucchange, quietly = TRUE)) {
sctest(modelYM, point=c(1965,4), data=bondyield, type="Chow") }
```

```
## Fluctuation test
if(require(strucchange, quietly = TRUE)) {
sctest(modelYM, type="fluctuation", data=bondyield, rescale=FALSE) }

## RESET
reset(modelYM, data=bondyield)
reset(modelYM, power=2, type="regressor", data=bondyield)
reset(modelYM, type="princomp", data=bondyield)

## Harvey-Collier
harvtest(modelYM, order.by= ~ MOOD, data=bondyield)
harvtest(modelYM, order.by= ~ Y, data=bondyield)
harvtest(modelYM, order.by= ~ K, data=bondyield)

## Rainbow
raintest(modelYM, order.by = "mahalanobis", data=bondyield)
```

`bptest` *Breusch-Pagan Test*

Description

Performs the Breusch-Pagan test against heteroskedasticity.

Usage

```
bptest(formula, varformula = NULL, studentize = TRUE, data = list())
```

Arguments

`formula`	a symbolic description for the model to be tested (or a fitted `"lm"` object).
`varformula`	a formula describing only the potential explanatory variables for the variance (no dependent variable needed). By default the same explanatory variables are taken as in the main regression model.
`studentize`	logical. If set to `TRUE` Koenker's studentized version of the test statistic will be used.
`data`	an optional data frame containing the variables in the model. By default the variables are taken from the environment which `bptest` is called from.

Details

The Breusch-Pagan test fits a linear regression model to the residuals of a linear regression model (by default the same explanatory variables are taken as in the main regression model) and rejects if too much of the variance is explained by the additional explanatory variables.

Under H_0 the test statistic of the Breusch-Pagan test follows a chi-squared distribution with `parameter` (the number of regressors without the constant in the model) degrees of freedom.

Examples can not only be found on this page, but also on the help pages of the data sets `bondyield`, `currencysubstitution`, `growthofmoney`, `moneydemand`, `unemployment`, `wages`.

Value

A list with class "htest" containing the following components:

statistic	the value of the test statistic.
p.value	the p-value of the test.
parameter	degrees of freedom.
method	a character string indicating what type of test was performed.
data.name	a character string giving the name(s) of the data.

References

T.S. Breusch & A.R. Pagan (1979), A Simple Test for Heteroscedasticity and Random Coefficient Variation. *Econometrica* **47**, 1287–1294

R. Koenker (1981), A Note on Studentizing a Test for Heteroscedasticity. *Journal of Econometrics* **17**, 107–112.

W. Krämer & H. Sonnberger (1986), *The Linear Regression Model under Test*. Heidelberg: Physica

See Also

lm, ncv.test

Examples

```
## generate a regressor
x <- rep(c(-1,1), 50)
## generate heteroskedastic and homoskedastic disturbances
err1 <- rnorm(100, sd=rep(c(1,2), 50))
err2 <- rnorm(100)
## generate a linear relationship
y1 <- 1 + x + err1
y2 <- 1 + x + err2
## perform Breusch-Pagan test
bptest(y1 ~ x)
bptest(y2 ~ x)
```

ChickEgg *Chickens, Eggs, and Causality*

Description

US chicken population and egg production.

Usage

```
data(ChickEgg)
```

Format

An annual time series from 1930 to 1983 with 2 variables.

chicken number of chickens (December 1 population of all US chickens excluding commercial broilers),

egg number of eggs (US egg production in millions of dozens).

Source

The data set was provided by Walter Thurman and made available for R by Roger Koenker. Unfortunately, the data is slightly different than the data analyzed in Thurman & Fisher (1988).

References

Thurman W.N. & Fisher M.E. (1988), Chickens, Eggs, and Causality, or Which Came First?, *American Journal of Agricultural Economics*, 237-238.

Examples

```
## Which came first: the chicken or the egg?
data(ChickEgg)
## chickens granger-cause eggs?
grangertest(egg ~ chicken, order = 3, data = ChickEgg)
## eggs granger-cause chickens?
grangertest(chicken ~ egg, order = 3, data = ChickEgg)

## To perform the same tests `by hand', you can use dynlm() and waldtest():
if(require(dynlm)) {
  ## chickens granger-cause eggs?
  em <- dynlm(egg ~ L(egg, 1) + L(egg, 2) + L(egg, 3), data = ChickEgg)
  em2 <- update(em, . ~ . + L(chicken, 1) + L(chicken, 2) + L(chicken, 3))
  waldtest(em, em2)

  ## eggs granger-cause chickens?
  cm <- dynlm(chicken ~ L(chicken, 1) + L(chicken, 2) + L(chicken, 3), data = ChickEgg)
  cm2 <- update(cm, . ~ . + L(egg, 1) + L(egg, 2) + L(egg, 3))
  waldtest(cm, cm2)
}
```

`coeftest` *Testing Estimated Coefficients*

Description

`coeftest` is a generic function for performing z and (quasi-)t tests of estimated coefficients.

Usage

```
coeftest(x, vcov. = NULL, df = NULL, ...)
```

Arguments

`x`	an object (for details see below).
`vcov.`	a specification of the covariance matrix of the estimated coefficients. This can be specified as a matrix or as a function yielding a matrix when applied to `x`.
`df`	the degrees of freedom to be used. If this is a finite positive number a t test with `df` degrees of freedom is performed. In all other cases, a z test (using a normal approximation) is performed. By default it tries to use `x$df.residual` and performs a z test if this is `NULL`.
`...`	further arguments passed to the methods.

Details

The generic function `coeftest` currently has a default method (which works in particular for `"lm"` and `"glm"` objects) and a method for objects of class `"breakpointsfull"` (as computed by `breakpointsfull`).

The default method assumes that a `coef` methods exists, such that `coef(x)` yields the estimated coefficients.

To specify a covariance matrix `vcov.` to be used, there are three possibilities: 1. It is pre-computed and supplied in argument `vcov.`. 2. A function for extracting the covariance matrix from `x` is supplied, e.g., `vcovHC` or `vcovHAC` from package **sandwich**. 3. `vcov.` is set to `NULL`, then it is assumed that a `vcov` method exists, such that `vcov(x)` yields a covariance matrix. For illustrations see below.

The degrees of freedom `df` determine whether a normal approximation is used or a t distribution with `df` degrees of freedoms is used. The default method uses `df.residual(x)` and if this is `NULL` a z test is performed.

Value

An object of class `"coeftest"` which is essentially a coefficient matrix with columns containing the estimates, associated standard errors, test statistics and p values.

See Also

`lm`, `waldtest`

Examples

```
## load data and fit model
data(Mandible)
fm <- lm(length ~ age, data=Mandible, subset=(age <= 28))

## the following commands lead to the same tests:
summary(fm)
coeftest(fm)

## a z test (instead of a t test) can be performed by
coeftest(fm, df = Inf)
```

```
if(require(sandwich)) {
## a different covariance matrix can be also used:
## either supplied as a function
coeftest(fm, df = Inf, vcov = vcovHC)
## or as a matrix
coeftest(fm, df = Inf, vcov = vcovHC(fm, type = "HC0"))
}
```

---

coxtest *Cox Test for Comparing Non-Nested Models*

Description

`coxtest` performs the Cox test for comparing two non-nested models.

Usage

```
coxtest(formula1, formula2, data = list())
```

Arguments

`formula1`	either a symbolic description for the first model to be tested, or a fitted object of class `"lm"`.
`formula2`	either a symbolic description for the second model to be tested, or a fitted object of class `"lm"`.
`data`	an optional data frame containing the variables in the model. By default the variables are taken from the environment which `coxtest` is called from.

Details

The idea of the Cox test is the following: if the first model contains the correct set of regressors, then a fit of the regressors from the second model to the fitted values from first model should have no further explanatory value. But if it has, it can be concluded that model 1 does not contain the correct set of regressors.

Hence, to compare both models the fitted values of model 1 are regressed on model 2 and vice versa. A Cox test statistic is computed for each auxiliary model which is asymptotically standard normally distributed.

For further details, see the references.

Value

An object of class `"anova"` which contains the estimate plus corresponding standard error, z test statistic and p value for each auxiliary test.

References

R. Davidson & J. MacKinnon (1981). Several Tests for Model Specification in the Presence of Alternative Hypotheses. *Econometrica*, **49**, 781-793.

W. H. Greene (1993), *Econometric Analysis*, 2nd ed. Macmillan Publishing Company, New York.

W. H. Greene (2003). *Econometric Analysis*, 5th ed. New Jersey, Prentice Hall.

See Also

`jtest`, `encomptest`

Examples

```
## Fit two competing, non-nested models for aggregate
## consumption, as in Greene (1993), Examples 7.11 and 7.12

## load data and compute lags
data(USDistLag)
usdl <- na.contiguous(cbind(USDistLag, lag(USDistLag, k = -1)))
colnames(usdl) <- c("con", "gnp", "con1", "gnp1")

## C(t) = a0 + a1*Y(t) + a2*C(t-1) + u
fm1 <- lm(con ~ gnp + con1, data = usdl)

## C(t) = b0 + b1*Y(t) + b2*Y(t-1) + v
fm2 <- lm(con ~ gnp + gnp1, data = usdl)

## Cox test in both directions:
coxtest(fm1, fm2)

## ...and do the same for jtest() and encomptest().
## Notice that in this particular case they are coincident.
jtest(fm1, fm2)
encomptest(fm1, fm2)
```

`currencysubstitution` *Currency Substitution*

Description

Currency Subsitution Data.

Usage

```
data(currencysubstitution)
```

Format

A multivariate quarterly time series from 1960(4) to 1975(4) with variables

logCUS logarithm of the ratio of canadian holdings of Canadian dollar balances and canadian holdings of U.S. dollar balances,

Iu yield on U.S. Treasury bills,

Ic yield on Canadian Treasury bills,

logY logarithm of Canadian real gross national product.

Source

The data was originally studied by Miles (1978), the data set is given in Krämer and Sonnberger (1986). Below we replicate a few examples from their book. Some of these results differ more or less seriously and are sometimes parameterized differently.

References

M. Miles (1978), Currency Substitution, Flexible Exchange Rates, and Monetary Independence. *American Economic Review* **68**, 428–436

W. Krämer & H. Sonnberger (1986), *The Linear Regression Model under Test*. Heidelberg: Physica

Examples

```
data(currencysubstitution)

## page 130, fit Miles OLS model and Bordo-Choudri OLS model
## third and last line in Table 6.3

modelMiles <- logCUS ~ log((1+Iu)/(1+Ic))
lm(modelMiles, data=currencysubstitution)
dwtest(modelMiles, data=currencysubstitution)

modelBordoChoudri <- logCUS ~ I(Iu-Ic) + Ic + logY
lm(modelBordoChoudri, data=currencysubstitution)
dwtest(modelBordoChoudri, data=currencysubstitution)

## page 131, fit test statistics in Table 6.4 b)
################################################

if(require(strucchange, quietly = TRUE)) {
## Fluctuation test
sctest(modelMiles, type="fluctuation", data=currencysubstitution,
 rescale=FALSE) }

## RESET
reset(modelMiles, data=currencysubstitution)
reset(modelMiles, power=2, type="regressor", data=currencysubstitution)
reset(modelMiles, type="princomp", data=currencysubstitution)

## Harvey-Collier
```

```
harvtest(modelMiles, order.by = ~log((1+Iu)/(1+Ic)), data=currencysubstitution)

## Rainbow
raintest(modelMiles, order.by = "mahalanobis", data=currencysubstitution)

## page 132, fit test statistics in Table 6.4 d)
################################################

if(require(strucchange, quietly = TRUE)) {
## Chow 1970(2)
sctest(modelBordoChoudri, point=c(1970,2), data=currencysubstitution,
 type="Chow") }

## Breusch-Pagan
bptest(modelBordoChoudri, data=currencysubstitution, studentize=FALSE)
bptest(modelBordoChoudri, data=currencysubstitution)

## RESET
reset(modelBordoChoudri, data=currencysubstitution)
reset(modelBordoChoudri, power=2, type="regressor", data=currencysubstitution)
reset(modelBordoChoudri, type="princomp", data=currencysubstitution)

## Harvey-Collier
harvtest(modelBordoChoudri, order.by = ~ I(Iu-Ic), data=currencysubstitution)
harvtest(modelBordoChoudri, order.by = ~ Ic, data=currencysubstitution)
harvtest(modelBordoChoudri, order.by = ~ logY, data=currencysubstitution)

## Rainbow
raintest(modelBordoChoudri, order.by = "mahalanobis", data=currencysubstitution)
```

`dwtest` *Durbin-Watson Test*

Description

Performs the Durbin-Watson test for autocorrelation of disturbances.

Usage

```
dwtest(formula, order.by = NULL, alternative = c("greater", "two.sided", "less"),
       iterations = 15, exact = NULL, tol = 1e-10, data = list())
```

Arguments

`formula`	a symbolic description for the model to be tested (or a fitted `"lm"` object).
`order.by`	Either a vector `z` or a formula with a single explanatory variable like `~ z`. The observations in the model are ordered by the size of `z`. If set to `NULL` (the default) the observations are assumed to be ordered (e.g., a time series).
`alternative`	a character string specifying the alternative hypothesis.

iterations	an integer specifying the number of iterations when calculating the p-value with the "pan" algorithm.
exact	logical. If set to FALSE a normal approximation will be used to compute the p value, if TRUE the "pan" algorithm is used. The default is to use "pan" if the sample size is < 100.
tol	tolerance. Eigenvalues computed have to be greater than tol to be treated as non-zero.
data	an optional data frame containing the variables in the model. By default the variables are taken from the environment which dwtest is called from.

Details

The Durbin-Watson test has the null hypothesis that the autocorrelation of the disturbances is 0. It is possible to test against the alternative that it is greater than, not equal to, or less than 0, respectively. This can be specified by the alternative argument.

Under the assumption of normally distributed disturbances, the null distribution of the Durbin-Watson statistic is the distribution of a linear combination of chi-squared variables. The p-value is computed using a Fortran version of the Applied Statistics Algorithm AS 153 by Farebrother (1980, 1984). This algorithm is called "pan" or "gradsol". For large sample sizes the algorithm might fail to compute the p value; in that case a warning is printed and an approximate p value will be given; this p value is computed using a normal approximation with mean and variance of the Durbin-Watson test statistic.

Examples can not only be found on this page, but also on the help pages of the data sets bondyield, currencysubstitution, growthofmoney, moneydemand, unemployment, wages.

For an overview on R and econometrics see Racine & Hyndman (2002).

Value

An object of class "htest" containing:

statistic	the test statistic.
p.value	the corresponding p-value.
method	a character string with the method used.
data.name	a character string with the data name.

References

J. Durbin & G.S. Watson (1950), Testing for Serial Correlation in Least Squares Regression I. *Biometrika* **37**, 409–428.

J. Durbin & G.S. Watson (1951), Testing for Serial Correlation in Least Squares Regression II. *Biometrika* **38**, 159–178.

J. Durbin & G.S. Watson (1971), Testing for Serial Correlation in Least Squares Regression III. *Biometrika* **58**, 1–19.

R.W. Farebrother (1980), Pan's Procedure for the Tail Probabilities of the Durbin-Watson Statistic (Corr: 81V30 p189; AS R52: 84V33 p363- 366; AS R53: 84V33 p366- 369). *Applied Statistics* **29**, 224–227.

R. W. Farebrother (1984), [AS R53] A Remark on Algorithms AS 106 (77V26 p92-98), AS 153 (80V29 p224-227) and AS 155: The Distribution of a Linear Combination of χ^2 Random Variables (80V29 p323-333) *Applied Statistics* **33**, 366–369.

W. Krämer & H. Sonnberger (1986), *The Linear Regression Model under Test*. Heidelberg: Physica.

J. Racine & R. Hyndman (2002), Using R To Teach Econometrics. *Journal of Applied Econometrics* **17**, 175–189.

See Also

lm

Examples

```
## generate two AR(1) error terms with parameter
## rho = 0 (white noise) and rho = 0.9 respectively
err1 <- rnorm(100)

## generate regressor and dependent variable
x <- rep(c(-1,1), 50)
y1 <- 1 + x + err1

## perform Durbin-Watson test
dwtest(y1 ~ x)

err2 <- filter(err1, 0.9, method="recursive")
y2 <- 1 + x + err2
dwtest(y2 ~ x)
```

---

encomptest — *Encompassing Test for Comparing Non-Nested Models*

Description

`encomptest` performs the encompassing test of Davidson & MacKinnon for comparing non-nested models.

Usage

```
encomptest(formula1, formula2, data = list(), vcov. = NULL, ...)
```

Arguments

`formula1`	either a symbolic description for the first model to be tested, or a fitted object of class `"lm"`.
`formula2`	either a symbolic description for the second model to be tested, or a fitted object of class `"lm"`.

data	an optional data frame containing the variables in the model. By default the variables are taken from the environment which encomptest is called from.
vcov.	a function for estimating the covariance matrix of the regression coefficients, e.g., vcovHC.
...	further arguments passed to waldtest.

Details

To compare two non-nested models, the encompassing test fits an encompassing model which contains all regressors from both models such that the two models are nested within the encompassing model. A Wald test for comparing each of the models with the encompassing model is carried out by waldtest.

For further details, see the references.

Value

An object of class "anova" which contains the residual degrees of freedom in the encompassing model, the difference in degrees of freedom, Wald statistic (either "F" or "Chisq") and corresponding p value.

References

R. Davidson & J. MacKinnon (1993). *Estimation and Inference in Econometrics*. New York, Oxford University Press.

W. H. Greene (1993), *Econometric Analysis*, 2nd ed. Macmillan Publishing Company, New York.

W. H. Greene (2003). *Econometric Analysis*, 5th ed. New Jersey, Prentice Hall.

See Also

coxtest, jtest

Examples

```
## Fit two competing, non-nested models for aggregate
## consumption, as in Greene (1993), Examples 7.11 and 7.12

## load data and compute lags
data(USDistLag)
usdl <- na.contiguous(cbind(USDistLag, lag(USDistLag, k = -1)))
colnames(usdl) <- c("con", "gnp", "con1", "gnp1")

## C(t) = a0 + a1*Y(t) + a2*C(t-1) + u
fm1 <- lm(con ~ gnp + con1, data = usdl)

## C(t) = b0 + b1*Y(t) + b2*Y(t-1) + v
fm2 <- lm(con ~ gnp + gnp1, data = usdl)

## Encompassing model
fm3 <- lm(con ~ gnp + con1 + gnp1, data = usdl)
```

```
## Cox test in both directions:
coxtest(fm1, fm2)

## ...and do the same for jtest() and encomptest().
## Notice that in this particular case they are coincident.
jtest(fm1, fm2)
encomptest(fm1, fm2)

## the encompassing test is essentially
waldtest(fm1, fm3, fm2)
```

`ftemp` *Femal Temperature Data*

Description

Daily morning temperature of adult female (in degrees Celsius).

Usage

```
data(ftemp)
```

Format

Univariate daily time series of 60 observations starting from 1990-07-11.

Details

The data gives the daily morning temperature of an adult woman measured in degrees Celsius at about 6.30am each morning.

At the start of the period the woman was sick, hence the high temperature. Then the usual monthly cycle can be seen. On the second cycle, the temperature doesn't complete the downward part of the pattern due to a conception.

Source

The data set is taken from the Time Series Data Library at

http://www-personal.buseco.monash.edu.au/~hyndman/TSDL/

maintained by Rob Hyndman and Muhammad Akram.

Examples

```
data(ftemp)
plot(ftemp)
y <- window(ftemp, start = 8, end = 60)
if(require(strucchange)) {
  bp <- breakpoints(y ~ 1)
  plot(bp)
  fm.seg <- lm(y ~ 0 + breakfactor(bp))
  plot(y)
  lines(8:60, fitted(fm.seg), col = 4)
  lines(confint(bp))
}
```

---

gqtest — *Goldfeld-Quandt Test*

Description

Goldfeld-Quandt test against heteroskedasticity.

Usage

```
gqtest(formula, point = 0.5, fraction = 0, alternative = c("greater", "two.sided",
  order.by = NULL, data = list())
```

Arguments

`formula` a symbolic description for the model to be tested (or a fitted `"lm"` object).

`point` numerical. If `point` is smaller than 1 it is interpreted as percentages of data, i.e. `n*point` is taken to be the (potential) breakpoint in the variances, if `n` is the number of observations in the model. If `point` is greater than 1 it is interpreted to be the index of the breakpoint.

`fraction` numerical. The number of central observations to be omitted. If `fraction` is smaller than 1, it is chosen to be `fraction*n` if `n` is the number of observations in the model.

`alternative` a character string specifying the alternative hypothesis. The default is to test for increasing variances.

`order.by` Either a vector `z` or a formula with a single explanatory variable like `~ z`. The observations in the model are ordered by the size of `z`. If set to `NULL` (the default) the observations are assumed to be ordered (e.g., a time series).

`data` an optional data frame containing the variables in the model. By default the variables are taken from the environment which `gqtest` is called from.

Details

The Goldfeld-Quandt test compares the variances of two submodels divided by a specified breakpoint and rejects if the variances differ.

Under H_0 the test statistic of the Goldfeld-Quandt test follows an F distribution with the degrees of freedom as given in `parameter`.

Examples can not only be found on this page, but also on the help pages of the data sets `bondyield`, `currencysubstitution`, `growthofmoney`, `moneydemand`, `unemployment`, `wages`.

Value

A list with class `"htest"` containing the following components:

`statistic`	the value of the test statistic.
`p.value`	the p-value of the test.
`parameter`	degrees of freedom.
`method`	a character string indicating what type of test was performed.
`data.name`	a character string giving the name(s) of the data.

References

S.M. Goldfeld & R.E. Quandt (1965), Some Tests for Homoskedasticity. *Journal of the American Statistical Association* **60**, 539–547

W. Krämer & H. Sonnberger (1986), *The Linear Regression Model under Test*. Heidelberg: Physica

See Also

`lm`

Examples

```
## generate a regressor
x <- rep(c(-1,1), 50)
## generate heteroskedastic and homoskedastic disturbances
err1 <- c(rnorm(50, sd=1), rnorm(50, sd=2))
err2 <- rnorm(100)
## generate a linear relationship
y1 <- 1 + x + err1
y2 <- 1 + x + err2
## perform Goldfeld-Quandt test
gqtest(y1 ~ x)
gqtest(y2 ~ x)
```

`grangertest` *Test for Granger Causality*

Description

`grangertest` is a generic function for performing a test for Granger causality.

Usage

```
## Default S3 method:
grangertest(x, y, order = 1, na.action = na.omit, ...)
## S3 method for class 'formula':
grangertest(formula, data = list(), ...)
```

Arguments

`x`	either a bivariate series (in which case `y` has to be missing) or a univariate series of observations.
`y`	a univariate series of observations (if `x` is univariate, too).
`order`	integer specifying th order of lags to include in the auxiliary regression.
`na.action`	a function for eliminating `NA`s after aligning the series `x` and `y`.
`...`	further arguments passed to `waldtest`.
`formula`	a formula specification of a bivariate series like `y ~ x`.
`data`	an optional data frame containing the variables in the model. By default the variables are taken from the environment which `grangertest` is called from.

Details

Currently, the methods for the generic function `grangertest` only perform tests for Granger causality in bivariate series. The test is simply a Wald test comparing the unrestricted model—in which `y` is explained by the lags (up to order `order`) of `y` and `x`—and the restricted model—in which `y` is only explained by the lags of `y`.

Both methods are simply convenience interfaces to `waldtest`.

Value

An object of class `"anova"` which contains the residual degrees of freedom, the difference in degrees of freedom, Wald statistic and corresponding p value.

See Also

`waldtest`, `ChickEgg`

Examples

```
## Which came first: the chicken or the egg?
data(ChickEgg)
grangertest(egg ~ chicken, order = 3, data = ChickEgg)
grangertest(chicken ~ egg, order = 3, data = ChickEgg)

## alternative ways of specifying the same test
grangertest(ChickEgg, order = 3)
grangertest(ChickEgg[, 1], ChickEgg[, 2], order = 3)
```

`growthofmoney` *Growth of Money Supply*

Description

Growth of Money Supply Data.

Usage

```
data(growthofmoney)
```

Format

A multivariate quarterly time series from 1970(2) to 1974(4) with variables

TG1.TG0 difference of current and preceding target for the growth rate of the money supply,

AG0.TG0 difference of actual growth rate and target growth rate for the preceding period.

Source

The data was originally studied by Hetzel (1981), the data set is given in Krämer and Sonnberger (1986). Below we replicate a few examples from their book. Some of these results differ more or less seriously and are sometimes parameterized differently.

References

R.L. Hetzel (1981), The Federal Reserve System and Control of the Money Supply in the 1970's. *Journal of Money, Credit and Banking* **13**, 31–43

W. Krämer & H. Sonnberger (1986), *The Linear Regression Model under Test*. Heidelberg: Physica

Examples

```
data(growthofmoney)

## page 137, fit Hetzel OLS model
## first/second line in Table 6.7

modelHetzel <- TG1.TG0 ~ AG0.TG0
```

```
lm(modelHetzel, data=growthofmoney)
dwtest(modelHetzel, data=growthofmoney)

## page 135, fit test statistics in Table 6.8
#############################################

if(require(strucchange, quietly = TRUE)) {
## Chow 1974(1)
sctest(modelHetzel, point=c(1973,4), data=growthofmoney, type="Chow") }

## RESET
reset(modelHetzel, data=growthofmoney)
reset(modelHetzel, power=2, type="regressor", data=growthofmoney)
reset(modelHetzel, type="princomp", data=growthofmoney)

## Harvey-Collier
harvtest(modelHetzel, order.by= ~ AG0.TG0, data=growthofmoney)

## Rainbow
raintest(modelHetzel, order.by = "mahalanobis", data=growthofmoney)

## Identification of outliers
#############################

## Figure 6.1
plot(modelHetzel, data=growthofmoney)
abline(v=0)
abline(h=0)
abline(coef(lm(modelHetzel, data=growthofmoney)), col=2)

## Table 6.7, last line
growthofmoney2 <- as.data.frame(growthofmoney[-c(5:6),])
lm(modelHetzel, data=growthofmoney2)
dwtest(modelHetzel, data=growthofmoney2)
```

`harvtest` *Harvey-Collier Test*

Description

Harvey-Collier test for linearity.

Usage

```
harvtest(formula, order.by = NULL, data = list())
```

Arguments

`formula` a symbolic description for the model to be tested (or a fitted `"lm"` object).

`order.by`	Either a vector `z` or a formula with a single explanatory variable like `~ z`. The observations in the model are ordered by the size of `z`. If set to `NULL` (the default) the observations are assumed to be ordered (e.g., a time series).
`data`	an optional data frame containing the variables in the model. By default the variables are taken from the environment which `harvtest` is called from.

Details

The Harvey-Collier test performs a t-test (with `parameter` degrees of freedom) on the recursive residuals. If the true relationship is not linear but convex or concave the mean of the recursive residuals should differ from 0 significantly.

Examples can not only be found on this page, but also on the help pages of the data sets `bondyield`, `currencysubstitution`, `growthofmoney`, `moneydemand`, `unemployment`, `wages`.

Value

A list with class `"htest"` containing the following components:

`statistic`	the value of the test statistic.
`p.value`	the p-value of the test.
`parameter`	degrees of freedom.
`method`	a character string indicating what type of test was performed.
`data.name`	a character string giving the name(s) of the data.

References

A. Harvey & P. Collier (1977), Testing for Functional Misspecification in Regression Analysis. *Journal of Econometrics* **6**, 103–119

W. Krämer & H. Sonnberger (1986), *The Linear Regression Model under Test*. Heidelberg: Physica

See Also

`lm`

Examples

```
# generate a regressor and dependent variable
x <- 1:50
y1 <- 1 + x + rnorm(50)
y2 <- y1 + 0.3*x^2

## perform Harvey-Collier test
harv <- harvtest(y1 ~ x)
harv
## calculate critical value vor 0.05 level
qt(0.95, harv$parameter)
harvtest(y2 ~ x)
```

hmctest *Harrison-McCabe test*

Description

Harrison-McCabe test for heteroskedasticity.

Usage

```
hmctest(formula, point = 0.5, order.by = NULL, simulate.p = TRUE, nsim = 1000,
  plot = FALSE, data = list())
```

Arguments

`formula`	a symbolic description for the model to be tested (or a fitted `"lm"` object).
`point`	numeric. If `point` is smaller than 1 it is interpreted as percentages of data, i.e. `n*point` is taken to be the (potential) breakpoint in the variances, if `n` is the number of observations in the model. If `point` is greater than 1 it is interpreted to be the index of the breakpoint.
`order.by`	Either a vector `z` or a formula with a single explanatory variable like `~ z`. The observations in the model are ordered by the size of `z`. If set to `NULL` (the default) the observations are assumed to be ordered (e.g., a time series).
`simulate.p`	logical. If `TRUE` a p value will be assessed by simulation, otherwise the p value is `NA`.
`nsim`	integer. Determins how many runs are used to simulate the p value.
`plot`	logical. If `TRUE` the test statistic for all possible breakpoints is plotted.
`data`	an optional data frame containing the variables in the model. By default the variables are taken from the environment which `hmctest` is called from.

Details

The Harrison-McCabe test statistic is the fraction of the residual sum of squares that relates to the fraction of the data before the breakpoint. Under H_0 the test statistic should be close to the size of this fraction, e.g. in the default case close to 0.5. The null hypothesis is reject if the statistic is too small.

Examples can not only be found on this page, but also on the help pages of the data sets `bondyield`, `currencysubstitution`, `growthofmoney`, `moneydemand`, `unemployment`, `wages`.

Value

A list with class `"htest"` containing the following components:

`statistic`	the value of the test statistic.
`p.value`	the simulated p-value of the test.
`method`	a character string indicating what type of test was performed.
`data.name`	a character string giving the name(s) of the data.

References

M.J. Harrison & B.P.M McCabe (1979), A Test for Heteroscedasticity based on Ordinary Least Squares Residuals. *Journal of the American Statistical Association* **74**, 494–499

W. Krämer & H. Sonnberger (1986), *The Linear Regression Model under Test*. Heidelberg: Physica

See Also

lm

Examples

```
## generate a regressor
x <- rep(c(-1,1), 50)
## generate heteroskedastic and homoskedastic disturbances
err1 <- c(rnorm(50, sd=1), rnorm(50, sd=2))
err2 <- rnorm(100)
## generate a linear relationship
y1 <- 1 + x + err1
y2 <- 1 + x + err2
## perform Harrison-McCabe test
hmctest(y1 ~ x)
hmctest(y2 ~ x)
```

---

jocci *U.S. Macroeconomic Time Series*

Description

Several macroeconomic time series from the U.S.

Usage

```
data(fyff)
data(gmdc)
data(ip)
data(jocci)
data(lhur)
data(pw561)
```

Format

All data sets are multivariate monthly time series from 1959(8) to 1993(12) (except 1993(10) for `jocci`) with variables

y original time series,

dy transformed times series (first differences or log first differences),

dy1 transformed series at lag 1,

dy2 transformed series at lag 2,

dy3 transformed series at lag 3,

dy4 transformed series at lag 4,

dy5 transformed series at lag 5,

dy6 transformed series at lag 6.

Details

The description from Stock & Watson (1996) for the time series (with the transformation used):

fyff interest rate (first differences),

gmdc pce, impl pr defl: pce (1987 = 100) (log first differences),

ip index of industrial production (log first differences),

jocci department of commerce commodity price index (log first differences),

lhur unemployment rate: all workers, 16 years & over (%, sa) (first differences),

pw561 producer price index: crude petroleum (82 = 100, nsa) (log first differences).

Stock & Watson (1996) fitted an AR(6) model to all transformed time series.

Source

Stock & Watson (1996) study the stability of 76 macroeconomic time series, which can be obtained from Mark W. Watson's homepage http://www.wws.princeton.edu/~mwatson/.

References

J.H. Stock & M.W. Watson (1996), Evidence on Structural Instability in Macroeconomic Time Series Relations. *Journal of Business & Economic Statistics* **14**, 11–30.

Examples

```
data(jocci)

dwtest(dy ~ 1, data = jocci)
bgtest(dy ~ 1, data = jocci)
ar6.model <- dy ~ dy1 + dy2 + dy3 + dy4 + dy5 +dy6
bgtest(ar6.model, data = jocci)

var.model <- ~ I(dy1^2) + I(dy2^2) + I(dy3^2) + I(dy4^2) + I(dy5^2) + I(dy6^2)
bptest(ar6.model, var.model, data = jocci)
```

`jtest` *J Test for Comparing Non-Nested Models*

Description

`jtest` performs the Davidson-MacKinnon J test for comparing non-nested models.

Usage

```
jtest(formula1, formula2, data = list(), vcov. = NULL, ...)
```

Arguments

`formula1`	either a symbolic description for the first model to be tested, or a fitted object of class `"lm"`.
`formula2`	either a symbolic description for the second model to be tested, or a fitted object of class `"lm"`.
`data`	an optional data frame containing the variables in the model. By default the variables are taken from the environment which `jtest` is called from.
`vcov.`	a function for estimating the covariance matrix of the regression coefficients, e.g., `vcovHC`.
`...`	further arguments passed to `coeftest`.

Details

The idea of the J test is the following: if the first model contains the correct set of regressors, then including the fitted values of the second model into the set of regressors should provide no significant improvement. But if it does, it can be concluded that model 1 does not contain the correct set of regressors.

Hence, to compare both models the fitted values of model 1 are included into model 2 and vice versa. The J test statistic is simply the marginal test of the fitted values in the augmented model. This is performed by `coeftest`.

For further details, see the references.

Value

An object of class `"anova"` which contains the coefficient estimate of the fitted values in the augmented regression plus corresponding standard error, test statistic and p value.

References

R. Davidson & J. MacKinnon (1981). Several Tests for Model Specification in the Presence of Alternative Hypotheses. *Econometrica*, **49**, 781-793.

W. H. Greene (1993), *Econometric Analysis*, 2nd ed. Macmillan Publishing Company, New York.

W. H. Greene (2003). *Econometric Analysis*, 5th ed. New Jersey, Prentice Hall.

See Also

coxtest, encomptest

Examples

```
## Fit two competing, non-nested models for aggregate
## consumption, as in Greene (1993), Examples 7.11 and 7.12

## load data and compute lags
data(USDistLag)
usdl <- na.contiguous(cbind(USDistLag, lag(USDistLag, k = -1)))
colnames(usdl) <- c("con", "gnp", "con1", "gnp1")

## C(t) = a0 + a1*Y(t) + a2*C(t-1) + u
fm1 <- lm(con ~ gnp + con1, data = usdl)

## C(t) = b0 + b1*Y(t) + b2*Y(t-1) + v
fm2 <- lm(con ~ gnp + gnp1, data = usdl)

## Cox test in both directions:
coxtest(fm1, fm2)

## ...and do the same for jtest() and encomptest().
## Notice that in this particular case they are coincident.
jtest(fm1, fm2)
encomptest(fm1, fm2)
```

---

lrtest *Likelihood Ratio Test of Nested Models*

Description

lrtest is a generic function for carrying out likelihood ratio tests. The default method can be employed for comparing nested (generalized) linear models (see details below).

Usage

```
lrtest(object, ...)

## Default S3 method:
lrtest(object, ..., name = NULL)

## S3 method for class 'formula':
lrtest(object, ..., data = list())
```

Arguments

`object`	an object. See below for details.
`...`	further object specifications passed to methods. See below for details.
`name`	a function for extracting a suitable name/description from a fitted model object. By default the name is queried by calling `formula`.
`data`	a data frame containing the variables in the model.

Details

`lrtest` is intended to be a generic function for comparisons of models via asympotic likelihood ratio tests. The default method consecutively compares the fitted model object `object` with the models passed in `...`. Instead of passing the fitted model objects in `...`, several other specifications are possible. The updating mechanism is the same as for `waldtest`: the models in `...` can be specified as integers, characters (both for terms that should be eliminated from the previous model), update formulas or fitted model objects. Except for the last case, the existence of an `update` method is assumed. See `waldtest` for details.

Subsequently, an asymptotic likelihood ratio test for each two consecutive models is carried out: Twice the difference in log-likelihoods (as derived by the `logLik` methods) is compared with a Chi-squared distribution.

The `"formula"` method fits a `lm` first and then calls the default method.

Value

An object of class `"anova"` which contains the log-likelihood, degrees of freedom, the difference in degrees of freedom, likelihood ratio Chi-squared statistic and corresponding p value.

See Also

`waldtest`

Examples

```
## with data from Greene (1993):
## load data and compute lags
data("USDistLag")
usdl <- na.contiguous(cbind(USDistLag, lag(USDistLag, k = -1)))
colnames(usdl) <- c("con", "gnp", "con1", "gnp1")

fm1 <- lm(con ~ gnp + gnp1, data = usdl)
fm2 <- lm(con ~ gnp + con1 + gnp1, data = usdl)

## various equivalent specifications of the LR test
lrtest(fm2, fm1)
lrtest(fm2, 2)
lrtest(fm2, "con1")
lrtest(fm2, . ~ . - con1)
```

Mandible *Mandible Data*

Description

Mandible Data.

Usage

```
data(Mandible)
```

Format

Data from 167 fetuses, especially:

age gestational age in weeks.

length mandible length in mm.

Source

The data was originally published by Chitty et al., 1993, and analysed in Royston and Altman, 1994 (the data is given there). Only measurments with `age <= 28` were used in this analysis.

References

L. S. Chitty and S. Campbell and D. G. Altman (1993), Measurement of the fetal mandible – feasibility and construction of a centile chart., *Prenatal Diagnosis*, **13**, 749–756.

P. Royston and D. G. Altman (1994), Regression Using Fractional Polynomials of Continuous Covariates: Parsimonious Parametric Modelling. *Applied Statistics*, **43**, 429–453.

Examples

```
data(Mandible)
lm(length ~ age, data=Mandible, subset=(age <= 28))
```

moneydemand *Money Demand*

Description

Money Demand Data.

Usage

```
data(moneydemand)
```

Format

A multivariate yearly time series from 1879 to 1974 with variables

logM logarithm of quantity of money,

logYp logarithm of real permanent income,

Rs short term interest rate,

Rm rate of return on money,

Rl not documented in the sources,

logSpp logarithm of an operational measure of the variability of the rate of price changes.

Source

The data was originally studied by Allen (1982), the data set is given in Krämer and Sonnberger (1986). Below we replicate a few examples from the book. Some of these results differ more or less seriously and are sometimes parameterized differently.

References

S.D. Allen (1982), Kleins's Price Variability Terms in the U.S. Demand for Money. *Journal of Money, Credit and Banking* **14**, 525–530

W. Krämer & H. Sonnberger (1986), *The Linear Regression Model under Test*. Heidelberg: Physica

Examples

```
data(moneydemand)
moneydemand <- window(moneydemand, start=1880, end=1972)

## page 125, fit Allen OLS model (and Durbin-Watson test),
## last line in Table 6.1

modelAllen <- logM ~ logYp + Rs + Rl + logSpp
lm(modelAllen, data = moneydemand)
dwtest(modelAllen, data = moneydemand)

## page 127, fit test statistics in Table 6.1 c)
################################################

## Breusch-Pagan
bptest(modelAllen, studentize = FALSE, data = moneydemand)
bptest(modelAllen, studentize = TRUE, data = moneydemand)

## RESET
reset(modelAllen, data = moneydemand)
reset(modelAllen, power = 2, type = "regressor", data = moneydemand)
reset(modelAllen, type = "princomp", data = moneydemand)

## Harvey-Collier tests (up to sign of the test statistic)
harvtest(modelAllen, order.by = ~logYp, data = moneydemand)
harvtest(modelAllen, order.by = ~Rs, data = moneydemand)
```

```
harvtest(modelAllen, order.by = ~Rl, data = moneydemand)
harvtest(modelAllen, order.by = ~logSpp, data = moneydemand)

## Rainbow test
raintest(modelAllen, order.by = "mahalanobis", data = moneydemand)

if(require(strucchange, quietly = TRUE)) {
## Chow (1913)
sctest(modelAllen, point=c(1913,1), data = moneydemand, type = "Chow") }

if(require(strucchange, quietly = TRUE)) {
## Fluctuation
sctest(modelAllen, type = "fluctuation", rescale = FALSE, data = moneydemand)}
```

`petest` *PE Test for Linear vs. Log-Linear Specifications*

Description

`petest` performs the MacKinnon-White-Davidson PE test for comparing linear vs. log-linear specifications in linear regressions.

Usage

```
petest(formula1, formula2, data = list(), vcov. = NULL, ...)
```

Arguments

`formula1`	either a symbolic description for the first model to be tested, or a fitted object of class `"lm"`.
`formula2`	either a symbolic description for the second model to be tested, or a fitted object of class `"lm"`.
`data`	an optional data frame containing the variables in the model. By default the variables are taken from the environment which `petest` is called from.
`vcov.`	a function for estimating the covariance matrix of the regression coefficients, e.g., `vcovHC`.
`...`	further arguments passed to `coeftest`.

Details

The PE test compares two non-nest models where one has a linear specification of type `y ~ x1 + x2` and the other has a log-linear specification of type `log(y) ~ z1 + z2`. Typically, the regressors in the latter model are logs of the regressors in the former, i.e., `z1` is `log(x1)` etc.

The idea of the PE test is the following: If the linear specification is correct then adding an auxiliary regressor with the difference of the log-fitted values from both models should be non-significant. Conversely, if the log-linear specification is correct then adding an auxiliary regressor with the difference of fitted values in levels should be non-significant. The PE test statistic is simply the marginal test of the auxiliary variable(s) in the augmented model(s). In `petest` this is performed by `coeftest`.

For further details, see the references.

Value

An object of class `"anova"` which contains the coefficient estimate of the auxiliary variables in the augmented regression plus corresponding standard error, test statistic and p value.

References

J. MacKinnon, H. White, R. Davidson (1983). Tests for Model Specification in the Presence of Alternative Hypotheses: Some Further Results. *Journal of Econometrics*, **21**, 53-70.

M. Verbeek (2004). *A Guide to Modern Econometrics*, 2nd ed. Chichester, UK: John Wiley.

See Also

`jtest`, `coxtest`, `encomptest`

Examples

```
if(require("AER")) {
## Verbeek (2004), Section 3
data("HousePrices", package = "AER")

### Verbeek (2004), Table 3.3
hp_lin <- lm(price ~ . , data = HousePrices)
summary(hp_lin)

### Verbeek (2004), Table 3.2
hp_log <- update(hp_lin, log(price) ~ . - lotsize + log(lotsize))
summary(hp_log)

## PE test
petest(hp_lin, hp_log)
}
```

`raintest` *Rainbow Test*

Description

Rainbow test for linearity.

Usage

```
raintest(formula, fraction = 0.5, order.by = NULL, center = NULL,
    data=list())
```

Arguments

formula	a symbolic description for the model to be tested (or a fitted "lm" object).
fraction	numeric. The percentage of observations in the subset is determined by fraction*n if n is the number of observations in the model.
order.by	Either a vector z or a formula with a single explanatory variable like ~ z. The observations in the model are ordered by the size of z. If set to NULL (the default) the observations are assumed to be ordered (e.g., a time series). If set to "mahalanobis" then the observations are ordered by their Mahalanobis distances from the mean regressor.
center	numeric. If center is smaller than 1 it is interpreted as percentages of data, i.e. the subset is chosen that n*fraction observations are around observation number n*center. If center is greater than 1 it is interpreted to be the index of the center of the subset. By default center is 0.5. If the Mahalanobis distance is chosen center is taken to be the mean regressor, but can be specified to be a k-dimensional vector if k is the number of regressors and should be in the range of the respective regressors.
data	an optional data frame containing the variables in the model. By default the variables are taken from the environment which raintest is called from.

Details

The basic idea of the Rainbow test is that even if the true relationship is non-linear, a good linear fit can be achieved on a subsample in the "middle" of the data. The null hypothesis is rejected whenever the overall fit is significantly worse than the fit for the subsample. The test statistic under H_0 follows an F distribution with parameter degrees of freedom.

Examples can not only be found on this page, but also on the help pages of the data sets bondyield, currencysubstitution, growthofmoney, moneydemand, unemployment, wages.

Value

A list with class "htest" containing the following components:

statistic	the value of the test statistic.
p.value	the p-value of the test.
parameter	degrees of freedom.
method	a character string indicating what type of test was performed.
data.name	a character string giving the name(s) of the data.

References

J.M. Utts (1982), The Rainbow Test for Lack of Fit in Regression. *Communications in Statistics - Theory and Methods* **11**, 1801–1815

W. Krämer & H. Sonnberger (1986), *The Linear Regression Model under Test*. Heidelberg: Physica

See Also

lm

Examples

```
x <- c(1:30)
y <- x^2 + rnorm(30,0,2)
rain <- raintest(y ~ x)
rain
## critical value
qf(0.95, rain$parameter[1], rain$parameter[2])
```

`resettest` *RESET Test*

Description

Ramsey's RESET test for functional form.

Usage

```
resettest(formula, power = 2:3, type = c("fitted", "regressor",
  "princomp"), data = list())
```

Arguments

`formula`	a symbolic description for the model to be tested (or a fitted `"lm"` object).
`power`	integers. A vector of positive integers indicating the powers of the variables that should be included. By default, the test is for quadratic or cubic influence of the fitted response.
`type`	a string indicating whether powers of the fitted response, the regressor variables (factors are left out), or the first principal component of the regressor matrix should be included in the extended model.
`data`	an optional data frame containing the variables in the model. By default the variables are taken from the environment which `resettest` is called from.

Details

The RESET test is a popular diagnostic for correctness of functional form. The basic assumption is that under the alternative the model can be written in the form $y = X\beta + Z\gamma + u$. `Z` is generated by taking powers either of the fitted response, the regressor variables, or the first principal component of `X`. A standard F-Test is then applied to determine whether these additional variables have significant influence. The test statistic under H_0 follows an F distribution with `parameter` degrees of freedom.

This function was called `reset` in previous versions of the package. Please use `resettest` instead.

Examples can not only be found on this page, but also on the help pages of the data sets `bondyield`, `currencysubstitution`, `growthofmoney`, `moneydemand`, `unemployment`, `wages`.

Value

An object of class "htest" containing:

statistic	the test statistic.
p.value	the corresponding p-value.
parameter	degrees of freedom.
method	a character string with the method used.
data.name	a character string with the data name.

References

J.B. Ramsey (1969), Tests for Specification Error in Classical Linear Least Squares Regression Analysis. *Journal of the Royal Statistical Society, Series B* **31**, 350–371

W. Krämer & H. Sonnberger (1986), *The Linear Regression Model under Test*. Heidelberg: Physica

See Also

lm

Examples

```
x <- c(1:30)
y1 <- 1 + x + x^2 + rnorm(30)
y2 <- 1 + x + rnorm(30)
resettest(y1 ~ x , power=2, type="regressor")
resettest(y2 ~ x , power=2, type="regressor")
```

---

unemployment *Unemployment Data*

Description

Unemployment Data.

Usage

```
data(unemployment)
```

Format

A multivariate yearly time series from 1890 to 1979 with variables

UN unemployment rate,

m broad money supply,

p implicit deflator of Gross National Product,

G real purchases of goods and services,

x real exports.

Source

The data was originally studied by Rea (1983), the data set is given in Krämer and Sonnberger (1986). Below we replicate a few examples from their book. Some of these results differ more or less seriously and are sometimes parameterized differently.

References

J.D. Rea (1983), The Explanatory Power of Alternative Theories of Inflation and Unemployment, 1895-1979. *Review of Economics and Statistics* **65**, 183–195

W. Krämer & H. Sonnberger (1986), *The Linear Regression Model under Test*. Heidelberg: Physica

Examples

```
data(unemployment)

## data transformation
myunemployment <- window(unemployment, start=1895, end=1956)
time <- 6:67

## page 144, fit Rea OLS model
## last line in Table 6.12

modelRea <- UN ~ log(m/p) + log(G) + log(x) + time
lm(modelRea, data = myunemployment)
## coefficients of logged variables differ by factor 100

## page 143, fit test statistics in table 6.11
##############################################

if(require(strucchange, quietly = TRUE)) {
## Chow 1941
sctest(modelRea, point=c(1940,1), data=myunemployment, type="Chow") }

## Breusch-Pagan
bptest(modelRea, data=myunemployment, studentize=FALSE)
bptest(modelRea, data=myunemployment)

## RESET (a)-(b)
reset(modelRea, data=myunemployment)
reset(modelRea, power=2, type="regressor", data=myunemployment)

## Harvey-Collier
harvtest(modelRea, order.by = ~ log(m/p), data=myunemployment)
harvtest(modelRea, order.by = ~ log(G), data=myunemployment)
harvtest(modelRea, order.by = ~ log(x), data=myunemployment)
harvtest(modelRea, data=myunemployment)

## Rainbow
raintest(modelRea, order.by = "mahalanobis", data=myunemployment)
```

`USDistLag` *US Macroecnomic Data*

Description

US macroeconomic data for fitting a distributed lag model.

Usage

```
data(USDistLag)
```

Format

An annual time series from 1963 to 1982 with 2 variables.

consumption real consumption,

gnp gross national product (deflated by CPI).

Source

Table 7.7 in Greene (1993)

References

Greene W.H. (1993), *Econometric Analysis*, 2nd edition. Macmillan Publishing Company, New York.

Executive Office of the President (1983), *Economic Report of the President*. US Government Printing Office, Washington, DC.

Examples

```
## Willam H. Greene, Econometric Analysis, 2nd Ed.
## Chapter 7
## load data set, p. 221, Table 7.7
data(USDistLag)

## fit distributed lag model, p.221, Example 7.8
usdl <- na.contiguous(cbind(USDistLag, lag(USDistLag, k = -1)))
colnames(usdl) <- c("con", "gnp", "con1", "gnp1")

## C(t) = b0 + b1*Y(t) + b2*C(t-1) + u
fm <- lm(con ~ gnp + con1, data = usdl)
summary(fm)
vcov(fm)
```

valueofstocks *Value of Stocks*

Description

Value of Stocks Data

Usage

```
data(valueofstocks)
```

Format

A multivariate quarterly time series from 1960(1) to 1977(3) with variables

VST value of stocks,

MB monetary base,

RTPD dollar rent on producer durables,

RTPS dollar rent on producer structures,

XBC production capacity for business output.

Source

The data was originally studied by Woglom (1981), the data set is given in Krämer and Sonnberger (1986).

References

G. Woglom (1981), A Reexamination of the Role of Stocks in the Consumption Function and the Transmission Mechanism. *Journal of Money, Credit and Banking* **13**, 215–220

W. Krämer & H. Sonnberger (1986), *The Linear Regression Model Under Test*. Heidelberg: Physica

Examples

```
data(valueofstocks)
lm(log(VST) ~., data=valueofstocks)
```

wages *Wages*

Description

Wages Data.

Usage

```
data(wages)
```

Format

A multivariate yearly time series from 1960 to 1979 with variables

w wages,

CPI consumer price index,

u unemployment,

mw minimum wage.

Source

The data was originally studied by Nicols (1983), the data set is given in Krämer and Sonnberger (1986). Below we replicate a few examples from their book. Some of these results differ more or less seriously and are sometimes parameterized differently.

References

D.A. Nicols (1983), Macroeconomic Determinants of Wage Adjustments in White Collar Occupations. *Review of Economics and Statistics* **65**, 203–213

W. Krämer & H. Sonnberger (1986), *The Linear Regression Model under Test*. Heidelberg: Physica

Examples

```
data(wages)

## data transformation to include lagged series
mywages <- cbind(wages, lag(wages[,2], k = -1), lag(wages[,2], k = -2))
colnames(mywages) <- c(colnames(wages), "CPI2", "CPI3")
mywages <- window(mywages, start=1962, end=1979)

## page 142, fit Nichols OLS model
## equation (6.10)

modelNichols <- w ~ CPI + CPI2 + CPI3 + u + mw
lm(modelNichols, data = mywages)

## page 143, fit test statistics in table 6.11
```

```
###############################################

if(require(strucchange, quietly = TRUE)) {
## Chow 1972
sctest(modelNichols, point=c(1971,1), data=mywages, type="Chow") }

## Breusch-Pagan
bptest(modelNichols, data=mywages, studentize=FALSE)
bptest(modelNichols, data=mywages)

## RESET (a)-(b)
reset(modelNichols, data=mywages)
reset(modelNichols, power=2, type="regressor", data=mywages)

## Harvey-Collier
harvtest(modelNichols, order.by = ~ CPI, data=mywages)
harvtest(modelNichols, order.by = ~ CPI2, data=mywages)
harvtest(modelNichols, order.by = ~ CPI3, data=mywages)
harvtest(modelNichols, order.by = ~ u, data=mywages)

## Rainbow
raintest(modelNichols, order.by = "mahalanobis", data=mywages)
```

`waldtest` *Wald Test of Nested Models*

Description

`waldtest` is a generic function for carrying out Wald tests. The default method can be employed for comparing nested (generalized) linear models (see details below).

Usage

```
waldtest(object, ...)

## Default S3 method:
waldtest(object, ..., vcov = NULL,
  test = c("Chisq", "F"), name = NULL)

## S3 method for class 'formula':
waldtest(object, ..., data = list())

## S3 method for class 'lm':
waldtest(object, ..., test = c("F", "Chisq"))
```

Arguments

`object`	an object. See below for details.
`...`	further object specifications passed to methods. See below for details.

vcov	a function for estimating the covariance matrix of the regression coefficients, e.g., vcovHC. If only two models are compared it can also be the covariance matrix of the more general model.
test	character specifying wether to compute the large sample Chi-squared statistic (with asymptotic Chi-squared distribution) or the finite sample F statistic (with approximate F distribution).
name	a function for extracting a suitable name/description from a fitted model object. By default the name is queried by calling formula.
data	a data frame containing the variables in the model.

Details

waldtest is intended to be a generic function for comparisons of models via Wald tests. The default method consecutively compares the fitted model object object with the models passed in .... Instead of passing the fitted model objects in ..., several other specifications are possible. For all objects in list(object, ...) the function tries to consecutively compute fitted models using the following updating algorithm:

1. For each two consecutive objects, object1 and object2 say, try to turn object2 into a fitted model that can be compared to (the already fitted model object) object1.
2. If object2 is numeric, the corresponding element of attr(terms(object1), "term.labels") is selected to be omitted.
3. If object2 is a character, the corresponding terms are included into an update formula like . ~ . - term2a - term2b.
4. If object2 is a formula, then compute the fitted model via update(object1, object2).

Consequently, the models in ... can be specified as integers, characters (both for terms that should be eliminated from the previous model), update formulas or fitted model objects. Except for the last case, the existence of an update method is assumed. See also the examples for an illustration.

Subsequently, a Wald test for each two consecutive models is carried out. This is similar to anova (which typically performs likelihood-ratio tests), but with a few differences. If only one fitted model object is specified, it is compared to the trivial model (with only an intercept). The test can be either the finite sample F statistic or the asymptotic Chi-squared statistic ($F = Chisq/k$ if k is the difference in degrees of freedom). The covariance matrix is always estimated on the more general of two subsequent models (and not only in the most general model overall). If vcov is specified, HC and HAC estimators can also be plugged into waldtest.

The default method is already very general and applicable to a broad range of fitted model objects, including lm and glm objects. It can be easily made applicable to other model classes as well by providing suitable methods to the standard generics terms (for deterimining the variables in the model along with their names), update (unless only fitted model objects are passed to waldtest, as mentioned above), residuals (only used for determining the number of observations), df.residual (needed for the F statisic), coef (for extracting the coefficients; needs to be named matching the names in terms), vcov (can be user-supplied; needs to be named matching the names in terms). Furthermore, some means of determining a suitable name for a fitted model object can be specified (by default this is taken to be the result of a call to formula).

The "formula" method fits a lm first and then calls the "lm" method. The "lm" method just calls the default method, but sets the default test to be the F test.

Value

An object of class "anova" which contains the residual degrees of freedom, the difference in degrees of freedom, Wald statistic (either "Chisq" or "F") and corresponding p value.

See Also

coeftest, anova, linear.hypothesis

Examples

```
## fit two competing, non-nested models and their encompassing
## model for aggregate consumption, as in Greene (1993),
## Examples 7.11 and 7.12

## load data and compute lags
data(USDistLag)
usdl <- na.contiguous(cbind(USDistLag, lag(USDistLag, k = -1)))
colnames(usdl) <- c("con", "gnp", "con1", "gnp1")

## C(t) = a0 + a1*Y(t) + a2*C(t-1) + u
fm1 <- lm(con ~ gnp + con1, data = usdl)

## C(t) = b0 + b1*Y(t) + b2*Y(t-1) + v
fm2 <- lm(con ~ gnp + gnp1, data = usdl)

## Encompassing model
fm3 <- lm(con ~ gnp + con1 + gnp1, data = usdl)

## a simple ANOVA for fm3 vs. fm2
waldtest(fm3, fm2)
anova(fm3, fm2)
## as df = 1, the test is equivalent to the corresponding t test in
coeftest(fm3)

## various equivalent specifications of the two models
waldtest(fm3, fm2)
waldtest(fm3, 2)
waldtest(fm3, "con1")
waldtest(fm3, . ~ . - con1)

## comparing more than one model
## (euqivalent to the encompassing test)
waldtest(fm1, fm3, fm2)
encomptest(fm1, fm2)

## using the asymptotic Chisq statistic
waldtest(fm3, fm2, test = "Chisq")
## plugging in a HC estimator
if(require(sandwich)) waldtest(fm3, fm2, vcov = vcovHC)
```

Index