Package 'skedastic'

June 8, 2025

Type Package

Title Handling Heteroskedasticity in the Linear Regression Model

Version 2.0.3

Description Implements numerous methods for testing for, modelling, and
correcting for heteroskedasticity in the classical linear regression
model. The most novel contribution of the
package is found in the functions that implement the as-yet-unpublished
auxiliary linear variance models and auxiliary nonlinear variance
models that are designed to estimate error variances in a heteroskedastic
linear regression model. These models follow principles of statistical
learning described in Hastie (2009) <doi:10.1007/978-0-387-21606-5>.
The nonlinear version of the model is estimated using quasi-likelihood
methods as described in Seber and Wild (2003, ISBN: 0-471-47135-6).
Bootstrap methods for approximate confidence intervals for error variances
are implemented as described in Efron and Tibshirani
(1993, ISBN: 978-1-4899-4541-9), including also the expansion technique
described in Hesterberg (2014) <doi:10.1080/00031305.2015.1089789>. The
wild bootstrap employed here follows the description in Davidson and
Flachaire (2008) <doi:10.1016/j.jeconom.2008.08.003>. Tuning of
hyper-parameters makes use of a golden section search function that is
modelled after the MATLAB function of Zarnowiec (2022)
<https://www.mathworks.com/matlabcentral/fileexchange/
25919-golden-section-method-algorithm>.
A methodological description of the algorithm can be found in Fox (2021,
ISBN: 978-1-003-00957-3).
There are 25 different functions that implement hypothesis tests for
heteroskedasticity. These include a test based on Anscombe (1961)
<https://projecteuclid.org/euclid.bsmsp/1200512155>, Ramsey's (1969)
BAMSET Test <doi:10.1111/j.2517-6161.1969.tb00796.x>, the tests of Bickel
(1978) <doi:10.1214/aos/1176344124>, Breusch and Pagan (1979)
<doi:10.2307/1911963> with and without the modification
proposed by Koenker (1981) <doi:10.1016/0304-4076(81)90062-2>, Carapeto and
Holt (2003) <doi:10.1080/0266476022000018475>, Cook and Weisberg (1983)
<doi:10.1093/biomet/70.1.1> (including their graphical methods), Diblasi
and Bowman (1997) <doi:10.1016/S0167-7152(96)00115-0>, Dufour, Khalaf,

Bernard, and Genest (2004) <doi:10.1016/j.jeconom.2003.10.024>, Evans and King (1985) <doi:10.1016/0304-4076(85)90085-5> and Evans and King (1988) <doi:10.1016/0304-4076(88)90006-1>, Glejser (1969) <doi:10.1080/01621459.1969.10500976> as formulated by Mittelhammer, Judge and Miller (2000, ISBN: 0-521-62394-4), Godfrey and Orme (1999) <doi:10.1080/07474939908800438>, Goldfeld and Quandt (1965) <doi:10.1080/01621459.1965.10480811>, Harrison and McCabe (1979) <doi:10.1080/01621459.1979.10482544>, Harvey (1976) <doi:10.2307/1913974>, Honda (1989) <doi:10.1111/j.2517-6161.1989.tb01749.x>, Horn (1981) <doi:10.1080/03610928108828074>, Li and Yao (2019) <doi:10.1016/j.ecosta.2018.01.001> with and without the modification of Bai, Pan, and Yin (2016) <doi:10.1007/s11749-017-0575-x>, Rackauskas and Zuokas (2007) <doi:10.1007/s10986-007-0018-6>, Simonoff and Tsai (1994) <doi:10.2307/2986026> with and without the modification of Ferrari, Cysneiros, and Cribari-Neto (2004) <doi:10.1016/S0378-3758(03)00210-6>, Szroeter (1978) <doi:10.2307/1913831>, Verbyla (1993) <doi:10.1111/j.2517-6161.1993.tb01918.x>, White (1980) <doi:10.2307/1912934>, Wilcox and Keselman (2006) <doi:10.1080/10629360500107923>, Yuce (2008) <https://dergipark.org.tr/en/pub/iuekois/issue/8989/112070>, and Zhou, Song, and Thompson (2015) <doi:10.1002/cjs.11252>. Besides these heteroskedasticity tests, there are supporting functions that compute the BLUS residuals of Theil (1965) <doi:10.1080/01621459.1965.10480851>, the conditional two-sided p-values of Kulinskaya (2008) <doi:10.48550/arXiv.0810.2124>, and probabilities for the nonparametric trend statistic of Lehmann (1975, ISBN: 0-816-24996-1). For handling heteroskedasticity, in addition to the new auxiliary variance model methods, there is a function to implement various existing Heteroskedasticity-Consistent Covariance Matrix Estimators from the literature, such as those of White (1980) <doi:10.2307/1912934>, MacKinnon and White (1985) <doi:10.1016/0304-4076(85)90158-7>, Cribari-Neto (2004) <doi:10.1016/S0167-9473(02)00366-3>, Cribari-Neto et al. (2007) <doi:10.1080/03610920601126589>, Cribari-Neto and da Silva (2011) <doi:10.1007/s10182-010-0141-2>, Aftab and Chang (2016) <doi:10.18187/pjsor.v12i2.983>, and Li et al. (2017) <doi:10.1080/00949655.2016.1198906>.

RdMacros Rdpack

Depends R (>= 3.6.0),

Imports Rdpack (>= 0.11.1), broom (>= 0.5.6), pracma (>= 2.2.9), CompQuadForm (>= 1.4.3), MASS (>= 7.3.47), quadprog (>= 1.5.8), inflection (>= 1.3.5), Rfast (>= 2.0.6), caret (>= 6.0.90), Matrix (>= 1.4.1), quadprogXT (>= 0.0.5), slam (>= 0.1.49), ROI (>= 1.0.0), osqp (>= 0.6.0.5), mgcv (>= 1.8.40), ROI.plugin.qpoases (>= 1.0.2),

Suggests knitr, rmarkdown, devtools, lmtest, car, tseries, tibble, testthat, mlbench, expm, arrangements, quantreg, gmp, Rmpfr, cubature, mvtnorm, lmboot, sandwich, cmna

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

URL https://github.com/tjfarrar/skedastic

BugReports https://github.com/tjfarrar/skedastic/issues

NeedsCompilation no

Author Thomas Farrar [aut, cre] (0000-0003-0744-6972),
University of the Western Cape [cph]

Maintainer Thomas Farrar <tjfarrar@alumni.uwaterloo.ca>

Repository CRAN

Date/Publication 2025-06-07 23:30:02 UTC

Contents

---

alvm.fit   *Auxiliary Linear Variance Model*

Description

Fits an Auxiliary Linear Variance Model (ALVM) to estimate the error variances of a heteroskedastic linear regression model.

Usage

```
alvm.fit(
  mainlm,
  M = NULL,
 model = c("cluster", "spline", "linear", "polynomial", "basic", "homoskedastic"),
  varselect = c("none", "hettest", "cv.linear", "cv.cluster", "qgcv.linear",
    "qgcv.cluster"),
  lambda = c("foldcv", "qgcv"),
  nclust = c("elbow.swd", "elbow.mwd", "elbow.both", "foldcv"),
  clustering = NULL,
  polypen = c("L2", "L1"),
  d = 2L,
  solver = c("auto", "quadprog", "quadprogXT", "roi", "osqp"),
  tsk = NULL,
  tsm = NULL,
  constol = 1e-10,
  cvoption = c("testsetols", "partitionres"),
  nfolds = 5L,
  reduce2homosked = TRUE,
  ...
)
```

Arguments

mainlm	Either an object of class "lm" (e.g., generated by [lm](lm)), or a list of two objects: a response vector and a design matrix. The objects are assumed to be in that order, unless they are given the names "X" and "y" to distinguish them. The design matrix passed in a list must begin with a column of ones if an intercept is to be included in the linear model. The design matrix passed in a list should not contain factors, as all columns are treated 'as is'. For tests that use ordinary least squares residuals, one can also pass a vector of residuals in the list, which should either be the third object or be named "e".
M	An $n \times n$ annihilator matrix. If NULL (the default), this will be calculated from the mainlm object
model	A character corresponding to the type of ALVM to be fitted: "cluster" for the clustering ALVM, "spline" for the thin-plate spline ALVM, "linear" for the linear ALVM, "polynomial" for the penalised polynomial ALVM, "basic" for the basic or naive ALVM, and "homoskedastic" for the homoskedastic error variance estimator, $e'e/(n-p)$.
varselect	Either a character indicating how variable selection should be conducted, or an integer vector giving indices of columns of the predictor matrix (model.matrix of mainlm) to select. The vector must include 1L for the intercept to be selected. If a character, it must be one of the following:

- "none": No variable selection is conducted;
- "hettest": Variable selection is conducted by applying a heteroskedasticity test with each feature in turn serving as the 'deflator' variable
- "cv.linear": Variable selection is conducted by best subset selection on the auxiliary linear variance model (linear specification), using squared-error loss computed under K-fold cross-validation
- "cv.cluster": Variable selection is conducted by best subset selection on the auxiliary linear variance model (clustering specification), using squared-error loss computed under K-fold cross-validation
- "qgcv.linear": Variable selection is conducted by best subset selection on the auxiliary linear variance model (linear specification), using squared-error loss computed under quasi-generalised cross-validation
- "qgcv.cluster": Variable selection is conducted by best subset selection on the auxiliary linear variance model (clustering specification), using squared-error loss computed under quasi-generalised cross-validation

lambda	Either a double of length 1 indicating the value of the penalty hyperparameter λ, or a character specifying the tuning method for choosing λ: "foldcv" for K-fold cross-validation (the default) or "qgcv" for quasi-generalised cross-validation. This argument is ignored if model is neither "polynomial" nor "spline".
nclust	Either an integer of length 1 indicating the value of the number of clusters n_c, or a character specifying the tuning method for choosing n_c: "elbow.swd" for the elbow method using a sum of within-cluster distances criterion, "elbow.mwd" for the elbow method using a maximum within-cluster distances criterion, "elbow.both" for rounded average of the results of the previous two, and "foldcv" for K-fold cross-validation. This argument is ignored if model is not "cluster".

clustering	A list object of class ″doclust″. If set to NULL (the default), such an object is generated (ignored if cluster is FALSE)
polypen	A character, either ″L2″ or ″L1″, indicating whether an L_2 norm penalty (ridge regression) or L_1 norm penalty (LASSO) should be used with the polynomial model. This argument is ignored if model is not ″polynomial″.
d	An integer specifying the degree of polynomial to use in the penalised polynomial ALVM; defaults to 2L. Ignored if model is other than ″polynomial″. Setting d to 1L is not identical to setting model to ″linear″, because the linear ALVM does not have a penalty term in the objective function.
solver	A character, indicating which Quadratic Programming solver function to use to estimate γ. The options are ″quadprog″, corresponding to solve.QP.compact from the **quadprog** package; package; ″quadprogXT″, corresponding to buildQP from the **quadprogXT** package; ″roi″, corresponding to the qpoases solver implemented in ROI_solve from the **ROI** package with **ROI.plugin.qpoases** add-on; and ″osqp″, corresponding to solve_osqp from the **osqp** package. Alternatively, the user can specify ″auto″ (the default), in which case the function will select the solver that seems to work best for the chosen model.
tsk	An integer corresponding to the basis dimension k to be passed to the [mgcv]{s} function for fitting of a thin-plate spline ALVM; see [mgcv]{choose.k} for more details about choosing the parameter and defaults. Ignored if model is not ″spline″.
tsm	An integer corresponding to the order m of the penalty to be passed to the [mgcv]{s} function for fitting of a thin-plate spline ALVM. If left as the default (NULL), it will be set to 2, corresponding to 2nd derivative penalties for a cubic spline.
constol	A double corresponding to the boundary value for the constraint on error variances. Of course, the error variances must be non-negative, but setting the constraint boundary to 0 can result in zero estimates that then result in infinite weights for Feasible Weighted Least Squares. The boundary value should thus be positive, but small enough not to bias estimation of very small variances. Defaults to 1e-10.
cvoption	A character, either ″testsetols″ or ″partitionres″, indicating how to obtain the observed response values for each test fold when performing K-fold cross-validation on an ALVM. The default technique, ″testsetols″, entails fitting a linear regression model to the test fold of observations from the original response vector y and predictor matrix X. The squared residuals from this regression are the observed responses that are predicted from the trained model to compute the cross-validated squared error loss function. Under the other technique, ″partitionres″, the squared residuals from the full linear regression model are partitioned into training and test folds and the squared residuals in the test fold are the observed responses that are predicted for computation of the cross-validated loss.
nfolds	An integer specifying the number of folds K to use for cross-validation, if the λ and/or n_c hyperparameters are to be tuned using cross-validation. Defaults to 5L. One must ensure that each test fold contains at least $p+1$ observations if the ″testsetols″ technique is used with cross-validation, so that there are enough degrees of freedom to fit a linear model to the test fold.

reduce2homosked

>A logical indicating whether the homoskedastic error variance estimator $e'e/(n-p)$ should be used if the variable selection procedure does not select any variables. Defaults to TRUE.

... Other arguments that can be passed to (non-exported) helper functions, namely:

- greedy, a logical passed to the functions implementing best subset selection, indicating whether or not to use a greedy search rather than exhaustive search for the best subset. Defaults to FALSE, but coerced to TRUE unconditionally if $p > 9$.
- distmetric, a character specifying the distance metric to use in computing distance for the clustering algorithm. Corresponds to the method argument of [dist] and defaults to "euclidean"
- linkage, a character specifying the linkage rule to use in agglomerative hierarchical clustering. Corresponds to the method argument of [hclust] and defaults to "complete"
- nclust2search, an integer vector specifying the values of n_c to try when tuning n_c by cross-validation. Defaults to 1L:20L
- alpha, a double specifying the significance level threshold to use when applying heteroskedasticity test for the purpose of feature selection in an ALVM; defaults to 0.1
- testname, a character corresponding to the name of a function that performs a heteroskedasticity test. The function must either be one that takes a deflator argument or [breusch_pagan]. Defaults to evans_king

Details

The ALVM model equation is

$$e \circ e = (M \circ M)L\gamma + u$$

, where e is the Ordinary Least Squares residual vector, M is the annihilator matrix $M = I - X(X'X)^{-1}X'$, L is a linear predictor matrix, u is a random error vector, γ is a p-vector of unknown parameters, and $\circ$ denotes the Hadamard (elementwise) product. The construction of L depends on the method used to model or estimate the assumed heteroskedastic function $g(\cdot)$, a continuous, differentiable function that is linear in γ and by which the error variances ω_i of the main linear model are related to the predictors $X_i.$. This method has been developed as part of the author's doctoral research project.

Depending on the model used, the estimation method could be Inequality-Constrained Least Squares or Inequality-Constrained Ridge Regression. However, these are both special cases of Quadratic Programming. Therefore, all of the models are fitted using Quadratic Programming.

Several techniques are available for feature selection within the model. The LASSO-type model handles feature selection via a shrinkage penalty. For this reason, if the user calls the polynomial model with L_1-norm penalty, it is not necessary to specify a variable selection method, since this is handled automatically. Another feature selection technique is to use a heteroskedasticity test that tests for heteroskedasticity linked to a particular predictor variable (the 'deflator'). This test can be conducted with each features in turn serving as the deflator. Those features for which the null hypothesis of homoskedasticity is rejected at a specified significance level alpha are selected. A third feature selection technique is best subset selection, where the model is fitted with all possible subsets of features. The models are scored in terms of some metric, and the best-performing

subset of features is selected. The metric could be squared-error loss computed under K-fold cross-validation or using quasi-generalised cross-validation. (The *quasi-* prefix refers to the fact that generalised cross-validation is, properly speaking, only applicable to a linear fitting method, as defined by Hastie et al. (2009). ALVMs are not linear fitting methods due to the inequality constraint). Since best subset selection requires fitting 2^{p-1} models (where $p - 1$ is the number of candidate features), it is infeasible for large p. A greedy search technique can therefore be used as an alternative, where one begins with a null model and adds the feature that leads to the best improvement in the metric, stopping when no new feature leads to an improvement.

The polynomial and thin-plate spline ALVMs have a penalty hyperparameter λ that must either be specified or tuned. K-fold cross-validation or quasi-generalised cross-validation can be used for tuning. The clustering ALVM has a hyperparameter n_c, the number of clusters into which to group the observations (where error variances are assumed to be equal within each cluster). n_c can be specified or tuned. The available tuning methods are an elbow method (using a sum of within-cluster distances criterion, a maximum within-cluster distance criterion, or a combination of the two) and K-fold cross-validation.

Value

An object of class `"alvm.fit"`, containing the following:

- `coef.est`, a vector of parameter estimates, $\hat{\gamma}$
- `var.est`, a vector of estimates $\hat{\omega}$ of the error variances for all observations
- `method`, a character corresponding to the `model` argument
- `ols`, the `lm` object corresponding to the original linear regression model
- `fitinfo`, a list containing four named objects: `Msq` (the elementwise-square of the annihilator matrix M), `L` (the linear predictor matrix L), `clustering` (a list object with results of the clustering procedure), and `gam.object`, an object of class `"gam"` (see [gamObject](#)). The last two are set to NA unless the clustering ALVM or thin-plate spline ALVM is used, respectively
- `hyperpar`, a named list of hyperparameter values, `lambda`, `nclust`, `tsk`, and `d`, and tuning methods, `lambdamethod` and `nclustmethod`. Values corresponding to unused hyperparameters are set to NA.
- `selectinfo`, a list containing two named objects, `varselect` (the value of the eponymous argument), and `selectedcols` (a numeric vector with column indices of X that were selected, with 1 denoting the intercept column)
- `pentype`, a character corresponding to the `polypen` argument
- `solver`, a character corresponding to the `solver` argument (or specifying the QP solver actually used, if `solver` was set to `"auto"`)
- `constol`, a double corresponding to the `constol` argument

References

Hastie T, Tibshirani R, Friedman JH (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd edition. Springer, New York.

See Also

[alvm.fit](#), [avm.ci](#)

Examples

```
mtcars_lm <- lm(mpg ~ wt + qsec + am, data = mtcars)
myalvm <- alvm.fit(mtcars_lm, model = "cluster")
```

---

anlvm.fit *Auxiliary Nonlinear Variance Model*

Description

Fits an Auxiliary Nonlinear Variance Model (ANLVM) to estimate the error variances of a heteroskedastic linear regression model.

Usage

```
anlvm.fit(
  mainlm,
  g,
  M = NULL,
  cluster = FALSE,
  varselect = c("none", "hettest", "cv.linear", "cv.cluster", "qgcv.linear",
    "qgcv.cluster"),
  nclust = c("elbow.swd", "elbow.mwd", "elbow.both"),
  clustering = NULL,
  param.init = function(q) stats::runif(n = q, min = -5, max = 5),
  maxgridrows = 20L,
  nconvstop = 3L,
  zerosallowed = FALSE,
  maxitql = 100L,
  tolql = 1e-08,
  nestedql = FALSE,
  reduce2homosked = TRUE,
  cvoption = c("testsetols", "partitionres"),
  nfolds = 5L,
  ...
)
```

Arguments

mainlm   Either an object of class "lm" (e.g., generated by [lm](lm)), or a list of two objects:
                a response vector and a design matrix. The objects are assumed to be in that
                order, unless they are given the names "X" and "y" to distinguish them. The
                design matrix passed in a list must begin with a column of ones if an intercept
                is to be included in the linear model. The design matrix passed in a list should
                not contain factors, as all columns are treated 'as is'. For tests that use ordinary
                least squares residuals, one can also pass a vector of residuals in the list, which
                should either be the third object or be named "e".

g	A numeric-valued function of one variable, or a character denoting the name of such a function. "sq" is allowed as a way of denoting function(x) x ^ 2.
M	An $n \times n$ annihilator matrix. If NULL (the default), this will be calculated from the mainlm object
cluster	A logical; should the design matrix X be replaced with an $n \times n_c$ matrix of ones and zeroes, with a single one in each row, indicating assignments of the n observations to n_c clusters using an agglomerative hierarchical clustering algorithm. In this case, the dimensionality of γ is n_c and not p. Defaults to FALSE
varselect	Either a character indicating how variable selection should be conducted, or an integer vector giving indices of columns of the predictor matrix (model.matrix of mainlm) to select. The vector must include 1L for the intercept to be selected. If a character, it must be one of the following: • "none": No variable selection is conducted; • "hettest": Variable selection is conducted by applying a heteroskedasticity test with each feature in turn serving as the 'deflator' variable • "cv.linear": Variable selection is conducted by best subset selection on the auxiliary linear variance model (linear specification), using squared-error loss computed under K-fold cross-validation • "cv.cluster": Variable selection is conducted by best subset selection on the auxiliary linear variance model (clustering specification), using squared-error loss computed under K-fold cross-validation • "qgcv.linear": Variable selection is conducted by best subset selection on the auxiliary linear variance model (linear specification), using squared-error loss computed under quasi-generalised cross-validation • "qgcv.cluster": Variable selection is conducted by best subset selection on the auxiliary linear variance model (clustering specification), using squared-error loss computed under quasi-generalised cross-validation
nclust	A character indicating which elbow method to use to select the number of clusters (ignored if cluster is FALSE). Alternatively, an integer specifying the number of clusters
clustering	A list object of class "doclust". If set to NULL (the default), such an object is generated (ignored if cluster is FALSE)
param.init	Specifies the initial values of the parameter vector to use in the Gauss-Newton fitting algorithm. This can either be a function for generating the initial values from a probability distribution, a list containing named objects corresponding to the arguments of seq (specifying a sequence of scalar values that will be passed to expand.grid), or a numeric vector specifying a single initial parameter vector
maxgridrows	An integer indicating the maximum number of initial values of the parameter vector to try, in case of param.init being a function or a list used to generate a grid. Defaults to 20L.
nconvstop	An integer indicating how many times the quasi-likelihood estimation algorithm should converge before the grid search across different initial parameter values is truncated. Defaults to 3L. If nconvstop >= maxgridrows, no early stopping rule will be used.

zerosallowed	A logical indicating whether 0 values are acceptable in the initial values of the parameter vector. Defaults to FALSE.
maxitql	An integer specifying the maximum number of iterations to run in the Gauss-Newton algorithm for quasi-likelihood estimation. Defaults to 100L.
tolql	A double specifying the convergence criterion for the Gauss-Newton algorithm; defaults to 1e-8. The criterion is applied to the L_2 norm of the difference between parameter vectors in successive iterations.
nestedql	A logical indicating whether to use the nested updating step suggested in Seber and Wild (2003). Defaults to FALSE due to the large computation time required.
reduce2homosked	
	A logical indicating whether the homoskedastic error variance estimator $e'e/(n-p)$ should be used if the variable selection procedure does not select any variables. Defaults to TRUE.
cvoption	A character, either "testsetols" or "partitionres", indicating how to obtain the observed response values for each test fold when performing K-fold cross-validation on an ALVM. The default technique, "testsetols", entails fitting a linear regression model to the test fold of observations from the original response vector y and predictor matrix X. The squared residuals from this regression are the observed responses that are predicted from the trained model to compute the cross-validated squared error loss function. Under the other technique, "partitionres", the squared residuals from the full linear regression model are partitioned into training and test folds and the squared residuals in the test fold are the observed responses that are predicted for computation of the cross-validated loss.
nfolds	An integer specifying the number of folds K to use for cross-validation, if the λ and/or n_c hyperparameters are to be tuned using cross-validation. Defaults to 5L. One must ensure that each test fold contains at least $p+1$ observations if the "testsetols" technique is used with cross-validation, so that there are enough degrees of freedom to fit a linear model to the test fold.
...	Other arguments that can be passed to (non-exported) helper functions, namely:

- greedy, a logical passed to the functions implementing best subset selection, indicating whether or not to use a greedy search rather than exhaustive search for the best subset. Defaults to FALSE, but coerced to TRUE unconditionally if $p > 9$.
- distmetric, a character specifying the distance metric to use in computing distance for the clustering algorithm. Corresponds to the method argument of dist and defaults to "euclidean"
- linkage, a character specifying the linkage rule to use in agglomerative hierarchical clustering. Corresponds to the method argument of hclust and defaults to "complete"
- alpha, a double specifying the significance level threshold to use when applying heteroskedasticity test for the purpose of feature selection in an ALVM; defaults to 0.1
- testname, a character corresponding to the name of a function that performs a heteroskedasticity test. The function must either be one that takes a deflator argument or breusch_pagan. Defaults to evans_king

Details

The ANLVM model equation is

$$e_i^2 = \sum_{k=1}^{n} g(X_{k.}'\gamma) m_{ik}^2 + u_i$$

, where e_i is the ith Ordinary Least Squares residual, $X_{k.}$ is a vector corresponding to the kth row of the $n \times p$ design matrix X, m_{ik}^2 is the (i,k)th element of the annihilator matrix $M = I - X(X'X)^{-1}X'$, u_i is a random error term, γ is a p-vector of unknown parameters, and $g(\cdot)$ is a continuous, differentiable function that need not be linear in γ, but must be expressible as a function of the linear predictor $X_{k.}'\gamma$. This method has been developed as part of the author's doctoral research project.

The parameter vector γ is estimated using the maximum quasi-likelihood method as described in section 2.3 of Seber and Wild (2003). The optimisation problem is solved numerically using a Gauss-Newton algorithm.

For further discussion of feature selection and the methods for choosing the number of clusters to use with the clustering version of the model, see `alvm.fit`.

Value

An object of class `"anlvm.fit"`, containing the following:

- `coef.est`, a vector of parameter estimates, $\hat{\gamma}$
- `var.est`, a vector of estimates $\hat{\omega}$ of the error variances for all observations
- `method`, either `"cluster"` or `"functionalform"`, depending on whether `cluster` was set to `TRUE`
- `ols`, the `lm` object corresponding to the original linear regression model
- `fitinfo`, a list containing three named objects, `g` (the heteroskedastic function), `Msq` (the elementwise-square of the annihilator matrix M), `Z` (the design matrix used in the ANLVM, after feature selection if applicable), and `clustering` (a list object with results of the clustering procedure, if applicable).
- `selectinfo`, a list containing two named objects, `varselect` (the value of the eponymous argument), and `selectedcols` (a numeric vector with column indices of X that were selected, with 1 denoting the intercept column)
- `qlinfo`, a list containing nine named objects: `converged` (a logical, indicating whether the Gauss-Newton algorithm converged for at least one initial value of the parameter vector), `iterations` (the number of Gauss-Newton iterations used to obtain the parameter estimates returned), `Smin` (the minimum achieved value of the objective function used in the Gauss-Newton routine), and six arguments passed to the function (`nested`, `param.init`, `maxgridrows`, `nconvstop`, `maxitql`, and `tolql`)

References

Seber GAF, Wild CJ (2003). *Nonlinear Regression*. Wiley, Hoboken, NJ.

See Also

`alvm.fit`, `avm.ci`

Examples

```
mtcars_lm <- lm(mpg ~ wt + qsec + am, data = mtcars)
myanlvm <- anlvm.fit(mtcars_lm, g = function(x) x ^ 2,
 varselect = "qgcv.linear")
```

---

anscombe	*Anscombe's Test for Heteroskedasticity in a Linear Regression Model*

Description

This function implements the method of Anscombe (1961) for testing for heteroskedasticity in a linear regression model, with or without the studentising modification of Bickel (1978).

Usage

```
anscombe(mainlm, studentise = TRUE, statonly = FALSE)
```

Arguments

mainlm   Either an object of [class](#) "lm" (e.g., generated by [lm](#)), or a list of two objects: a response vector and a design matrix. The objects are assumed to be in that order, unless they are given the names "X" and "y" to distinguish them. The design matrix passed in a list must begin with a column of ones if an intercept is to be included in the linear model. The design matrix passed in a list should not contain factors, as all columns are treated 'as is'. For tests that use ordinary least squares residuals, one can also pass a vector of residuals in the list, which should either be the third object or be named "e".

studentise   A logical. Should studentising modification of Bickel (1978) be implemented? Defaults to TRUE; if FALSE, the original form of the test proposed by Anscombe (1961) is used.

statonly   A logical. If TRUE, only the test statistic value is returned, instead of an object of [class](#) "htest". Defaults to FALSE.

Details

Anscombe's Test is among the earliest suggestions for heteroskedasticity diagnostics in the linear regression model. The test is not based on formally derived theory but on a test statistic that Anscombe intuited to be approximately standard normal under the null hypothesis of homoskedasticity. Bickel (1978) discusses the test and suggests a studentising modification (included in this function) as well as a robustifying modification (included in [bickel](#)). The test is two-tailed.

Value

An object of [class](#) "htest". If object is not assigned, its attributes are displayed in the console as a [tibble](#) using [tidy](#).

References

Anscombe FJ (1961). "Examination of Residuals." In Neyman J (ed.), *Fourth Berkeley Symposium on Mathematical Statistics and Probability June 20-July 30, 1960*, 1–36. Berkeley: University of California Press.

Bickel PJ (1978). "Using Residuals Robustly I: Tests for Heteroscedasticity, Nonlinearity." *The Annals of Statistics*, **6**(2), 266–291.

See Also

bickel, which is a robust extension of this test.

Examples

```
mtcars_lm <- lm(mpg ~ wt + qsec + am, data = mtcars)
anscombe(mtcars_lm)
```

---

avm.ci   *Bootstrap Confidence Intervals for Linear Regression Error Variances*

Description

Uses bootstrap methods to compute approximate confidence intervals for error variances in a heteroskedastic linear regression model based on an auxiliary linear variance model (ALVM) or auxiliary nonlinear variance model (ANLVM).

Usage

```
avm.ci(
  object,
  bootobject = NULL,
  bootavmobject = NULL,
  jackobject = NULL,
  bootCImethod = c("pct", "bca", "stdnorm"),
  bootsampmethod = c("pairs", "wild"),
  Bextra = 500L,
  Brequired = 1000L,
  conf.level = 0.95,
  expand = TRUE,
  retune = FALSE,
  resfunc = c("identity", "hccme"),
  qtype = 6,
  rm_on_constraint = TRUE,
  rm_nonconverged = TRUE,
  jackknife_point = FALSE,
  ...
)
```

Arguments

object	An object of class "alvm.fit" or of class "anlvm.fit", containing information on a fitted ALVM or ANLVM
bootobject	An object of class "bootlm", containing information on a set of B bootstrapped versions of a linear regression model, obtained by a nonparametric bootstrap method suitable for heteroskedastic linear models. If set to NULL (the default), it is generated by calling [bootlm]().
bootavmobject	An object of class "bootavm", containing information on an ALVM or ANLVM fit to B bootstrapped linear regression models. If set to NULL (the default), it is generated by calling the non-exported function bootavm.
jackobject	An object of class "jackavm", containing information on ALVMs or ANVLMs fit to jackknife versions of a linear regression model. If set to NULL (the default), it is generated by calling the non-exported function jackavm.
bootCImethod	A character specifying the method to use when computing the approximate bootstrap confidence interval. The default, "pct", corresponds to the percentile interval. "bca" corresponds to the Bias-Corrected and accelerated (BCa) modification of the percentile interval. "stdnorm" corresponds to a naive standard normal interval with bootstrap standard error estimates.
bootsampmethod	A character specifying the method to use for generating nonparametric bootstrap linear regression models. Corresponds to the sampmethod argument of [bootlm]() and defaults to "pairs". **Warning:** in simulations, bootstrap intervals computed using the wild bootstrap have shown very poor coverage probabilities. Ignored unless bootobject is NULL.
Bextra	An integer indicating the maximum number of additional bootstrap models that should be fitted in an attempt to obtain Brequired appropriate sets of bootstrap variance estimates, as explained above under Brequired. Defaults to 500L. Ignored if rm_on_constraint is set to FALSE (for an ALVM) or if rm_nonconverged is set to FALSE (for an ANLVM).
Brequired	An integer indicating the number of bootstrap regression models that should be used to compute the bootstrap confidence intervals. The default behaviour is to base the interval estimates only on bootstrap ALVM variance estimates that are not on the constraint boundary or on bootstrap ANLVMs where the estimation algorithm converged. Consequently, if this is not the case for all of the first Brequired bootstrap models, additional bootstrap models are used (up to a maximum of Bextra). Defaults to 1000L.
conf.level	A double representing the confidence level $1 - \alpha$; must be between 0 and 1. Defaults to 0.95.
expand	A logical specifying whether to implement the expansion technique described in Hesterberg (2015). Defaults to TRUE.
retune	A logical specifying whether to re-tune hyperparameters and re-select features each time an ALVM or (in the case of feature selection) ANLVM is fit to a bootstrapped regression model. If FALSE (the default), the hyperparameter value and selected features from the ALVM fit to the original model are reused in every bootstrap model. Setting to TRUE is more theoretically sound but increases computation time substantially.

resfunc   Either a character naming a function to call to apply a transformation to the
                    Ordinary Least Squares residuals, or a function to apply for the same purpose.
                    This argument is ignored if sampmethod is "pairs". The only two character
                    values accepted are "identity", in which case no transformation is applied
                    to the residuals, and "hccme", in which case the transformation corresponds
                    to a heteroskedasticity-consistent covariance matrix estimator calculated from
                    hccme. If resfunc is a function, it is assumed that its first argument is the
                    numeric vector of residuals.

qtype   A numeric corresponding to the type argument of quantile. Defaults to 6.

rm_on_constraint

                    A logical specifying whether to exclude bootstrapped ALVMs from the interval
                    estimation method where the ALVM parameter estimate falls on the constraint
                    boundary. Defaults to TRUE.

rm_nonconverged

                    A logical specifying whether to exclude bootstrapped ANLVMs from the inter-
                    val estimation method where the optimisation algorithm used in quasi-likelihood
                    estimation of the ANLVM parameter did not converge. Defaults to TRUE.

jackknife_point

                    A logical specifying whether to replace the point estimates of the error vari-
                    ances ω with jackknife estimates based only on the leave-one-out auxiliary mod-
                    els where the parameter estimates do not lie on the constraint boundary (in the
                    ALVM case) or where the quasi-likelihood estimation algorithm converged (in
                    the ANLVM case). Defaults to FALSE.

...   Other arguments to pass to non-exported helper functions

Details

B resampled versions of the original linear regression model (which can be accessed using object$ols)
are generated using a nonparametric bootstrap method that is suitable for heteroskedastic linear re-
gression models, namely either the pairs bootstrap or the wild bootstrap (bootstrapping residuals is
not suitable). Depending on the class of object, either an ALVM or an ANLVM is fit to each of the
bootstrapped regression models. The distribution of the B bootstrap estimates of each error vari-
ance ω_i, $i = 1, 2, \ldots, n$, is used to construct an approximate confidence interval for ω_i. This is done
using one of three methods. The first is the percentile interval, which simply takes the empirical
$\alpha/2$ and $1 - \alpha/2$ quantiles of the ith bootstrap variance estimates. The second is the Bias-Corrected
and accelerated (BCa) method as described in Efron and Tibshirani (1993), which is intended to
improve on the percentile interval (although simulations have not found it to yield better coverage
probabilities). The third is the naive standard normal interval, which takes $\hat{\omega}_i \pm z_{1-\alpha/2}\widehat{\mathrm{SE}}$, where
$\widehat{\mathrm{SE}}$ is the standard deviation of the B bootstrap estimates of ω_i. By default, the expansion technique
described in Hesterberg (2015) is also applied; evidence from simulations suggests that this *does*
improve coverage probabilities.

Technically, the hyperparameters of the ALVM, such as λ (for a penalised polynomial or thin-plate
spline model) or n_c (for a clustering model) should be re-tuned every time the ALVM is fitted to
another bootstrapped regression model. However, due to the computational cost, this is not done by
avm.ci unless retune is set to TRUE.

When obtained from ALVMs, bootstrap estimates of ω_i that fall on the constraint boundary (i.e.,
are estimated to be near 0) are ignored by default; there is an attempt to obtain Brequired bootstrap

estimates of each ω_i that do not fall on the constraint boundary. This fine-tuning can be turned off by setting the `rm_onconstraint` argument to `FALSE`; the amount of effort put into obtaining non-boundary estimates is controlled using the `Bextra` argument. When ANLVMs are used, the default behaviour is to try to obtain `Brequired` bootstrap estimates of ω where the Gauss-Newton algorithm applied for quasi-likelihood estimation has converged, and ignore estimates obtained from non-convergent cases. This behaviour can be toggled using the `rm_nonconverged` argument.

Value

An object of class `"avm.ci"`, containing the following:

- `climits`, an $n \times 2$ matrix with lower confidence limits in the first column and upper confidence limits in the second
- `var.est`, a vector of length n of point estimates $\hat{\omega}$ of the error variances. This is the same vector passed within `object`, unless `jackknife_point` is `TRUE`.
- `conf.level`, corresponding to the eponymous argument
- `bootCImethod`, corresponding to the eponymous argument
- `bootsampmethod`, corresponding to the eponymous argument or otherwise extracted from `bootobject`

References

Efron B, Tibshirani RJ (1993). *An Introduction to the Bootstrap*. Springer Science+Business Media, Dordrecht.

Hesterberg T (2015). "What Teachers Should Know about the Bootstrap: Resampling in the Undergraduate Statistics Curriculum."

See Also

alvm.fit, anlvm.fit, Efron and Tibshirani (1993)

Examples

```
mtcars_lm <- lm(mpg ~ wt + qsec + am, data = mtcars)
myalvm <- alvm.fit(mtcars_lm, model = "cluster")
# Brequired would of course not be so small in practice
ci.alvm <- avm.ci(myalvm, Brequired = 5)
```

---

avm.fwls   *Apply Feasible Weighted Least Squares to a Linear Regression Model*

Description

This function applies feasible weighted least squares (FWLS) to a linear regression model using error variance estimates obtained from an auxiliary linear variance model fit using `alvm.fit` or from an auxiliary nonlinear variance model fit using `anlvm.fit`.

Usage

```
avm.fwls(object, fastfit = FALSE)
```

Arguments

object   Either an object of class "alvm.fit" or an object of class "anlvm.fit"

fastfit   A logical. If FALSE (the default), the linear regression model is fit using lm;
                otherwise, using lm.wfit

Details

The function simply calculates

$$\hat{\beta} = (X'\hat{\Omega}^{-1}X)^{-1}X'\hat{\Omega}^{-1}y$$

, where X is the design matrix, y is the response vector, and $\hat{\Omega}$ is the diagonal variance-covariance matrix of the random errors, whose diagonal elements have been estimated by an auxiliary variance model.

Value

Either an object of class "lm" (if fastfit is FALSE) or otherwise a generic list object

References

There are no references for Rd macro \insertAllCites on this help page.

See Also

alvm.fit, anlvm.fit, avm.vcov

Examples

```
mtcars_lm <- lm(mpg ~ wt + qsec + am, data = mtcars)
myalvm <- alvm.fit(mainlm = mtcars_lm, model = "linear",
    varselect = "qgcv.linear")
myfwls <- avm.fwls(myalvm)
cbind(coef(mtcars_lm), coef(myfwls))
```

---

avm.vcov   *Estimate Covariance Matrix of Ordinary Least Squares Estimators*
            Using Error Variance Estimates from an Auxiliary Variance Model

---

Description

The function simply calculates

$$\mathrm{Cov}\hat{\beta} = (X'X)^{-1}X'\hat{\Omega}X(X'X)^{-1}$$

, where X is the design matrix of a linear regression model and $\hat{\Omega}$ is an estimate of the diagonal variance-covariance matrix of the random errors, whose diagonal elements have been obtained from an auxiliary variance model fit with `alvm.fit` or `anlvm.fit`.

Usage

```
avm.vcov(object, as_matrix = TRUE)
```

Arguments

object   Either an object of class `"alvm.fit"` or an object of class `"anlvm.fit"`

as_matrix   A logical. If `TRUE` (the default), a $p \times p$ matrix is returned, where p is the number of columns in X. Otherwise, a numeric vector of length p is returned.

Value

Either a numeric matrix or a numeric vector, whose (diagonal) elements are $\widehat{\mathrm{Var}}(\hat{\beta}_j)$, $j = 1, 2, \ldots, p$.

References

There are no references for Rd macro \insertAllCites on this help page.

See Also

[alvm.fit](), [anlvm.fit](), [avm.fwls](). If a matrix is returned, it can be passed to [coeftest]() for implementation of a quasi-t-test of significance of the β coefficients.

Examples

```
mtcars_lm <- lm(mpg ~ wt + qsec + am, data = mtcars)
myalvm <- alvm.fit(mainlm = mtcars_lm, model = "linear",
   varselect = "qgcv.linear")
myvcov <- avm.vcov(myalvm)
lmtest::coeftest(mtcars_lm, vcov. = myvcov)
```

---

bamset   *Ramsey's BAMSET Test for Heteroskedasticity in a Linear Regression Model*

Description

This function implements the Bartlett's M Specification Error Test (BAMSET) method of Ramsey (1969) for testing for heteroskedasticity in a linear regression model.

Usage

```
bamset(
  mainlm,
  k = 3,
  deflator = NA,
  correct = TRUE,
  omitatmargins = TRUE,
  omit = NA,
  categorical = FALSE,
  statonly = FALSE
)
```

Arguments

mainlm   Either an object of class "lm" (e.g., generated by [lm](lm)), or a list of two objects:
               a response vector and a design matrix. The objects are assumed to be in that
               order, unless they are given the names "X" and "y" to distinguish them. The
               design matrix passed in a list must begin with a column of ones if an intercept
               is to be included in the linear model. The design matrix passed in a list should
               not contain factors, as all columns are treated 'as is'. For tests that use ordinary
               least squares residuals, one can also pass a vector of residuals in the list, which
               should either be the third object or be named "e".

k   An integer. The number of subsets (>= 2) into which the BLUS residuals are to
               be partitioned. Defaults to 3, the value suggested in Ramsey (1969).

deflator   Either a character specifying a column name from the design matrix of mainlm
               or an integer giving the index of a column of the design matrix. This variable
               is suspected to be related to the error variance under the alternative hypothesis.
               deflator may not correspond to a column of 1's (intercept). Default NA means
               the data will be left in its current order (e.g. in case the existing index is believed
               to be associated with error variance).

correct   A logical. Should the test statistic be divided by a scaling constant to improve
               the chi-squared approximation? Defaults to TRUE.

omitatmargins A logical. Should the indices of observations at the margins of the k subsets be
               passed to blus as the omit argument? If TRUE (the default), this overrides any
               omit argument passed directly. If FALSE, the omit argument must be specified
               and cannot be left as NA. If categorical is TRUE, setting omitatmargins to

TRUE results in omitting observations from the most frequently occurring factor levels or values.

omit   A numeric vector of length p (the number of columns in the linear model design matrix) giving the indices of p observations to omit in the BLUS residual vector; or a character partially matching `"first"` (for the first p) observations, `"last"` (for the last p observations), or `"random"` (for a random sample of p indices between 1 and n). Defaults to `"first"`.

categorical   A logical. Is the deflator a categorical variable? If so, the number of levels will be used as k with each level forming a subset. Defaults to FALSE.

statonly   A logical. If TRUE, only the test statistic value is returned, instead of an object of [class](#) `"htest"`. Defaults to FALSE.

Details

BAMSET is an analogue of Bartlett's M Test for heterogeneity of variances across independent samples from k populations. In this case the populations are k subsets of the residuals from a linear regression model. In order to meet the independence assumption, [BLUS residuals](#) are computed, meaning that only $n - p$ observations are used (where n is the number of rows and p the number of columns in the design matrix). Under the null hypothesis of homoskedasticity, the test statistic is asymptotically chi-squared distributed with $k - 1$ degrees of freedom. The test is right-tailed.

Value

An object of [class](#) `"htest"`. If object is not assigned, its attributes are displayed in the console as a [tibble](#) using [tidy](#).

References

Ramsey JB (1969). "Tests for Specification Errors in Classical Linear Least-Squares Regression Analysis." *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, **31**(2), 350–371.

Examples

```
mtcars_lm <- lm(mpg ~ wt + qsec + am, data = mtcars)
bamset(mtcars_lm, deflator = "qsec", k = 3)

# BLUS residuals cannot be computed with given `omit` argument and so
# omitted indices are randomised:
bamset(mtcars_lm, deflator = "qsec", k = 4, omitatmargins = FALSE, omit = "last")
```

---

bickel   *Bickel's Test for Heteroskedasticity in a Linear Regression Model*

Description

This function implements the method of Bickel (1978) for testing for heteroskedasticity in a linear regression model, with or without the scale-invariance modification of Carroll and Ruppert (1981).

Usage

```
bickel(
  mainlm,
  fitmethod = c("lm", "rlm"),
  a = "identity",
  b = c("hubersq", "tanhsq"),
  scale_invariant = TRUE,
  k = 1.345,
  statonly = FALSE,
  ...
)
```

Arguments

mainlm   Either an object of [class](#) "lm" (e.g., generated by [lm](#)), or a list of two objects:
                a response vector and a design matrix. The objects are assumed to be in that
                order, unless they are given the names "X" and "y" to distinguish them. The
                design matrix passed in a list must begin with a column of ones if an intercept
                is to be included in the linear model. The design matrix passed in a list should
                not contain factors, as all columns are treated 'as is'. For tests that use ordinary
                least squares residuals, one can also pass a vector of residuals in the list, which
                should either be the third object or be named "e".

fitmethod   A character indicating the method to be used to fit the regression model. This
                can be "OLS" for ordinary least squares (the default) or "robust" in which case
                a robust fitting method is called from [rlm](#).

a   A character argument specifying the name of a function to be applied to the
                fitted values, or an integer m in which case the function applied is $f(x) = x^m$.
                Defaults to "identity" for [identity](#).

b   A character argument specifying a function to be applied to the residuals. De-
                faults to Huber's function squared, as recommended by Carroll and Ruppert
                (1981). Currently the only supported functions are "hubersq" (for Huber's
                function squared) and "tanhsq" (for $b(x) = \tanh(x)^2$.)

scale_invariant
                A logical indicating whether the scale-invariance modification proposed by Car-
                roll and Ruppert (1981) should be implemented. Defaults to TRUE.

k	A double argument specifying a parameter for Huber's function squared; used only if b == "hubersq". This is not to be confused with the argument k2 that could be passed to rlm if the regression is fitted using robust methods. k defaults to 1.345.
statonly	A logical. If TRUE, only the test statistic value is returned, instead of an object of class "htest". Defaults to FALSE.
...	Optional arguments to be passed to rlm

Details

Bickel's Test is a robust extension of Anscombe's Test (Anscombe 1961) in which the OLS residuals and estimated standard error are replaced with an M estimator. Under the null hypothesis of homoskedasticity, the distribution of the test statistic is asymptotically standard normally distributed. The test is two-tailed.

Value

An object of class "htest". If object is not assigned, its attributes are displayed in the console as a tibble using tidy.

References

Anscombe FJ (1961). "Examination of Residuals." In Neyman J (ed.), *Fourth Berkeley Symposium on Mathematical Statistics and Probability June 20-July 30, 1960*, 1–36. Berkeley: University of California Press.

Bickel PJ (1978). "Using Residuals Robustly I: Tests for Heteroscedasticity, Nonlinearity." *The Annals of Statistics*, **6**(2), 266–291.

Carroll RJ, Ruppert D (1981). "On Robust Tests for Heteroscedasticity." *The Annals of Statistics*, **9**(1), 206–210.

See Also

discussions of this test in Carroll and Ruppert (1981) and Ali and Giaccotto (1984).

Examples

```
mtcars_lm <- lm(mpg ~ wt + qsec + am, data = mtcars)
bickel(mtcars_lm)
bickel(mtcars_lm, fitmethod = "rlm")
bickel(mtcars_lm, scale_invariant = FALSE)
```

---

blus   *Compute Best Linear Unbiased Scalar-Covariance (BLUS) residuals*
                               from a linear model

Description

This function computes the Best Linear Unbiased Scalar-Covariance (BLUS) residuals from a linear
model, as defined in Theil (1965) and explained further in Theil (1968).

Usage

```
blus(
  mainlm,
  omit = c("first", "last", "random"),
  keepNA = TRUE,
  exhaust = NA,
  seed = 1234
)
```

Arguments

mainlm   Either an object of [class](#) "lm" (e.g., generated by [lm](#)), or a list of two objects:
                a response vector and a design matrix. The objects are assumed to be in that
                order, unless they are given the names "X" and "y" to distinguish them. The
                design matrix passed in a list must begin with a column of ones if an intercept
                is to be included in the linear model. The design matrix passed in a list should
                not contain factors, as all columns are treated 'as is'. For tests that use ordinary
                least squares residuals, one can also pass a vector of residuals in the list, which
                should either be the third object or be named "e".

omit   A numeric vector of length p (the number of columns in the linear model design
                matrix) giving the indices of p observations to omit in the BLUS residual vector;
                or a character partially matching "first" (for the first p) observations, "last"
                (for the last p observations), or "random" (for a random sample of p indices
                between 1 and n). Defaults to "first".

keepNA   A logical. Should BLUS residuals for omitted observations be returned as
                NA_real_ to preserve the length of the residual vector?

exhaust   An integer. If singular matrices are encountered using the passed value of omit,
                how many random combinations of p indices should be attempted before an
                error is thrown? If NA (the default), all possible combinations are attempted
                provided that $\binom{n}{p} \leq 10^4$; otherwise up to 10^4 random samples of size p from
                1:n are attempted (with replacement). Integer values of exhaust greater than
                1e4L are treated as NA.

seed   An integer specifying a seed to pass to [set.seed](#) for random number generation.
                This allows reproducibility of bootstrap results. The value NA results in not
                setting a seed.

Details

Under the ideal linear model conditions, the BLUS residuals have a scalar covariance matrix ωI (meaning they have a constant variance and are mutually uncorrelated), unlike the OLS residuals, which have covariance matrix ωM where M is a function of the design matrix. Use of BLUS residuals could improve the performance of tests for heteroskedasticity and/or autocorrelation in the linear model. A linear model with n observations and an $n \times p$ design matrix yields only $n - p$ BLUS residuals. The choice of which p observations will not be represented in the BLUS residuals is specified within the algorithm.

Value

A double vector of length n containing the BLUS residuals (with `NA_real_`) for omitted observations), or a double vector of length $n - p$ containing the BLUS residuals only (if `keepNA` is set to `FALSE`)

References

Theil H (1965). "The Analysis of Disturbances in Regression Analysis." *Journal of the American Statistical Association*, **60**(312), 1067–1079.

Theil H (1968). "A Simplification of the BLUS Procedure for Analyzing Regression Disturbances." *Journal of the American Statistical Association*, **63**(321), 242–251.

See Also

H. D. Vinod's online article, Theil's BLUS Residuals and R Tools for Testing and Removing Autocorrelation and Heteroscedasticity, for an alternative function for computing BLUS residuals.

Examples

```
mtcars_lm <- lm(mpg ~ wt + qsec + am, data = mtcars)
blus(mtcars_lm)
plot(mtcars_lm$residuals, blus(mtcars_lm))
# Same as first example
mtcars_list <- list("y" = mtcars$mpg, "X" = cbind(1, mtcars$wt, mtcars$qsec, mtcars$am))
blus(mtcars_list)
# Again same as first example
mtcars_list2 <- list("e" = mtcars_lm$residuals, "X" = cbind(1, mtcars$wt, mtcars$qsec, mtcars$am))
blus(mtcars_list2)
# BLUS residuals cannot be computed with `omit = "last"` in this example, so
# omitted indices are randomised:
blus(mtcars_lm, omit = "last")
```

---

bootlm   *Nonparametric Bootstrapping of Heteroskedastic Linear Regression*
                                Models

Description

Generates B nonparametric bootstrap replications of a linear regression model that may have heteroskedasticity.

Usage

```
bootlm(
  object,
  sampmethod = c("pairs", "wild"),
  B = 1000L,
  resfunc = c("identity", "hccme"),
  fastfit = TRUE,
  ...
)
```

Arguments

object	Either an object of class "lm" (generated by lm), an object of class "alvm.fit" (generated by alvm.fit), or an object of class "anlvm.fit" (generated by anlvm.fit)
sampmethod	A character, either "pairs" or "wild", indicating the method to be used to generate the resampled models: bootstrapping pairs (Efron and Tibshirani 1993) or the wild bootstrap (Davidson and Flachaire 2008). Defaults to "pairs".
B	An integer representing the number of bootstrapped linear regression models to generate; defaults to 1000L
resfunc	Either a character naming a function to call to apply a transformation to the Ordinary Least Squares residuals, or a function to apply for the same purpose. This argument is ignored if sampmethod is "pairs". The only two character values accepted are "identity", in which case no transformation is applied to the residuals, and "hccme", in which case the transformation corresponds to a heteroskedasticity-consistent covariance matrix estimator calculated from hccme. If resfunc is a function, it is assumed that its first argument is the numeric vector of residuals.
fastfit	A logical indicating whether the lm.fit function should be used to fit the bootstrapped regression models for greater computational speed; defaults to TRUE. This affects the class of the returned value (see below). If FALSE, lm is used to fit the bootstrapped regression models.
...	Other arguments to pass to hccme

Details

Each replication of the pairs bootstrap entails drawing a sample of size n (the number of observations) with replacement from the indices $i = 1, 2, \ldots, n$. The pair or case (y_i, X_i) is included as an observation in the bootstrapped data set for each sampled index. An Ordinary Least Squares fit to the bootstrapped data set is then computed.

Under the wild bootstrap, each replication of the linear regression model is generated by first independently drawing n random values r_i, $i = 1, 2, \ldots, n$, from a distribution with zero mean and unit variance. The ith bootstrap response is then computed as

$$X_i'\hat{\beta} + f_i(e_i)r_i$$

, where X_i is the ith design observation, $\hat{\beta}$ is the Ordinary Least Squares estimate of the coefficient vector β, e_i is the ith Ordinary Least Squares residual, and $f_i(\cdot)$ is a function performing some transformation on the residual. An Ordinary Least Squares fit is then computed on the original design matrix and the bootstrap response vector.

Value

A list object of class `"bootlm"`, containing B objects, each of which is a bootstrapped linear regression model fit by Ordinary Least Squares. If `fastfit` was set to `TRUE`, each of these objects will be a list containing named objects y (the bootstrap response vector), X (the bootstrap design matrix, which is just the original design matrix under the wild bootstrap), e (the residual vector from the Ordinary Least Squares fit to this bootstrap data set), beta.hat (the vector of coefficient estimates from the Ordinary Least Squares fit to this bootstrap data set), sampmethod, and ind (a vector of the indices from the original data set used in this bootstrap sample; ignored under the wild bootstrap) of the kind returned by `lm.fit`; otherwise, each will be an object of class `"lm"`.

References

Davidson R, Flachaire E (2008). "The Wild Bootstrap, Tamed at Last." *Journal of Econometrics*, **146**, 162–169.

Efron B, Tibshirani RJ (1993). *An Introduction to the Bootstrap*. Springer Science+Business Media, Dordrecht.

See Also

`paired.boot` and `wild.boot` for the pairs bootstrap and wild bootstrap, respectively. The latter function does not appear to allow transformations of the residuals in the wild bootstrap.

Examples

```
mtcars_lm <- lm(mpg ~ wt + qsec + am, data = mtcars)
mybootlm <- bootlm(mtcars_lm)
```

---

breusch_pagan | *Breusch-Pagan Test for Heteroskedasticity in a Linear Regression Model*

Description

This function implements the popular method of Breusch and Pagan (1979) for testing for heteroskedasticity in a linear regression model, with or without the studentising modification of Koenker (1981).

Usage

```
breusch_pagan(mainlm, auxdesign = NA, koenker = TRUE, statonly = FALSE)
```

Arguments

mainlm | Either an object of [class](#) "lm" (e.g., generated by [lm](#)), or a list of two objects: a response vector and a design matrix. The objects are assumed to be in that order, unless they are given the names "X" and "y" to distinguish them. The design matrix passed in a list must begin with a column of ones if an intercept is to be included in the linear model. The design matrix passed in a list should not contain factors, as all columns are treated 'as is'. For tests that use ordinary least squares residuals, one can also pass a vector of residuals in the list, which should either be the third object or be named "e".

auxdesign | A [data.frame](#) or [matrix](#) representing an auxiliary design matrix of containing exogenous variables that (under alternative hypothesis) are related to error variance, or a character "fitted.values" indicating that the fitted $\hat{y}_i$ values from OLS should be used. If set to NA (the default), the design matrix of the original regression model is used. An intercept is included in the auxiliary regression even if the first column of auxdesign is not a vector of ones.

koenker | A logical. Should studentising modification of Koenker (1981) be implemented? Defaults to TRUE; if FALSE, the original form of the test proposed by Breusch and Pagan (1979) is used.

statonly | A logical. If TRUE, only the test statistic value is returned, instead of an object of [class](#) "htest". Defaults to FALSE.

Details

The Breusch-Pagan Test entails fitting an auxiliary regression model in which the response variable is the vector of squared residuals from the original model and the design matrix Z consists of one or more exogenous variables that are suspected of being related to the error variance. In the absence of prior information on a possible choice of Z, one would typically use the explanatory variables from the original model. Under the null hypothesis of homoskedasticity, the distribution of the test statistic is asymptotically chi-squared with parameter degrees of freedom. The test is right-tailed.

Value

An object of [class](#) "htest". If object is not assigned, its attributes are displayed in the console as a [tibble](#) using [tidy](#).

References

Breusch TS, Pagan AR (1979). "A Simple Test for Heteroscedasticity and Random Coefficient Variation." *Econometrica*, **47**(5), 1287–1294.

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

See Also

[lmtest::bptest](#), which performs exactly the same test as this function; [car::ncvTest](#), which is not the same test but is implemented in [cook_weisberg](#); [white](#), which is a special case of the Breusch-Pagan Test.

Examples

```
mtcars_lm <- lm(mpg ~ wt + qsec + am, data = mtcars)
breusch_pagan(mtcars_lm)
breusch_pagan(mtcars_lm, koenker = FALSE)
# Same as first example
mtcars_list <- list("y" = mtcars$mpg, "X" = cbind(1, mtcars$wt, mtcars$qsec, mtcars$am))
breusch_pagan(mtcars_list)
```

---

carapeto_holt	*Carapeto-Holt Test for Heteroskedasticity in a Linear Regression Model*

Description

This function implements the two methods (parametric and nonparametric) of Carapeto and Holt (2003) for testing for heteroskedasticity in a linear regression model.

Usage

```
carapeto_holt(
  mainlm,
  deflator = NA,
  prop_central = 1/3,
  group1prop = 1/2,
  qfmethod = "imhof",
  alternative = c("greater", "less", "two.sided"),
  twosidedmethod = c("doubled", "kulinskaya"),
  statonly = FALSE
)
```

Arguments

mainlm	Either an object of [class](#) "lm" (e.g., generated by [lm](#)), or a list of two objects: a response vector and a design matrix. The objects are assumed to be in that order, unless they are given the names "X" and "y" to distinguish them. The design matrix passed in a list must begin with a column of ones if an intercept is to be included in the linear model. The design matrix passed in a list should not contain factors, as all columns are treated 'as is'. For tests that use ordinary least squares residuals, one can also pass a vector of residuals in the list, which should either be the third object or be named "e".
deflator	Either a character specifying a column name from the design matrix of mainlm or an integer giving the index of a column of the design matrix. This variable is suspected to be related to the error variance under the alternative hypothesis. deflator may not correspond to a column of 1's (intercept). Default NA means the data will be left in its current order (e.g. in case the existing index is believed to be associated with error variance).
prop_central	A double specifying the proportion of central observations to exclude when comparing the two subsets of observations. [round](#) is used to ensure the number of central observations is an integer. Defaults to $\frac{1}{3}$.
group1prop	A double specifying the proportion of remaining observations *(after excluding central observations)* to allocate to the first group. The default value of 1 / 2 means that an equal number of observations is assigned to the first and second groups.
qfmethod	A character, either "imhof", "davies", or "integrate", corresponding to the algorithm argument of [pRQF](#). The default is "imhof".
alternative	A character specifying the form of alternative hypothesis. If it is suspected that the error variance is positively associated with the deflator variable, "greater". If it is suspected that the error variance is negatively associated with deflator variable, "less". If no information is available on the suspected direction of the association, "two.sided". Defaults to "greater".
twosidedmethod	A character indicating the method to be used to compute two-sided p-values for the parametric test when alternative is "two.sided". The argument is passed to [twosidedpval](#) as its method argument.
statonly	A logical. If TRUE, only the test statistic value is returned, instead of an object of [class](#) "htest". Defaults to FALSE.

Details

The test is based on the methodology of Goldfeld and Quandt (1965) but does not require any auxiliary regression. It entails ordering the observations by some suspected deflator (one of the explanatory variables) in such a way that, under the alternative hypothesis, the observations would now be arranged in decreasing order of error variance. A specified proportion of the most central observations (under this ordering) is removed, leaving a subset of lower observations and a subset of upper observations. The test statistic is then computed as a ratio of quadratic forms corresponding to the sums of squared residuals of the upper and lower observations respectively. p-values are computed by the Imhof algorithm in [pRQF](#).

Value

An object of [class](#) "htest". If object is not assigned, its attributes are displayed in the console as a [tibble](#) using [tidy](#).

References

Carapeto M, Holt W (2003). "Testing for Heteroscedasticity in Regression Models." *Journal of Applied Statistics*, **30**(1), 13–20.

Goldfeld SM, Quandt RE (1965). "Some Tests for Homoscedasticity." *Journal of the American Statistical Association*, **60**(310), 539–547.

Examples

```
mtcars_lm <- lm(mpg ~ wt + qsec + am, data = mtcars)
carapeto_holt(mtcars_lm, deflator = "qsec", prop_central = 0.25)
# Same as previous example
mtcars_list <- list("y" = mtcars$mpg, "X" = cbind(1, mtcars$wt, mtcars$qsec, mtcars$am))
carapeto_holt(mtcars_list, deflator = 3, prop_central = 0.25)
```

---

cook_weisberg *Cook-Weisberg Score Test for Heteroskedasticity in a Linear Regression Model*

Description

This function implements the score test of Cook and Weisberg (1983) for testing for heteroskedasticity in a linear regression model.

Usage

```
cook_weisberg(
  mainlm,
  auxdesign = NA,
  hetfun = c("mult", "add", "logmult"),
  statonly = FALSE
)
```

Arguments

mainlm   Either an object of [class](#) "lm" (e.g., generated by [lm](#)), or a list of two objects:
            a response vector and a design matrix. The objects are assumed to be in that
            order, unless they are given the names "X" and "y" to distinguish them. The
            design matrix passed in a list must begin with a column of ones if an intercept
            is to be included in the linear model. The design matrix passed in a list should
            not contain factors, as all columns are treated 'as is'. For tests that use ordinary
            least squares residuals, one can also pass a vector of residuals in the list, which
            should either be the third object or be named "e".

auxdesign   A [data.frame](#) or [matrix](#) representing an auxiliary design matrix of contain-
                ing exogenous variables that (under alternative hypothesis) are related to error
                variance, or a character "fitted.values" indicating that the fitted $\hat{y}_i$ values from
                OLS should be used. If set to NA (the default), the design matrix of the original
                regression model is used. An intercept is included in the auxiliary regression
                even if the first column of auxdesign is not a vector of ones.

hetfun   A character describing the form of $w(\cdot)$, the error variance function under the
                heteroskedastic alternative. Possible values are "mult" (the default), corre-
                sponding to $w(Z_i, \lambda) = \exp\left\{\sum_{j=1}^{q} \lambda_j Z_{ij}\right\}$, "add", corresponding to $w(Z_i, \lambda) = $
                $\left(1 + \sum_{j=1}^{q} \lambda_j Z_{ij}\right)^2$, and "logmult", corresponding to $w(Z_i, \lambda) = \exp\left\{\sum_{j=1}^{q} \lambda_j \log Z_{ij}\right\}$.
                The multiplicative and log-multiplicative cases are considered in Cook and Weis-
                berg (1983); the additive case is discussed, *inter alia*, by Griffiths and Surekha
                (1986). Results for the additive and multiplicative models are identical for this
                test. Partial matching is used.

statonly   A logical. If TRUE, only the test statistic value is returned, instead of an object
                of [class](#) "htest". Defaults to FALSE.

Details

The Cook-Weisberg Score Test entails fitting an auxiliary regression model in which the response
variable is the vector of standardised squared residuals $e_i^2/\hat{\omega}$ from the original OLS model and
the design matrix is some function of Z, an $n \times q$ matrix consisting of q exogenous variables,
appended to a column of ones. The test statistic is half the residual sum of squares from this
auxiliary regression. Under the null hypothesis of homoskedasticity, the distribution of the test
statistic is asymptotically chi-squared with q degrees of freedom. The test is right-tailed.

Value

An object of [class](#) "htest". If object is not assigned, its attributes are displayed in the console as
a [tibble](#) using [tidy](#).

References

Cook RD, Weisberg S (1983). "Diagnostics for Heteroscedasticity in Regression." *Biometrika*,
70(1), 1–10.

Griffiths WE, Surekha K (1986). "A Monte Carlo Evaluation of the Power of Some Tests for
Heteroscedasticity." *Journal of Econometrics*, **31**(1), 219–231.

See Also

[car::ncvTest](#), which implements the same test. Calling car::ncvTest with var.formula argu-
ment omitted is equivalent to calling skedastic::cook_weisberg with auxdesign = "fitted.values",
hetfun = "additive". Calling car::ncvTest with var.formula = ~ X (where X is the design ma-
trix of the linear model with the intercept column omitted) is equivalent to calling skedastic::cook_weisberg
with default auxdesign and hetfun values. The hetfun = "multiplicative" option has no equiv-
alent in car::ncvTest.

Examples

```
mtcars_lm <- lm(mpg ~ wt + qsec + am, data = mtcars)
cook_weisberg(mtcars_lm)
cook_weisberg(mtcars_lm, auxdesign = "fitted.values", hetfun = "logmult")
```

---

countpeaks   *Count peaks in a data sequence*

Description

This function computes the number of peaks in a double vector, with peak defined as per Goldfeld and Quandt (1965). The function is used in the Goldfeld-Quandt nonparametric test for heteroskedasticity in a linear model. `NA` and `NaN` values in the sequence are ignored.

Usage

```
countpeaks(x)
```

Arguments

x   A double vector.

Value

An integer value between `0` and `length(x) - 1` representing the number of peaks in the sequence.

References

Goldfeld SM, Quandt RE (1965). "Some Tests for Homoscedasticity." *Journal of the American Statistical Association*, **60**(310), 539–547.

See Also

goldfeld_quandt

Examples

```
set.seed(9586)
countpeaks(stats::rnorm(20))
mtcars_lm <- lm(mpg ~ wt + qsec + am, data = mtcars)
countpeaks(mtcars_lm$residuals)
```

---

dDtrend   *Probability mass function of nonparametric trend statistic D*

Description

This function computes $\Pr(D = k)$, i.e. the probability mass function for $D = \sum_{i=1}^{n}(R_i - i)^2$, the nonparametric trend statistic proposed by Lehmann (1975), under the assumption that the ranks R_i are computed on a series of n independent and identically distributed random variables with no ties.

Usage

```
dDtrend(k = "all", n, override = FALSE)
```

Arguments

k	An integer of length ≥ 1 or a character `"all"` (the default) indicating that the probability mass function should be applied to the entire support of D.
n	A positive integer representing the number of observations in the series. Note that computation time increases rapidly with n and is infeasible for $n > 11$.
override	A logical. By default, the function aborts if $n > 11$ due to the prohibitively slow computation (which may cause some systems to crash). Setting this argument to `TRUE` overrides the abort.

Details

The function is used within [horn](#) in computing p-values for Horn's nonparametric test for heteroskedasticity in a linear regression model (Horn 1981). The support of D consists of consecutive even numbers from 0 to $\frac{n(n-1)(n+1)}{3}$, with the exception of the case $n = 3$, when the value 4 is excluded from the support. Note that computation speed for `k = "all"` is about the same as when k is set to an individual integer value, because the entire distribution is still computed in the latter case.

Value

A double vector containing the probabilities corresponding to the integers in its `names` attribute.

References

Horn P (1981). "Heteroscedasticity of Residuals: A Non-Parametric Alternative to the Goldfeld-Quandt Peak Test." *Communications in Statistics - Theory and Methods*, **10**(8), 795–808.

Lehmann EL (1975). *Nonparametrics: Statistical Methods Based on Ranks*. Holden-Day, San Francisco.

See Also

[horn](#)

Examples

```
prob <- dDtrend(k = "all", n = 6)
values <- as.integer(names(prob))
plot(c(values[1], values[1]), c(0, prob[1]), type = "l",
  axes = FALSE, xlab = expression(k), ylab = expression(Pr(D == k)),
  xlim = c(0, 70), yaxs = "i", ylim = c(0, 1.05 * max(prob)))
  axis(side = 1, at = seq(0, 70, 10), las = 2)
for (i in seq_along(values)) {
 lines(c(values[i], values[i]), c(0, prob[i]))
}
```

---

diblasi_bowman	*Diblasi and Bowman's Test for Heteroskedasticity in a Linear Regression Model*

Description

This function implements the nonparametric test of Diblasi and Bowman (1997) for testing for heteroskedasticity in a linear regression model.

Usage

```
diblasi_bowman(
  mainlm,
  distmethod = c("moment.match", "bootstrap"),
  H = 0.08,
  ignorecov = TRUE,
  B = 500L,
  seed = 1234,
  statonly = FALSE
)
```

Arguments

mainlm   Either an object of [class](#) "lm" (e.g., generated by [lm](#)), or a list of two objects: a response vector and a design matrix. The objects are assumed to be in that order, unless they are given the names "X" and "y" to distinguish them. The design matrix passed in a list must begin with a column of ones if an intercept is to be included in the linear model. The design matrix passed in a list should not contain factors, as all columns are treated 'as is'. For tests that use ordinary least squares residuals, one can also pass a vector of residuals in the list, which should either be the third object or be named "e".

distmethod   A character specifying the method by which to estimate the *p*-values, either "moment.match" or "bootstrap".

H	A hyperparameter denoting the bandwidth matrix in the kernel function used for weights in nonparametric smoothing. If a double of length 1 (the default), H is set to $hI_{p'}$ where h is the scalar bandwidth value entered and $I_{p'}$ is the $p' \times p'$ identity matrix (where p' is the number of columns in the X matrix, excluding an intercept if present). If a double of length p', H is set to $diag(h)$ where h is the bandwidth vector entered. If H is a $p' \times p'$ matrix it is used as is. Any other dimensionality of H results in an error.
ignorecov	A logical. If TRUE (the default), the variance-covariance matrix of s is assumed to be diagonal. (This assumption is, strictly speaking, invalid, but usually yields a reasonable approximation. Computation time is prohibitive for large sample sizes if set to FALSE).
B	An integer specifying the number of nonparametric bootstrap replications to be used, if distmethod="bootstrap".
seed	An integer specifying a seed to pass to set.seed for random number generation. This allows reproducibility of bootstrap results. The value NA results in not setting a seed.
statonly	A logical. If TRUE, only the test statistic value is returned, instead of an object of class "htest". Defaults to FALSE.

Details

The test entails undertaking a transformation of the OLS residuals $s_i = \sqrt{|e_i|} - E_0(\sqrt{|e_i|})$, where E_0 denotes expectation under the null hypothesis of homoskedasticity. The kernel method of nonparametric regression is used to fit the relationship between these s_i and the explanatory variables. This leads to a test statistic T that is a ratio of quadratic forms involving the vector of s_i and the matrix of normal kernel weights. Although nonparametric in its method of fitting the possible heteroskedastic relationship, the distributional approximation used to compute p-values assumes normality of the errors.

Value

An object of class "htest". If object is not assigned, its attributes are displayed in the console as a tibble using tidy.

References

Diblasi A, Bowman A (1997). "Testing for Constant Variance in a Linear Model." *Statistics & Probability Letters*, **33**, 95–103.

Examples

```
mtcars_lm <- lm(mpg ~ wt + qsec + am, data = mtcars)
diblasi_bowman(mtcars_lm)
diblasi_bowman(mtcars_lm, ignorecov = FALSE)
diblasi_bowman(mtcars_lm, distmethod = "bootstrap")
```

---

dpeak | *Probability mass function of number of peaks in an i.i.d. random sequence*

Description

This function computes $P(n, k)$ as defined by Goldfeld and Quandt (1965), i.e. the probability that a sequence of n independent and identically distributed random variables contains exactly k peaks, with peaks also as defined by Goldfeld and Quandt (1965). The function is used in ppeak to compute p-values for the Goldfeld-Quandt nonparametric test for heteroskedasticity in a linear model.

Usage

```
dpeak(k, n, usedata = FALSE)
```

Arguments

k   An integer or a sequence of integers strictly incrementing by 1, with all values between 0 and n - 1 inclusive. Represents the number of peaks in the sequence.

n   A positive integer representing the number of observations in the sequence.

usedata A logical. Should probability mass function values be read from dpeakdat rather than computing them? This option will save significantly on computation time if $n < 170$ but is currently only available for $n \leq 1000$.

Value

A double between 0 and 1 representing the probability of exactly k peaks occurring in a sequence of n independent and identically distributed continuous random variables. The double has a names attribute with the values corresponding to the probabilities. Computation time is very slow for $n > 170$ (if usedata is FALSE) and for $n > 1000$ regardless of usedata value.

References

Goldfeld SM, Quandt RE (1965). "Some Tests for Homoscedasticity." *Journal of the American Statistical Association*, **60**(310), 539–547.

See Also

ppeak, goldfeld_quandt

Examples

```
dpeak(0:9, 10)
plot(0:9, dpeak(0:9, 10), type = "p", pch = 20, xlab = "Number of Peaks",
         ylab = "Probability")

# This would be extremely slow if usedata were set to FALSE:
```

```
prob <- dpeak(0:199, 200, usedata = TRUE)
plot(0:199, prob, type = "l", xlab = "Number of Peaks", ylab = "Probability")

# `dpeakdat` is a dataset containing probabilities generated from `dpeak`
utils::data(dpeakdat)
expval <- unlist(lapply(dpeakdat,
                 function(p) sum(p * 0:(length(p) - 1))))
plot(1:1000, expval[1:1000], type = "l", xlab = parse(text = "n"),
     ylab = "Expected Number of Peaks")
```

---

dpeakdat   *Probability distribution for number of peaks in a continuous, uncorre-*
                         lated stochastic series

Description

A dataset containing the probability mass function for the distribution of the number of peaks in a
continuous, uncorrelated stochastic series.

Usage

```
dpeakdat
```

Format

A list of 1000 objects. The nth object is a double vector of length n, with elements representing the
probability of k peaks for $k = 0, 1, \ldots, n - 1$.

Details

These probabilities were generated from the dpeak function. This function is computationally very
slow for $n > 170$; thus the functions of skedastic package that require peak probabilities (ppeak
and goldfeld_quandt) by default obtain the probabilities from this data set rather than from dpeak,
provided that $n \leq 1000$. For $n > 1000$, good luck!

---

dufour_etal   *Dufour et al.'s Monte Carlo Test for Heteroskedasticity in a Linear*
                         Regression Model

Description

This function implements the method of Dufour et al. (2004) for testing for heteroskedasticity in a
linear regression model.

Usage

```
dufour_etal(
  mainlm,
  hettest,
  R = 1000L,
  alternative = c("greater", "less", "two.sided"),
  errorgen = stats::rnorm,
  errorparam = list(),
  seed = 1234,
  ...
)
```

Arguments

mainlm   Either an object of class "lm" (e.g., generated by [lm](lm)), or a list of two objects:
                a response vector and a design matrix. The objects are assumed to be in that
                order, unless they are given the names "X" and "y" to distinguish them. The
                design matrix passed in a list must begin with a column of ones if an intercept
                is to be included in the linear model. The design matrix passed in a list should
                not contain factors, as all columns are treated 'as is'. For tests that use ordinary
                least squares residuals, one can also pass a vector of residuals in the list, which
                should either be the third object or be named "e".

hettest   A character specifying the name of a function that implements a heteroskedas-
                ticity test on a linear regression model. The function is called with the statonly
                argument set to TRUE to improve computational efficiency.

R   An integer specifying the number of Monte Carlo replicates to generate. De-
                faults to 1000.

alternative   The tailedness of the test whose statistic is computed by hettest function; one
                of "greater" (the default), "less", or "two.sided".

errorgen   A function, or a character specifying the name of a function, from which the
                random errors are to be generated. The function should correspond to a continu-
                ous probability distribution that has (or at least can have) a mean of 0. Defaults
                to rnorm.

errorparam   An optional list of parameters to pass to errorgen. This argument is ignored if
                errorgen is rnorm, since mean must be set to 0, and sd is set to 1 because the
                heteroskedasticity test implemented by hettest function is assumed to be scale
                invariant. If errorgen is not rnorm, errorparam should be chosen in such a
                way that the error distribution has a mean of 0.

seed   An integer specifying a seed to pass to set.seed for random number generation.
                This allows reproducibility of Monte Carlo results. A value of NA results in not
                setting a seed.

...   Additional arguments to pass to function with name hettest

Details

The test implements a Monte Carlo procedure as follows. (1) The observed value of the test statistic
T_0 is computed using function with name hettest. (2) R replications of the random error vector

are generated from the distribution specified using errorgen. (3) R replications of the test statistic, $T_1, T_2, \ldots, T_R$, are computed from the generated error vectors. (4) The empirical p-value is computed as $\frac{\hat{G}_R(T_0)+1}{R+1}$, where $\hat{G}_R(x) = \sum_{j=1}^{R} 1_{T_j \geq x}$, $1_\bullet$ being the indicator function. The test is right-tailed, regardless of the tailedness of hettest. Note that the heteroskedasticity test implemented by hettest must have a test statistic that is continuous and that is invariant with respect to nuisance parameters (ω and β). Note further that if hettest is goldfeld_quandt with method argument "parametric", the replicated Goldfeld-Quandt F statistics are computed directly within this function rather than by calling goldfeld_quandt, due to some idiosyncratic features of this test. Note that, if alternative is set to "two.sided", the one-sided p-value is doubled (twosidedpval cannot be used in this case).

Value

An object of class "htest". If object is not assigned, its attributes are displayed in the console as a tibble using tidy.

References

Dufour J, Khalaf L, Bernard J, Genest I (2004). "Simulation-Based Finite-Sample Tests for Heteroskedasticity and ARCH Effects." *Journal of Econometrics*, **122**, 317–347.

Examples

```
mtcars_lm <- lm(mpg ~ wt + qsec + am, data = mtcars)
dufour_etal(mtcars_lm, hettest = "breusch_pagan")
```

---

evans_king   *Evans-King Tests for Heteroskedasticity in a Linear Regression Model*

Description

This function implements the two methods of Evans and King (1988) for testing for heteroskedasticity in a linear regression model.

Usage

```
evans_king(
  mainlm,
  method = c("GLS", "LM"),
  deflator = NA,
  lambda_star = 5,
  qfmethod = "imhof",
  statonly = FALSE
)
```

Arguments

mainlm	Either an object of [class](#) ″lm″ (e.g., generated by [lm](#)), or a list of two objects: a response vector and a design matrix. The objects are assumed to be in that order, unless they are given the names ″X″ and ″y″ to distinguish them. The design matrix passed in a list must begin with a column of ones if an intercept is to be included in the linear model. The design matrix passed in a list should not contain factors, as all columns are treated 'as is'. For tests that use ordinary least squares residuals, one can also pass a vector of residuals in the list, which should either be the third object or be named ″e″.
method	A character indicating which of the two tests derived in Evans and King (1988) should be implemented. Possible values are ″GLS″ and ″LM″; partial matching is used (which is not case-sensitive).
deflator	Either a character specifying a column name from the design matrix of mainlm or an integer giving the index of a column of the design matrix. This variable is suspected to be related to the error variance under the alternative hypothesis. deflator may not correspond to a column of 1's (intercept). Default NA means the data will be left in its current order (e.g. in case the existing index is believed to be associated with error variance).
lambda_star	A double; coefficient representing the degree of heteroskedasticity under the alternative hypothesis. Evans and King (1985) suggests 2.5, 5, 7.5, and 10 as values to consider, and Evans and King (1988) finds that 2.5 and 5 perform best empirically. This parameter is used only for the ″GLS″ method; the ″LM″ method represents the limiting case as $\lambda^\star \to 0$. Defaults to 5.
qfmethod	A character, either ″imhof″, ″davies″, or ″integrate″, corresponding to the algorithm argument of [pRQF](#). The default is ″imhof″.
statonly	A logical. If TRUE, only the test statistic value is returned, instead of an object of [class](#) ″htest″. Defaults to FALSE.

Details

The test entails putting the data rows in increasing order of some specified deflator (e.g., one of the explanatory variables) that is believed to be related to the error variance by some non-decreasing function. There are two statistics that can be used, corresponding to the two values of the method argument. In both cases the test statistic can be expressed as a ratio of quadratic forms in the errors, and thus the Imhof algorithm is used to compute p-values. Both methods involve a left-tailed test.

Value

An object of [class](#) ″htest″. If object is not assigned, its attributes are displayed in the console as a [tibble](#) using [tidy](#).

References

Evans M, King ML (1985). "A Point Optimal Test for Heteroscedastic Disturbances." *Journal of Econometrics*, **27**(2), 163–178.

Evans MA, King ML (1988). "A Further Class of Tests for Heteroscedasticity." *Journal of Econometrics*, **37**, 265–276.

See Also

Evans and King (1985), which already anticipates one of the tests.

Examples

```
mtcars_lm <- lm(mpg ~ wt + qsec + am, data = mtcars)
evans_king(mtcars_lm, deflator = "qsec", method = "GLS")
evans_king(mtcars_lm, deflator = "qsec", method = "LM")
```

---

glejser   *Glejser Test for Heteroskedasticity in a Linear Regression Model*

Description

This function implements the method of Glejser (1969) for testing for "multiplicative" heteroskedasticity in a linear regression model. Mittelhammer et al. (2000) gives the formulation of the test used here.

Usage

```
glejser(
  mainlm,
  auxdesign = NA,
  sigmaest = c("main", "auxiliary"),
  statonly = FALSE
)
```

Arguments

mainlm   Either an object of class "lm" (e.g., generated by [lm](lm)), or a list of two objects: a response vector and a design matrix. The objects are assumed to be in that order, unless they are given the names "X" and "y" to distinguish them. The design matrix passed in a list must begin with a column of ones if an intercept is to be included in the linear model. The design matrix passed in a list should not contain factors, as all columns are treated 'as is'. For tests that use ordinary least squares residuals, one can also pass a vector of residuals in the list, which should either be the third object or be named "e".

auxdesign   A data.frame or matrix representing an auxiliary design matrix of containing exogenous variables that (under alternative hypothesis) are related to error variance, or a character "fitted.values" indicating that the fitted $\hat{y}_i$ values from OLS should be used. If set to NA (the default), the design matrix of the original regression model is used. An intercept is included in the auxiliary regression even if the first column of auxdesign is not a vector of ones.

sigmaest   A character indicating which model residuals to use in the $\hat{\omega}$ estimator in the denominator of the test statistic. If `"main"` (the default), the OLS residuals from the original model are used; this produces results identical to the Glejser Test in SHAZAM software. If `"auxiliary"`, the OLS residuals from the auxiliary model are used, as in Mittelhammer et al. (2000). Partial matching is used.

statonly   A logical. If `TRUE`, only the test statistic value is returned, instead of an object of [class](#) `"htest"`. Defaults to `FALSE`.

Details

Glejser's Test entails fitting an auxiliary regression model in which the response variable is the absolute residual from the original model and the design matrix Z consists of one or more exogenous variables that are suspected of being related to the error variance. In the absence of prior information on a possible choice of Z, one would typically use the explanatory variables from the original model. Under the null hypothesis of homoskedasticity, the distribution of the test statistic is asymptotically chi-squared with `parameter` degrees of freedom. The test is right-tailed.

Value

An object of [class](#) `"htest"`. If object is not assigned, its attributes are displayed in the console as a [tibble](#) using [tidy](#).

References

Glejser H (1969). "A New Test for Heteroskedasticity." *Journal of the American Statistical Association*, **64**(325), 316–323.

Mittelhammer RC, Judge GG, Miller DJ (2000). *Econometric Foundations*. Cambridge University Press, Cambridge.

See Also

the description of the test in SHAZAM software (which produces identical results).

Examples

```
mtcars_lm <- lm(mpg ~ wt + qsec + am, data = mtcars)
glejser(mtcars_lm)
```

---

godfrey_orme   *Godfrey and Orme's Nonparametric Bootstrap Test for Heteroskedasticity in a Linear Regression Model*

Description

This function implements the method of Godfrey and Orme (1999) for testing for heteroskedasticity in a linear regression model. The procedure is more clearly described in Godfrey et al. (2006).

Usage

```
godfrey_orme(
  mainlm,
  hettest,
  B = 1000L,
  alternative = c("greater", "less", "two.sided"),
  seed = 1234,
  ...
)
```

Arguments

mainlm   Either an object of [class](#) "lm" (e.g., generated by [lm](#)), or a list of two objects: a response vector and a design matrix. The objects are assumed to be in that order, unless they are given the names "X" and "y" to distinguish them. The design matrix passed in a list must begin with a column of ones if an intercept is to be included in the linear model. The design matrix passed in a list should not contain factors, as all columns are treated 'as is'. For tests that use ordinary least squares residuals, one can also pass a vector of residuals in the list, which should either be the third object or be named "e".

hettest   A character specifying the name of a function that implements a heteroskedasticity test on a linear regression model. The function is called with the `statonly` argument set to `TRUE` to improve computational efficiency.

B   An integer specifying the number of nonparametric bootstrap samples to generate. Defaults to `1000L`.

alternative   The tailedness of the test whose statistic is computed by `hettest` function; one of "greater" (the default), "less", or "two.sided".

seed   An integer specifying a seed to pass to [set.seed](#) for random number generation. This allows reproducibility of bootstrap results. A value of `NA` results in not setting a seed.

...   Additional arguments to pass to function with name `hettest`

Details

The procedure runs as follows. (1) The observed value of the test statistic T_0 is computed using function with name `hettest`. (2) A sample $e_1^\star, e_2^\star, \ldots, e_n^\star$ is drawn with replacement from the OLS residuals. (3) Bootstrapped response values are computed as $y_i^\star = x_i'\hat{\beta} + e_i^\star, i = 1, 2, \ldots, n$. (4) Bootstrapped test statistic value $T^\star$ is computed from the regression of $y^\star$ on X using function `hettest`. (5) Steps (2)-(4) are repeated until B bootstrapped test statistic values are computed. (6) Empirical p-value is computed by comparing the bootstrapped test statistic values to the observed test statistic value. Note that, if `alternative` is set to "two.sided", the one-sided p-value is doubled ([twosidedpval](#) cannot be used in this case).

Value

An object of [class](#) "htest". If object is not assigned, its attributes are displayed in the console as a [tibble](#) using [tidy](#).

References

Godfrey LG, Orme CD (1999). "The Robustness, Reliability and Power of Heteroskedasticity Tests." *Econometric Reviews*, **18**(2), 169–194.

Godfrey LG, Orme CD, Silva JS (2006). "Simulation-Based Tests for Heteroskedasticity in Linear Regression Models: Some Further Results." *Econometrics Journal*, **9**, 76–97.

Examples

```
mtcars_lm <- lm(mpg ~ wt + qsec + am, data = mtcars)
godfrey_orme(mtcars_lm, hettest = "breusch_pagan")
```

---

goldfeld_quandt   *Goldfeld-Quandt Tests for Heteroskedasticity in a Linear Regression Model*

Description

This function implements the two methods (parametric and nonparametric) of Goldfeld and Quandt (1965) for testing for heteroskedasticity in a linear regression model.

Usage

```
goldfeld_quandt(
  mainlm,
  method = c("parametric", "nonparametric"),
  deflator = NA,
  prop_central = 1/3,
  group1prop = 1/2,
  alternative = c("greater", "less", "two.sided"),
  prob = NA,
  twosidedmethod = c("doubled", "kulinskaya"),
  restype = c("ols", "blus"),
  statonly = FALSE,
  ...
)
```

Arguments

mainlm   Either an object of class "lm" (e.g., generated by [lm](lm)), or a list of two objects: a response vector and a design matrix. The objects are assumed to be in that order, unless they are given the names "X" and "y" to distinguish them. The design matrix passed in a list must begin with a column of ones if an intercept is to be included in the linear model. The design matrix passed in a list should not contain factors, as all columns are treated 'as is'. For tests that use ordinary least squares residuals, one can also pass a vector of residuals in the list, which should either be the third object or be named "e".

method   A character indicating which of the two tests derived in Goldfeld and Quandt
                (1965) should be implemented. Possible values are "parametric" and "nonpara-
                metric". Default is "parametric". It is acceptable to specify only the first letter.

deflator   Either a character specifying a column name from the design matrix of mainlm
                or an integer giving the index of a column of the design matrix. This variable
                is suspected to be related to the error variance under the alternative hypothesis.
                deflator may not correspond to a column of 1's (intercept). Default NA means
                the data will be left in its current order (e.g. in case the existing index is believed
                to be associated with error variance).

prop_central   A double specifying the proportion of central observations to exclude from the F
                test (when method is "parametric" only). round is used to ensure the number
                of central observations is an integer. The value must be small enough to allow
                the two auxiliary regressions to be fit; otherwise an error is thrown. Defaults to
                1 / 3.

group1prop   A double specifying the proportion of remaining observations *(after excluding
                central observations)* to allocate to the first group. The default value of 1 / 2
                means that an equal number of observations is assigned to the first and second
                groups.

alternative   A character specifying the form of alternative hypothesis. If it is suspected that
                the error variance is positively associated with the deflator variable, "greater".
                If it is suspected that the error variance is negatively associated with deflator
                variable, "less". If no information is available on the suspected direction of the
                association, "two.sided". Defaults to "greater".

prob   A vector of probabilities corresponding to values of the test statistic (number of
                peaks) from 0 to $n-1$ inclusive (used only when method is "nonparametric").
                If NA (the default), probabilities are calculated within the function by calling
                ppeak. The user can improve computational performance of the test (for in-
                stance, when the test is being used repeatedly in a simulation) by pre-specifying
                the exact probability distribution of the number of peaks using this argument,
                e.g. by calling the nth element of dpeakdat (or $(n - p)$th element, if BLUS
                residuals are used).

twosidedmethod A character indicating the method to be used to compute two-sided p-values for
                the parametric test when alternative is "two.sided". The argument is passed
                to twosidedpval as its method argument.

restype   A character specifying which residuals to use: "ols" for OLS residuals (the de-
                fault) or the "blus" for BLUS residuals. The advantage of using BLUS resid-
                uals is that, under the null hypothesis, the assumption that the random series is
                independent and identically distributed is met (whereas with OLS residuals it is
                not). The disadvantage of using BLUS residuals is that only $n - p$ residuals are
                used rather than the full n. This argument is ignored if method is "parametric".

statonly   A logical. If TRUE, only the test statistic value is returned, instead of an object
                of class "htest". Defaults to FALSE.

...   Optional further arguments to pass to blus.

Details

The parametric test entails putting the data rows in increasing order of some specified deflator (one of the explanatory variables). A specified proportion of the most central observations (under this ordering) is removed, leaving a subset of lower observations and a subset of upper observations. Separate OLS regressions are fit to these two subsets of observations (using all variables from the original model). The test statistic is the ratio of the sum of squared residuals from the 'upper' model to the sum of squared residuals from the 'lower' model. Under the null hypothesis, the test statistic is exactly F-distributed with numerator and denominator degrees of freedom equal to $(n-c)/2 - p$ where n is the number of observations in the original regression model, c is the number of central observations removed, and p is the number of columns in the design matrix (number of parameters to be estimated, including intercept).

The nonparametric test entails putting the residuals of the linear model in increasing order of some specified deflator (one of the explanatory variables). The test statistic is the number of peaks, with the jth absolute residual $|e_j|$ defined as a peak if $|e_j| \geq |e_i|$ for all $i < j$. The first observation does not constitute a peak. If the number of peaks is large relative to the distribution of peaks under the null hypothesis, this constitutes evidence for heteroskedasticity.

Value

An object of [class](#) "htest". If object is not assigned, its attributes are displayed in the console as a [tibble](#) using [tidy](#).

References

Goldfeld SM, Quandt RE (1965). "Some Tests for Homoscedasticity." *Journal of the American Statistical Association*, **60**(310), 539–547.

See Also

[lmtest::gqtest](#), another implementation of the Goldfeld-Quandt Test (parametric method only).

Examples

```
mtcars_lm <- lm(mpg ~ wt + qsec + am, data = mtcars)
goldfeld_quandt(mtcars_lm, deflator = "qsec", prop_central = 0.25)
# This is equivalent to lmtest::gqtest(mtcars_lm, fraction = 0.25, order.by = mtcars$qsec)
goldfeld_quandt(mtcars_lm, deflator = "qsec", method = "nonparametric",
 restype = "blus")
goldfeld_quandt(mtcars_lm, deflator = "qsec", prop_central = 0.25, alternative = "two.sided")
goldfeld_quandt(mtcars_lm, deflator = "qsec", method = "nonparametric",
 restype = "blus", alternative = "two.sided")
```

GSS	*Golden Section Search for Minimising Univariate Function over a Closed Interval*

Description

Golden Section Search (GSS) is a useful algorithm for minimising a continuous univariate function $f(x)$ over an interval $[a, b]$ in instances where the first derivative $f'(x)$ is not easily obtainable, such as with loss functions that need to be minimised under cross-validation to tune a hyperparameter in a machine learning model. The method is described by Fox (2021).

Usage

```
GSS(f, a, b, tol = 1e-08, maxitgss = 100L, ...)
```

Arguments

f	A function of one variable that returns a numeric value
a	A numeric of length 1 representing the lower endpoint of the search interval
b	A numeric of length 1 representing the upper endpoint of the search interval; must be greater than a
tol	A numeric of length 1 representing the tolerance used to determine when the search interval has been narrowed sufficiently for convergence
maxitgss	An integer of length 1 representing the maximum number of iterations to use in the search
...	Additional arguments to pass to f

Details

This function is modelled after a MATLAB function written by (). The method assumes that there is one local minimum in this interval. The solution produced is also an interval, the width of which can be made arbitrarily small by setting a tolerance value. Since the desired solution is a single point, the midpoint of the final interval can be taken as the best approximation to the solution.

The premise of the method is to shorten the interval $[a, b]$ by comparing the values of the function at two test points, $x_1 < x_2$, where $x_1 = a + r(b - a)$ and $x_2 = b - r(b - a)$, and $r = (\sqrt{5} - 1)/2 \approx 0.618$ is the reciprocal of the golden ratio. One compares $f(x_1)$ to $f(x_2)$ and updates the search interval $[a, b]$ as follows:

- If $f(x_1) < f(x_2)$, the solution cannot lie in $[x_2, b]$; thus, update the search interval to

$$[a_{\text{new}}, b_{\text{new}}] = [a, x_2]$$

- If $f(x_1) > f(x_2)$, the solution cannot lie in $[a, x_1]$; thus, update the search interval to

$$[a_{\text{new}}, b_{\text{new}}] = [x_1, b]$$

One then chooses two new test points by replacing a and b with a_{new} and b_{new} and recalculating x_1 and x_2 based on these new endpoints. One continues iterating in this fashion until $b - a < \tau$, where τ is the desired tolerance.

Value

A list object containing the following:

- argmin, the argument of f that minimises f
- funmin, the minimum value of f achieved at argmin
- converged, a logical indicating whether the convergence tolerance was satisfied
- iterations, an integer indicating the number of search iterations used

References

(????). *Golden Section Method Algorithm, author=Katarzyna Zarnowiec, organization=MATLAB Central File Exchange, url=https://www.mathworks.com/matlabcentral/fileexchange/25919-golden-section-method-algorithm, urldate=2022-10-19, year=2022,.*

Fox WP (2021). *Nonlinear Optimization: Models and Applications*, 1st edition. Chapman and Hall/CRC, Boca Raton, FL.

See Also

goldsect is similar to this function, but does not allow the user to pass additional arguments to f

Examples

```
f <- function(x) (x - 1) ^ 2
GSS(f, a = 0, b = 2)
```

---

harrison_mccabe	*Harrison and McCabe's Test for Heteroskedasticity in a Linear Regression Model*

Description

This function implements the method of Harrison and McCabe (1979) for testing for heteroskedasticity in a linear regression model.

Usage

```
harrison_mccabe(
  mainlm,
  deflator = NA,
  m = 0.5,
  alternative = c("less", "greater", "two.sided"),
  twosidedmethod = c("doubled", "kulinskaya"),
  qfmethod = "imhof",
  statonly = FALSE
)
```

Arguments

mainlm	Either an object of [class](#) "lm" (e.g., generated by [lm](#)), or a list of two objects: a response vector and a design matrix. The objects are assumed to be in that order, unless they are given the names "X" and "y" to distinguish them. The design matrix passed in a list must begin with a column of ones if an intercept is to be included in the linear model. The design matrix passed in a list should not contain factors, as all columns are treated 'as is'. For tests that use ordinary least squares residuals, one can also pass a vector of residuals in the list, which should either be the third object or be named "e".
deflator	Either a character specifying a column name from the design matrix of mainlm or an integer giving the index of a column of the design matrix. This variable is suspected to be related to the error variance under the alternative hypothesis. deflator may not correspond to a column of 1's (intercept). Default NA means the data will be left in its current order (e.g. in case the existing index is believed to be associated with error variance).
m	Either a double giving the proportion of the n diagonal elements of A that are ones, or an integer giving the index m up to which the diagonal elements are ones. Defaults to 0.5.
alternative	A character specifying the form of alternative hypothesis. If it is suspected that the error variance is positively associated with the deflator variable, "less". If it is suspected that the error variance is negatively associated with deflator variable, "greater". If no information is available on the suspected direction of the association, "two.sided". Defaults to "less".
twosidedmethod	A character indicating the method to be used to compute two-sided p-values for the parametric test when alternative is "two.sided". The argument is passed to [twosidedpval](#) as its method argument.
qfmethod	A character, either "imhof", "davies", or "integrate", corresponding to the algorithm argument of [pRQF](#). The default is "imhof".
statonly	A logical. If TRUE, only the test statistic value is returned, instead of an object of [class](#) "htest". Defaults to FALSE.

Details

The test assumes that heteroskedasticity, if present, is monotonically related to one of the explanatory variables (known as the deflator). The OLS residuals e are placed in increasing order of the deflator variable and we let A be an $n \times n$ selector matrix whose first m diagonal elements are 1 and all other elements are 0. The alternative hypothesis posits that the error variance changes around index m. Under the null hypothesis of homoskedasticity, the ratio of quadratic forms $Q = \frac{e'Ae}{e'e}$ should be close to $\frac{m}{n}$. Since the test statistic Q is a ratio of quadratic forms in the errors, the Imhof algorithm is used to compute p-values (with normality of errors assumed).

Value

An object of [class](#) "htest". If object is not assigned, its attributes are displayed in the console as a [tibble](#) using [tidy](#).

References

Harrison MJ, McCabe BPM (1979). "A Test for Heteroscedasticity Based on Ordinary Least Squares Residuals." *Journal of the American Statistical Association*, **74**(366), 494–499.

See Also

[lmtest::hmctest](), another implementation of the Harrison-McCabe Test. Note that the p-values from that function are simulated rather than computed from the distribution of a ratio of quadratic forms in normal random vectors.

Examples

```
mtcars_lm <- lm(mpg ~ wt + qsec + am, data = mtcars)
harrison_mccabe(mtcars_lm, deflator = "qsec")
```

---

harvey   *Harvey Test for Heteroskedasticity in a Linear Regression Model*

Description

This function implements the method of Harvey (1976) for testing for "multiplicative" heteroskedasticity in a linear regression model. Mittelhammer et al. (2000) gives the formulation of the test used here.

Usage

```
harvey(mainlm, auxdesign = NA, statonly = FALSE)
```

Arguments

mainlm   Either an object of [class]() "lm" (e.g., generated by [lm]()), or a list of two objects: a response vector and a design matrix. The objects are assumed to be in that order, unless they are given the names "X" and "y" to distinguish them. The design matrix passed in a list must begin with a column of ones if an intercept is to be included in the linear model. The design matrix passed in a list should not contain factors, as all columns are treated 'as is'. For tests that use ordinary least squares residuals, one can also pass a vector of residuals in the list, which should either be the third object or be named "e".

auxdesign   A [data.frame]() or [matrix]() representing an auxiliary design matrix of containing exogenous variables that (under alternative hypothesis) are related to error variance, or a character "fitted.values" indicating that the fitted $\hat{y}_i$ values from OLS should be used. If set to NA (the default), the design matrix of the original regression model is used. An intercept is included in the auxiliary regression even if the first column of auxdesign is not a vector of ones.

statonly   A logical. If TRUE, only the test statistic value is returned, instead of an object of [class]() "htest". Defaults to FALSE.

Details

Harvey's Test entails fitting an auxiliary regression model in which the response variable is the log of the vector of squared residuals from the original model and the design matrix Z consists of one or more exogenous variables that are suspected of being related to the error variance. In the absence of prior information on a possible choice of Z, one would typically use the explanatory variables from the original model. Under the null hypothesis of homoskedasticity, the distribution of the test statistic is asymptotically chi-squared with parameter degrees of freedom. The test is right-tailed.

Value

An object of class "htest". If object is not assigned, its attributes are displayed in the console as a tibble using tidy.

References

Harvey AC (1976). "Estimating Regression Models with Multiplicative Heteroscedasticity." *Econometrica*, **44**(3), 461–465.

Mittelhammer RC, Judge GG, Miller DJ (2000). *Econometric Foundations*. Cambridge University Press, Cambridge.

See Also

the description of the test in SHAZAM software (which produces identical results).

Examples

```
mtcars_lm <- lm(mpg ~ wt + qsec + am, data = mtcars)
harvey(mtcars_lm)
harvey(mtcars_lm, auxdesign = "fitted.values")
```

---

hccme   *Heteroskedasticity-Consistent Covariance Matrix Estimators for Linear Regression Models*

Description

Computes an estimate of the $n \times n$ covariance matrix Ω (assumed to be diagonal) of the random error vector of a linear regression model, using a specified method

Usage

```
hccme(
  mainlm,
  hcnum = c("3", "0", "1", "2", "4", "5", "6", "7", "4m", "5m", "const"),
  sandwich = FALSE,
  as_matrix = TRUE
)
```

Arguments

mainlm   Either an object of [class]("lm" (e.g., generated by [lm]), or a list of two objects: a response vector and a design matrix. The objects are assumed to be in that order, unless they are given the names "X" and "y" to distinguish them. The design matrix passed in a list must begin with a column of ones if an intercept is to be included in the linear model. The design matrix passed in a list should not contain factors, as all columns are treated 'as is'. For tests that use ordinary least squares residuals, one can also pass a vector of residuals in the list, which should either be the third object or be named "e".

hcnum   A character corresponding to a subscript in the name of an HCCME according to the usual nomenclature HC#. Possible values are:

- "3", the default, corresponding to HC3 (MacKinnon and White 1985)
- "0", corresponding to HC0 (White 1980)
- "1", corresponding to HC1 (MacKinnon and White 1985)
- "2", corresponding to HC1 (MacKinnon and White 1985)
- "4", corresponding to HC4 (Cribari-Neto 2004)
- "5", corresponding to HC5 (Cribari-Neto et al. 2007)
- "6", corresponding to HC6 (Aftab and Chand 2016)
- "7", corresponding to HC7 (Aftab and Chand 2018)
- "4m", corresponding to HC4m (Cribari-Neto and da Silva 2011)
- "5m", corresponding to HC5m (Li et al. 2017)
- "const", corresponding to the homoskedastic estimator, $(n-p)^{-1}\sum_{i=1}^{n} e_i^2$

sandwich   A logical, defaulting to FALSE, indicating whether or not the sandwich estimator

$$\mathrm{Cov}\hat{\beta} = (X'X)^{-1}X'\hat{\Omega}X(X'X)^{-1}$$

should be returned instead of $\mathrm{Cov}(\epsilon) = \hat{\Omega}$

as_matrix   A logical, defaulting to TRUE, indicating whether a covariance matrix estimate should be returned rather than a vector of variance estimates

Value

A numeric matrix (if as_matrix is TRUE) or else a numeric vector

References

Aftab N, Chand S (2016). "A New Heteroskedastic Consistent Covariance Matrix Estimator Using Deviance Measure." *Pakistan Journal of Statistics and Operations Research*, **12**(2), 235–244.

Aftab N, Chand S (2018). "A Simulation-Based Evidence on the Improved Performance of a New Modified Leverage Adjusted Heteroskedastic Consistent Covariance Matrix Estimator in the Linear Regression Model." *Kuwait Journal of Science*, **45**(3), 29–38.

Cribari-Neto F (2004). "Asymptotic Inference under Heteroskedasticity of Unknown Form." *Computational Statistics & Data Analysis*, **45**, 215–233.

Cribari-Neto F, Souza TC, Vasconcellos KLP (2007). "Inference under Heteroskedasticity and Leveraged Data." *Communications in Statistics - Theory and Methods*, **36**(10), 1877–1888.

Cribari-Neto F, da Silva WB (2011). "A New Heteroskedasticity-Consistent Covariance Matrix Estimator for the Linear Regression Model." *Advances in Statistical Analysis*, **95**(2), 129–146.

Li S, Zhang N, Zhang X, Wang G (2017). "A New Heteroskedasticity-Consistent Covariance Matrix Estimator and Inference under Heteroskedasticity." *Journal of Statistical Computation and Simulation*, **87**(1), 198–210.

MacKinnon JG, White H (1985). "Some Heteroskedasticity-Consistent Covariance Matrix Estimators with Improved Finite Sample Properties." *Journal of Econometrics*, **29**(3), 305–325.

White H (1980). "A Heteroskedasticity-Consistent Covariance Matrix Estimator and a Direct Test for Heteroskedasticity." *Econometrica*, **48**(4), 817–838.

See Also

vcovHC

Examples

```
mtcars_lm <- lm(mpg ~ wt + qsec + am, data = mtcars)
Omega_hat <- hccme(mtcars_lm, hcnum = "4")
Cov_beta_hat <- hccme(mtcars_lm, hcnum = "4", sandwich = TRUE)
```

---

hetplot	*Graphical Methods for Detecting Heteroskedasticity in a Linear Regression Model*

Description

This function creates various two-dimensional scatter plots that can aid in detecting heteroskedasticity in a linear regression model.

Usage

```
hetplot(
  mainlm,
  horzvar = "index",
  vertvar = "res",
  vertfun = "identity",
  filetype = NA,
  values = FALSE,
  ...
)
```

Arguments

mainlm	Either an object of [class](#) ″lm″ (e.g., generated by [lm](#)), or a list of two objects: a response vector and a design matrix. The objects are assumed to be in that order, unless they are given the names ″X″ and ″y″ to distinguish them. The design matrix passed in a list must begin with a column of ones if an intercept is to be included in the linear model. The design matrix passed in a list should not contain factors, as all columns are treated 'as is'. For tests that use ordinary least squares residuals, one can also pass a vector of residuals in the list, which should either be the third object or be named ″e″.
horzvar	A character vector describing the variable(s) to plot on horizontal axes (″index″ for the data index i, ″fitted.values″ for the OLS predicted values $\hat{y}_i$, ″fitted.values2″ for transformed OLS predicted values $m_{ii}\hat{y}_i$, and/or names of explanatory variable columns). ″explanatory″ passes all explanatory variable columns. ″log_″ concatenated with names of explanatory variable columns passes logs of those explanatory variables. ″log_explanatory″ passes logs of all explanatory variables. If more than one variable is specified, a separate plot is created for each.
vertvar	A character vector describing the variable to plot on the vertical axis (″res″ for OLS residuals [the default], ″res_blus″ for [BLUS](#) residuals, ″res_stand″ for standardised OLS residuals: $e_i/\sqrt{\hat{\omega}}$, ″res_constvar″ for OLS residuals transformed to have constant variance: $e_i/\sqrt{m_{ii}}$, ″res_stud″ for studentised OLS residuals: $e_i/(s\sqrt{m_{ii}})$. If more than one value is specified, a separate plot is created for each.
vertfun	A character vector giving the name of a function to apply to the vertvar variable. Values that can be coerced to numeric such as ″2″ are taken to be powers to which vertvar should be set. If multiple values are specified, they are all applied to each element of vertvar. Common choices might be ″identity″ (the default), ″abs″ (absolute value), and ″2″ (squaring).
filetype	A character giving the type of image file to which the plot(s) should be written. Values can be ″png″, ″bmp″, ″jpeg″, or ″tiff″. Image files are written to a subdirectory called "hetplot" within the R session's temporary directory, which can be located using tempdir(). The files should be moved or copied to another location if they are needed after the R session is ended. Default filenames contain timestamps for uniqueness. If NA (the default), no image files are written, and in this case, if there are multiple plots, they are plotted on a single device using the ″mfrow″ graphical parameter. If many plots are requested at once, it is advisable to write them to image files.
values	A logical. Should the sequences corresponding to the horizontal and vertical variable(s) be returned in a list object?
...	Arguments to be passed to methods, such as graphical parameters (see [par](#)), parameters for [plot](#), for [graphics](#) [devices](#), and/or the omit argument for function [blus](#), if BLUS residuals are being plotted. If it is desired to pass the type argument to a graphics device, use gtype = , since a type argument will be passed to [plot](#).

Details

The variable plotted on the horizontal axis could be the original data indices, one of the explanatory variables, the OLS predicted (fitted) values, or any other numeric vector specified by the user. The variable plotted on the vertical axis is some function of the OLS residuals or transformed version thereof such as the BLUS residuals Theil (1968) or standardised or studentised residuals as discussed in Cook and Weisberg (1983). A separate plot is created for each (`horzvar`, `vertvar`, `vertfun`) combination.

Value

A list containing two [data frames](#), one for vectors plotted on horizontal axes and one for vectors plotted on vertical axes.

References

Cook RD, Weisberg S (1983). "Diagnostics for Heteroscedasticity in Regression." *Biometrika*, **70**(1), 1–10.

Theil H (1968). "A Simplification of the BLUS Procedure for Analyzing Regression Disturbances." *Journal of the American Statistical Association*, **63**(321), 242–251.

See Also

[plot.lm](#)

Examples

```
mtcars_lm <- lm(mpg ~ wt + qsec + am, data = mtcars)
# Generates 2 x 2 matrix of plots in console
hetplot(mtcars_lm, horzvar = c("index", "fitted.values"),
vertvar = c("res_blus"), vertfun = c("2", "abs"), filetype = NA)

# Generates 84 png files in tempdir() folder
## Not run: hetplot(mainlm = mtcars_lm, horzvar = c("explanatory", "log_explanatory",
"fitted.values2"), vertvar = c("res", "res_stand", "res_stud",
"res_constvar"), vertfun = c("identity", "abs", "2"), filetype = "png")
## End(Not run)
```

---

honda | *Honda's Test for Heteroskedasticity in a Linear Regression Model*

Description

This function implements the two-sided LM Test of Honda (1989) for testing for heteroskedasticity in a linear regression model.

Usage

```
honda(
  mainlm,
  deflator = NA,
  alternative = c("two.sided", "greater", "less"),
  twosidedmethod = c("doubled", "kulinskaya"),
  qfmethod = "imhof",
  statonly = FALSE
)
```

Arguments

mainlm
: Either an object of [class](#) "lm" (e.g., generated by [lm](#)), or a list of two objects: a response vector and a design matrix. The objects are assumed to be in that order, unless they are given the names "X" and "y" to distinguish them. The design matrix passed in a list must begin with a column of ones if an intercept is to be included in the linear model. The design matrix passed in a list should not contain factors, as all columns are treated 'as is'. For tests that use ordinary least squares residuals, one can also pass a vector of residuals in the list, which should either be the third object or be named "e".

deflator
: Either a character specifying a column name from the design matrix of mainlm or an integer giving the index of a column of the design matrix. This variable is suspected to be related to the error variance under the alternative hypothesis. deflator may not correspond to a column of 1's (intercept). Default NA means the data will be left in its current order (e.g. in case the existing index is believed to be associated with error variance).

alternative
: A character specifying the form of alternative hypothesis. If it is suspected that the error variance is positively associated with the deflator variable, "greater". If it is suspected that the error variance is negatively associated with deflator variable, "less". If no information is available on the suspected direction of the association, "two.sided". Defaults to "greater".

twosidedmethod
: A character indicating the method to be used to compute two-sided p-values for the parametric test when alternative is "two.sided". The argument is passed to [twosidedpval](#) as its method argument.

qfmethod
: A character, either "imhof", "davies", or "integrate", corresponding to the algorithm argument of [pRQF](#). The default is "imhof".

statonly
: A logical. If TRUE, only the test statistic value is returned, instead of an object of [class](#) "htest". Defaults to FALSE.

Details

The test assumes that heteroskedasticity, if present, would be either of the form $\omega_i = \omega(1 + \theta z_i)$ or of the form $\omega_i = \omega e^{\theta z_i}$, where where z_i is a deflator (a nonstochastic variable suspected of being related to the error variance), ω is some unknown constant, and θ is an unknown parameter representing the degree of heteroskedasticity. Since the test statistic $Q = \frac{e'A_0e}{e'e}$ is a ratio of quadratic forms in the errors, the Imhof algorithm is used to compute p-values. Since the null distribution

is in general asymmetrical, the two-sided p-value is computed using the conditional method of
Kulinskaya (2008), setting $A = 1$.

Value

An object of [class]{.blue} "htest". If object is not assigned, its attributes are displayed in the console as
a [tibble]{.blue} using [tidy]{.blue}.

References

Honda Y (1989). "On the Optimality of Some Tests of the Error Covariance Matrix in the Linear
Regression Model." *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*,
51(1), 71–79.

Kulinskaya E (2008). "On Two-Sided p-Values for Non-Symmetric Distributions." 0810.2124,
0810.2124.

Examples

```
mtcars_lm <- lm(mpg ~ wt + qsec + am, data = mtcars)
# In this example, only the test statistic is returned, to save time
honda(mtcars_lm, deflator = "qsec", statonly = TRUE)
```

---

horn   *Horn's Test for Heteroskedasticity in a Linear Regression Model*

Description

This function implements the nonparametric test of Horn (1981) for testing for heteroskedasticity
in a linear regression model.

Usage

```
horn(
  mainlm,
  deflator = NA,
  restype = c("ols", "blus"),
  alternative = c("two.sided", "greater", "less"),
  exact = (nres <= 10),
  statonly = FALSE,
  ...
)
```

Arguments

mainlm	Either an object of [class](#) ″lm″ (e.g., generated by [lm](#)), or a list of two objects: a response vector and a design matrix. The objects are assumed to be in that order, unless they are given the names ″X″ and ″y″ to distinguish them. The design matrix passed in a list must begin with a column of ones if an intercept is to be included in the linear model. The design matrix passed in a list should not contain factors, as all columns are treated 'as is'. For tests that use ordinary least squares residuals, one can also pass a vector of residuals in the list, which should either be the third object or be named ″e″.
deflator	Either a character specifying a column name from the design matrix of mainlm or an integer giving the index of a column of the design matrix. This variable is suspected to be related to the error variance under the alternative hypothesis. deflator may not correspond to a column of 1's (intercept). Default NA means the data will be left in its current order (e.g. in case the existing index is believed to be associated with error variance).
restype	A character specifying which residuals to use: ″ols″ for OLS residuals (the default) or the ″blus″ for [BLUS](#) residuals. The advantage of using BLUS residuals is that, under the null hypothesis, the assumption that the random series is independent and identically distributed is met (whereas with OLS residuals it is not). The disadvantage of using BLUS residuals is that only $n - p$ residuals are used rather than the full n.
alternative	A character specifying the form of alternative hypothesis; one of ″two.sided″ (default), ″greater″, or ″less″. ″two.sided″ corresponds to any trend in the absolute residuals when ordered by deflator. ″greater″ corresponds to a negative trend in the absolute residuals when ordered by deflator. ″less″ corresponds to a positive trend in the absolute residuals when ordered by deflator. (Notice that D tends to be small when there is a positive trend.)
exact	A logical. Should exact p-values be computed? If FALSE, a normal approximation is used. Defaults to TRUE only if the number of absolute residuals being ranked is ≤ 10.
statonly	A logical. If TRUE, only the test statistic value is returned, instead of an object of [class](#) ″htest″. Defaults to FALSE.
...	Optional further arguments to pass to [blus](#).

Details

The test entails specifying a 'deflator', an explanatory variable suspected of being related to the error variance. Residuals are ordered by the deflator and the nonparametric trend statistic $D = \sum (R_i - i)^2$ proposed by Lehmann (1975) is then computed on the absolute residuals and used to test for an increasing or decreasing trend, either of which would correspond to heteroskedasticity. Exact probabilities for the null distribution of D can be obtained from functions [dDtrend](#) and [pDtrend](#), but since computation time increases rapidly with n, use of a normal approximation is recommended for $n > 10$. Lehmann (1975) proves that D is asymptotically normally distributed and the approximation of the statistic $Z = (D - E(D))/\sqrt{V(D)}$ to the standard normal distribution is already quite good for $n = 11$.

The expectation and variance of D (when ties are absent) are respectively $E(D) = \frac{n^3-n}{6}$ and $V(D) = \frac{n^2(n+1)^2(n-1)}{36}$; see Lehmann (1975) for $E(D)$ and $V(D)$ when ties are present. When ties are absent, a continuity correction is used to improve the normal approximation. When exact distribution is used, two-sided p-value is computed by doubling the one-sided p-value, since the distribution of D is symmetric. The function does not support the exact distribution of D in the presence of ties, so in this case the normal approximation is used regardless of n.

Value

An object of `class` "htest". If object is not assigned, its attributes are displayed in the console as a `tibble` using `tidy`.

References

Horn P (1981). "Heteroscedasticity of Residuals: A Non-Parametric Alternative to the Goldfeld-Quandt Peak Test." *Communications in Statistics - Theory and Methods*, **10**(8), 795–808.

Lehmann EL (1975). *Nonparametrics: Statistical Methods Based on Ranks*. Holden-Day, San Francisco.

Examples

```
mtcars_lm <- lm(mpg ~ wt + qsec + am, data = mtcars)
horn(mtcars_lm, deflator = "qsec")
horn(mtcars_lm, deflator = "qsec", restype = "blus")
```

---

li_yao   *Li-Yao ALRT and CVT Tests for Heteroskedasticity in a Linear Regression Model*

Description

This function implements the two methods of Li and Yao (2019) for testing for heteroskedasticity in a linear regression model.

Usage

```
li_yao(mainlm, method = c("cvt", "alrt"), baipanyin = TRUE, statonly = FALSE)
```

Arguments

mainlm   Either an object of `class` "lm" (e.g., generated by `lm`), or a list of two objects: a response vector and a design matrix. The objects are assumed to be in that order, unless they are given the names "X" and "y" to distinguish them. The design matrix passed in a list must begin with a column of ones if an intercept is to be included in the linear model. The design matrix passed in a list should not contain factors, as all columns are treated 'as is'. For tests that use ordinary

least squares residuals, one can also pass a vector of residuals in the list, which should either be the third object or be named "e".

method   A character indicating which of the two tests derived in Li and Yao (2019) should be implemented. Possible values are "alrt" (approximate likelihood-ratio test) and "cvt" (coefficient-of-variation test). Default is "cvt". Partial matching is used.

baipanyin   A logical. Should the central limit theorem of Bai et al. (2016) be used to determine the p-value for the coefficient-of-variation test (assuming normally distributed errors)? If FALSE, the asymptotic null distribution in Li and Yao (2019) is used, which requires the assumption that the design variables are random and normally distributed. This argument is ignored if method is "alrt".

statonly   A logical. If TRUE, only the test statistic value is returned, instead of an object of [class](#) "htest". Defaults to FALSE.

Details

These two tests are straightforward to implement; in both cases the test statistic is a function only of the residuals of the linear regression model. Furthermore, in both cases the test statistic is asymptotically normally distributed under the null hypothesis of homoskedasticity. Both tests are right-tailed. These tests are designed to be especially powerful in high-dimensional regressions, i.e. when the number of explanatory variables is large.

Value

An object of [class](#) "htest". If object is not assigned, its attributes are displayed in the console as a [tibble](#) using [tidy](#).

References

Bai Z, Pan G, Yin Y (2016). "Homoscedasticity Tests for Both Low and High-Dimensional Fixed Design Regressions." 1603.03830, 1603.03830.

Li Z, Yao J (2019). "Testing for Heteroscedasticity in High-Dimensional Regressions." *Econometrics and Statistics*, **9**, 122–139.

Examples

```
mtcars_lm <- lm(mpg ~ wt + qsec + am, data = mtcars)
li_yao(mtcars_lm, method = "alrt")
li_yao(mtcars_lm, method = "cvt")
li_yao(mtcars_lm, method = "cvt", baipanyin = FALSE)
# Same as first example
li_yao(list("e" = mtcars_lm$residuals), method = "alrt")
```

Description

This function computes $\Pr(D \leq k)$ ($\Pr(D \geq k)$), i.e. lower (upper) cumulative probabilities for $D = \sum_{i=1}^{n}(R_i - i)^2$, the nonparametric trend statistic proposed by Lehmann (1975), under the assumption that the ranks R_i are computed on a series of n independent and identically distributed random variables with no ties. The function may be used to compute one-sided p-values for the nonparametric test for heteroskedasticity of Horn (1981). Computation time is extremely slow for $n > 10$ if usedata is set to FALSE; thus a normal approximation is implemented, including a continuity correction.

Usage

```
pDtrend(
  k,
  n,
  lower.tail = TRUE,
  exact = (n <= 10),
  tiefreq = NA,
  override = FALSE
)
```

Arguments

k	An integer of length ≥ 1 or a character "all" indicating that the cumulative distribution function should be applied to the entire support of D. The latter is only acceptable when exact is TRUE, since the distribution is otherwise continuous.
n	A positive integer representing the number of observations in the series.
lower.tail	A logical. Should lower tailed cumulative probability be calculated? Defaults to TRUE. Note that both lower- and upper- tailed cumulative probabilities are computed inclusive of k.
exact	A logical. Should exact distribution of D be used by calling [dDtrend](#)? If FALSE, a normal approximation is used. If tiefreq is not NA (ties are present), normal approximation is used regardless of the value of exact. By default, exact is set to TRUE provided that n <= 10. Setting exact to TRUE for n > 11 results in an error unless override is set to TRUE.
tiefreq	A double vector corresponding to the value of d_i in Lehmann (1975). These are the frequencies of the various tied ranks. If ties are absent, NA (the default).
override	A logical. By default, the dDtrend function aborts if $n > 11$ due to the prohibitively slow computation (which may cause some systems to crash). Setting this argument to TRUE overrides the abort. Ignored unless exact is TRUE.

Value

A double between 0 and 1 representing the probability/ies of D taking on at least (at most) the value(s) in the names attribute.

References

Horn P (1981). "Heteroscedasticity of Residuals: A Non-Parametric Alternative to the Goldfeld-Quandt Peak Test." *Communications in Statistics - Theory and Methods*, **10**(8), 795–808.

Lehmann EL (1975). *Nonparametrics: Statistical Methods Based on Ranks*. Holden-Day, San Francisco.

See Also

dDtrend, horn

Examples

```
# For an independent sample of size 6, the probability that D is <= 50 is
# 0.8222
pDtrend(k = 50, n = 6)
# Normal approximation of the above with continuity correction is
# 0.8145
pDtrend(k = 50, n = 6, exact = FALSE)
```

---

ppeak	*Cumulative distribution function of number of peaks in an i.i.d. random sequence*

Description

This function computes $\sum_k P(n, k)$, i.e. the probability that a sequence of n independent and identically distributed random variables contains $\geq k$ ($\leq k$) peaks, with peaks as defined in Goldfeld and Quandt (1965). The function may be used to compute p-values for the Goldfeld-Quandt nonparametric test for heteroskedasticity in a linear model. Computation time is very slow for $n > 170$ if usedata is set to FALSE.

Usage

```
ppeak(k, n, lower.tail = FALSE, usedata = TRUE)
```

Arguments

k	An integer or a sequence of integers strictly incrementing by 1, with all values between 0 and n - 1 inclusive. Represents the number of peaks in the sequence.
n	A positive integer representing the number of observations in the sequence.

lower.tail	A logical. Should lower tailed cumulative probability be calculated? Defaults to FALSE due to function being designed primarily for calculating p-values for the peaks test, which is by default an upper-tailed test. Note that both upper and lower tailed cumulative probabilities are computed inclusive of k.
usedata	A logical. Should probability mass function values be read from dpeakdat rather than computing them from dpeak? This option will save significantly on computation time if $n < 170$ but is currently only available for $n \leq 1000$.

Value

A double between 0 and 1 representing the probability of at least (at most) k peaks occurring in a sequence of n independent and identically distributed continuous random variables. The double has a names attribute with the values corresponding to the probabilities.

References

Goldfeld SM, Quandt RE (1965). "Some Tests for Homoscedasticity." *Journal of the American Statistical Association*, **60**(310), 539–547.

See Also

dpeak, goldfeld_quandt

Examples

```
# For an independent sample of size 250, the probability of at least 10
# peaks is 0.02650008
ppeak(k = 10, n = 250, lower.tail = FALSE, usedata = TRUE)
# For an independent sample of size 10, the probability of at most 2 peaks
# is 0.7060615
ppeak(k = 2, n = 10, lower.tail = TRUE, usedata = FALSE)
```

---

pRQF   *Probabilities for a Ratio of Quadratic Forms in a Normal Random Vector*

Description

This function computes cumulative probabilities (lower or upper tail) on a ratio of quadratic forms in a vector of normally distributed random variables.

Usage

```
pRQF(
  r,
  A,
  B,
```

```
    Sigma = diag(nrow(A)),
    algorithm = c("imhof", "davies", "integrate"),
    lower.tail = TRUE,
    usenames = FALSE
)
```

Arguments

r	A double representing the value(s) for which $\Pr(R \le r)$ or $\Pr(R \ge r)$ should be computed.
A	A numeric, symmetric matrix that is symmetric
B	A numeric, symmetric, non-negative definite matrix having the same dimensions as A.
Sigma	A numeric, symmetric matrix with the same dimensions as A and B, denoting the covariance matrix of the normal random vector. Defaults to the identity matrix, corresponding to the case in which the normal random variables are independent and identically distributed.
algorithm	A character, either "imhof", "davies", or "integrate". Values "imhof" and "integrate" both implement the Imhof algorithm. The difference is that "imhof" means that imhof is used, whereas "integrate" means that integrate is used (which is slower). The Imhof algorithm is more precise than the Davies algorithm.
lower.tail	A logical. If TRUE, the cumulative distribution function $\Pr(R \le r)$ is computed; if FALSE, the survival function $\Pr(R \ge r)$ is computed.
usenames	A logical. If TRUE, the function value has a names attribute corresponding to r.

Details

Most of the work is done by other functions, namely imhof, davies, or integrate (depending on the algorithm argument). It is assumed that the ratio of quadratic forms can be expressed as

$$R = \frac{x'Ax}{x'Bx}$$

where x is an n-dimensional normally distributed random variable with mean vector μ and co-variance matrix Σ, and A and B are real-valued, symmetric $n \times n$ matrices. Matrix B must be non-negative definite to ensure that the denominator of the ratio of quadratic forms is nonzero.

The function makes use of the fact that a probability statement involving a ratio of quadratic forms can be rewritten as a probability statement involving a quadratic form. Hence, methods for computing probabilities for a quadratic form in normal random variables, such as the Imhof algorithm (Imhof 1961) or the Davies algorithm (Davies 1980) can be applied to the rearranged expression to obtain the probability for the ratio of quadratic forms. Note that the Ruben-Farebrother algorithm (as implemented in farebrother) cannot be used here because the A matrix within the quadratic form (after rearrangement of the probability statement involving a ratio of quadratic forms) is not in general positive semi-definite.

Value

A double denoting the probability/ies corresponding to the value(s) r.

References

Davies RB (1980). "Algorithm AS 155: The Distribution of a Linear Combination of χ^2 Random Variables." *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, **29**, 323–333.

Imhof JP (1961). "Computing the Distribution of Quadratic Forms in Normal Variables." *Biometrika*, **48**(3/4), 419–426.

See Also

Duchesne and de Micheaux (2010), the article associated with the `imhof` and `davies` functions.

Examples

```
n <- 20
A <- matrix(data = 1, nrow = n, ncol = n)
B <- diag(n)
pRQF(r = 1, A = A, B = B)
pRQF(r = 1, A = A, B = B, algorithm = "integrate")
pRQF(r = 1:3, A = A, B = B, algorithm = "davies")
```

---

rackauskas_zuokas	*Rackauskas-Zuokas Test for Heteroskedasticity in a Linear Regression Model*

Description

This function implements the two methods of Rackauskas and Zuokas (2007) for testing for heteroskedasticity in a linear regression model.

Usage

```
rackauskas_zuokas(
  mainlm,
  alpha = 0,
  pvalmethod = c("data", "sim"),
  R = 2^14,
  m = 2^17,
  sqZ = FALSE,
  seed = 1234,
  statonly = FALSE
)
```

Arguments

mainlm	Either an object of [class](#) ″lm″ (e.g., generated by [lm](#)), or a list of two objects: a response vector and a design matrix. The objects are assumed to be in that order, unless they are given the names ″X″ and ″y″ to distinguish them. The design matrix passed in a list must begin with a column of ones if an intercept is to be included in the linear model. The design matrix passed in a list should not contain factors, as all columns are treated 'as is'. For tests that use ordinary least squares residuals, one can also pass a vector of residuals in the list, which should either be the third object or be named ″e″.
alpha	A double such that $0 \leq \alpha < 1/2$; a hyperparameter of the test. Defaults to 0.
pvalmethod	A character, either ″data″ or ″sim″, determining which method to use to compute the empirical p-value. If ″data″, the dataset [T_alpha](#) consisting of pre-generated Monte Carlo replicates from the asymptotic null distribution of the test statistic is loaded and used to compute empirical p-value. This is only available for certain values of alpha, namely $i/32$ where $i = 0, 1, \ldots, 15$. If ″sim″, Monte Carlo replicates are generated from the asymptotic null distribution. Partial matching is used.
R	An integer representing the number of Monte Carlo replicates to generate, if pvalmethod == ″sim″. Ignored if pvalmethod == ″data″.
m	An integer representing the number of standard normal variates to use when generating the Brownian Bridge for each replicate, if pvalmethod == ″sim″. Ignored if pvalmethod == ″data″. If number of observations is small, Rackauskas and Zuokas (2007) recommends using $m = n$. The dataset [T_alpha](#) used $m = 2^17$ which is computationally intensive.
sqZ	A logical. If TRUE, the standard normal variates used in the Brownian Bridge when generating from the asymptotic null distribution are first squared, i.e. transformed to $\chi^2(1)$ variates. This is recommended by Rackauskas and Zuokas (2007) when the number of observations is small. Ignored if pvalmethod == ″data″.
seed	An integer representing the seed to be used for pseudorandom number generation when simulating values from the asymptotic null distribution. This is to provide reproducibility of test results. Ignored if pvalmethod == ″data″. If user does not wish to set the seed, pass NA.
statonly	A logical. If TRUE, only the test statistic value is returned, instead of an object of [class](#) ″htest″. Defaults to FALSE.

Details

Rackauskas and Zuokas propose a class of tests that entails determining the largest weighted difference in variance of estimated error. The asymptotic behaviour of their test statistic $T_{n,\alpha}$ is studied using the empirical polygonal process constructed from partial sums of the squared residuals. The test is right-tailed.

Value

An object of [class](#) ″htest″. If object is not assigned, its attributes are displayed in the console as a [tibble](#) using [tidy](#).

References

Rackauskas A, Zuokas D (2007). "New Tests of Heteroskedasticity in Linear Regression Model." *Lithuanian Mathematical Journal*, **47**(3), 248–265.

Examples

```
mtcars_lm <- lm(mpg ~ wt + qsec + am, data = mtcars)
rackauskas_zuokas(mtcars_lm)
rackauskas_zuokas(mtcars_lm, alpha = 7 / 16)

n <- length(mtcars_lm$residuals)
rackauskas_zuokas(mtcars_lm, pvalmethod = "sim", m = n, sqZ = TRUE)
```

---

simonoff_tsai	*Simonoff-Tsai Tests for Heteroskedasticity in a Linear Regression Model*

Description

This function implements the modified profile likelihood ratio test and score test of Simonoff and Tsai (1994) for testing for heteroskedasticity in a linear regression model.

Usage

```
simonoff_tsai(
  mainlm,
  auxdesign = NA,
  method = c("mlr", "score"),
  hetfun = c("mult", "add", "logmult"),
  basetest = c("koenker", "cook_weisberg"),
  bartlett = TRUE,
  optmethod = "Nelder-Mead",
  statonly = FALSE,
  ...
)
```

Arguments

mainlm   Either an object of [class](#) "lm" (e.g., generated by [lm](#)), or a list of two objects:
                a response vector and a design matrix. The objects are assumed to be in that
                order, unless they are given the names "X" and "y" to distinguish them. The
                design matrix passed in a list must begin with a column of ones if an intercept
                is to be included in the linear model. The design matrix passed in a list should
                not contain factors, as all columns are treated 'as is'. For tests that use ordinary
                least squares residuals, one can also pass a vector of residuals in the list, which
                should either be the third object or be named "e".

auxdesign	A [data.frame](#) or [matrix](#) representing an auxiliary design matrix of containing exogenous variables that (under alternative hypothesis) are related to error variance, or a character "fitted.values" indicating that the fitted $\hat{y}_i$ values from OLS should be used. If set to NA (the default), the design matrix of the original regression model is used. An intercept is included in the auxiliary regression even if the first column of auxdesign is not a vector of ones.
method	A character specifying which of the tests proposed in Simonoff and Tsai (1994) to implement. "mlr" corresponds to the modified profile likelihood ratio test, and "score" corresponds to the score test.
hetfun	A character describing the form of $w(\cdot)$, the error variance function under the heteroskedastic alternative. Possible values are "mult" (the default), corresponding to $w(Z_i, \lambda) = \exp\left\{\sum_{j=1}^{q} \lambda_j Z_{ij}\right\}$, "add", corresponding to $w(Z_i, \lambda) = \left(1 + \sum_{j=1}^{q} \lambda_j Z_{ij}\right)^2$, and "logmult", corresponding to $w(Z_i, \lambda) = \exp\left\{\sum_{j=1}^{q} \lambda_j \log Z_{ij}\right\}$. The multiplicative and log-multiplicative cases are considered in Cook and Weisberg (1983); the additive case is discussed, *inter alia*, by Griffiths and Surekha (1986). Results for the additive and multiplicative models are identical for this test. Partial matching is used.
basetest	A character specifying the base test statistic which is robustified using the added term described in Details. "koenker" corresponds to the test statistic produced by [breusch_pagan](#) with argument koenker set to TRUE, while "cook_weisberg" corresponds to the test statistic produced by [cook_weisberg](#). Partial matching is used. This argument is only used if method is "score".
bartlett	A logical specifying whether a Bartlett correction should be made, as per Ferrari et al. (2004), to improve the fit of the test statistic to the asymptotic null distribution. This argument is only applicable where method is "mlr", and is implemented only where hetfun is "mult" or "logmult".
optmethod	A character specifying the optimisation method to use with [optim](#), if method is "mlr". The default, "Nelder-Mead", corresponds to the default method value in [optim](#). Warnings about Nelder-Mead algorithm being unreliable for one-dimensional optimization have been suppressed, since the algorithm does appear to work for the three implemented choices of hetfun.
statonly	A logical. If TRUE, only the test statistic value is returned, instead of an object of [class](#) "htest". Defaults to FALSE.
...	Optional arguments to pass to [optim](#), such as par (initial value of λ) and maxit (maximum number of iterations to use in optimisation algorithm), and trace (to provide detailed output on optimisation algorithm). Default initial value of λ is rep(1e-3, q).

Details

The Simonoff-Tsai Likelihood Ratio Test involves a modification of the profile likelihood function so that the nuisance parameter will be orthogonal to the parameter of interest. The maximum likelihood estimate of λ (called δ in Simonoff and Tsai (1994)) is computed from the modified profile log-likelihood function using the Nelder-Mead algorithm in [optim](#). Under the null hypothesis of homoskedasticity, the distribution of the test statistic is asymptotically chi-squared with q degrees of freedom. The test is right-tailed.

The Simonoff-Tsai Score Test entails adding a term to either the score statistic of Cook and Weisberg (1983) (a test implemented in `cook_weisberg`) or to that of Koenker (1981) (a test implemented in `breusch_pagan` with argument koenker set to TRUE), in order to improve the robustness of these respective tests in the presence of non-normality. This test likewise has a test statistic that is asymptotically $\chi^2(q)$-distributed and the test is likewise right-tailed.

The assumption of underlying both tests is that $\mathrm{Cov}(\epsilon) = \omega W$, where W is an $n \times n$ diagonal matrix with ith diagonal element $w_i = w(Z_i, \lambda)$. Here, Z_i is the ith row of an $n \times q$ nonstochastic auxiliary design matrix Z. Note: Z as defined here does not have a column of ones, but is concatenated to a column of ones when used in an auxiliary regression. λ is a q-vector of unknown parameters, and $w(\cdot)$ is a real-valued, twice-differentiable function having the property that there exists some λ_0 for which $w(Z_i, \lambda_0) = 0$ for all $i = 1, 2, \ldots, n$. Thus, the null hypothesis of homoskedasticity may be expressed as $\lambda = \lambda_0$.

In the score test, the added term in the test statistic is of the form

$$\sum_{j=1}^{q} \left(\sum_{i=1}^{n} h_{ii} t_{ij} \right) \tau_j$$

, where t_{ij} is the (i,j)th element of the Jacobian matrix J evaluated at $\lambda = \lambda_0$:

$$t_{ij} = \left. \frac{\partial w(Z_i, \lambda)}{\partial \lambda_j} \right|_{\lambda = \lambda_0}$$

, and $\tau = (\bar{J}'\bar{J})^{-1}\bar{J}'d$, where d is the n-vector whose ith element is $e_i^2 \bar{\omega}^{-1}$, $\bar{\omega} = n^{-1}e'e$, and $\bar{J} = (I_n - 1_n 1_n'/n)J$.

Value

An object of `class` "htest". If object is not assigned, its attributes are displayed in the console as a `tibble` using `tidy`.

References

Cook RD, Weisberg S (1983). "Diagnostics for Heteroscedasticity in Regression." *Biometrika*, **70**(1), 1–10.

Ferrari SL, Cysneiros AH, Cribari-Neto F (2004). "An Improved Test for Heteroskedasticity Using Adjusted Modified Profile Likelihood Inference." *Journal of Statistical Planning and Inference*, **124**, 423–437.

Griffiths WE, Surekha K (1986). "A Monte Carlo Evaluation of the Power of Some Tests for Heteroscedasticity." *Journal of Econometrics*, **31**(1), 219–231.

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

Simonoff JS, Tsai C (1994). "Use of Modified Profile Likelihood for Improved Tests of Constancy of Variance in Regression." *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, **43**(2), 357–370.

Examples

```
mtcars_lm <- lm(mpg ~ wt + qsec + am, data = mtcars)
simonoff_tsai(mtcars_lm, method = "score")
simonoff_tsai(mtcars_lm, method = "score", basetest = "cook_weisberg")
simonoff_tsai(mtcars_lm, method = "mlr")
simonoff_tsai(mtcars_lm, method = "mlr", bartlett = FALSE)
## Not run: simonoff_tsai(mtcars_lm, auxdesign = data.frame(mtcars$wt, mtcars$qsec),
 method = "mlr", hetfun = "logmult")
## End(Not run)
```

---

szroeter   *Szroeter's Test for Heteroskedasticity in a Linear Regression Model*

Description

This function implements the method of Szroeter (1978) for testing for heteroskedasticity in a linear regression model.

Usage

```
szroeter(mainlm, deflator = NA, h = SKH, qfmethod = "imhof", statonly = FALSE)
```

Arguments

mainlm	Either an object of [class](#) "lm" (e.g., generated by [lm](#)), or a list of two objects: a response vector and a design matrix. The objects are assumed to be in that order, unless they are given the names "X" and "y" to distinguish them. The design matrix passed in a list must begin with a column of ones if an intercept is to be included in the linear model. The design matrix passed in a list should not contain factors, as all columns are treated 'as is'. For tests that use ordinary least squares residuals, one can also pass a vector of residuals in the list, which should either be the third object or be named "e".
deflator	Either a character specifying a column name from the design matrix of mainlm or an integer giving the index of a column of the design matrix. This variable is suspected to be related to the error variance under the alternative hypothesis. deflator may not correspond to a column of 1's (intercept). Default NA means the data will be left in its current order (e.g. in case the existing index is believed to be associated with error variance).
h	A non-decreasing function taking as its argument the index i of observations from 1 to n. Defaults to SKH, which is equivalent to $h(i) = 2(1 - \cos\frac{\pi i}{n+1})$. The function must be able to take a vector argument of length n.
qfmethod	A character, either "imhof", "davies", or "integrate", corresponding to the algorithm argument of [pRQF](#). The default is "imhof".
statonly	A logical. If TRUE, only the test statistic value is returned, instead of an object of [class](#) "htest". Defaults to FALSE.

Details

The test entails putting the data rows in increasing order of some specified deflator (e.g., one of the explanatory variables) that is believed to be related to the error variance by some non-decreasing function. The test statistic is a ratio of quadratic forms in the OLS residuals. It is a right-tailed test.

Value

An object of [class](#) "htest". If object is not assigned, its attributes are displayed in the console as a [tibble](#) using [tidy](#).

References

Szroeter J (1978). "A Class of Parametric Tests for Heteroscedasticity in Linear Econometric Models." *Econometrica*, **46**(6), 1311–1327.

Examples

```
mtcars_lm <- lm(mpg ~ wt + qsec + am, data = mtcars)
szroeter(mtcars_lm, deflator = "qsec")
```

---

twosidedpval   *Computation of Conditional Two-Sided p-Values*

Description

Computes the conditional p-value P_C for a continuous or discrete test statistic, as defined in Kulinskaya (2008). This provides a method for computing a two-sided p-value from an asymmetric null distribution.

Usage

```
twosidedpval(
  q,
  CDF,
  continuous,
  method = c("doubled", "kulinskaya", "minlikelihood"),
  locpar,
  supportlim = c(-Inf, Inf),
  ...
)
```

Arguments

q	A double representing the quantile, i.e. the observed value of the test statistic for which a two-sided p-value is to be computed
CDF	A function representing the cumulative distribution function of the test statistic under the null hypothesis, i.e. $\Pr(T \leq q \vert \mathrm{H}_0)$.
continuous	A logical indicating whether the test statistic is a continuous (TRUE) or discrete (FALSE) random variable. Defaults to TRUE.
method	A character specifying the method to use to calculate two-sided p-value; one of "doubled" (representing doubling of the one-sided p-value), "kulinskaya" (representing the method of Kulinskaya (2008)), or "minlikelihood" (representing the sum of probabilities for values with probability less than or equal to that of the observed value. Partial matching is used. Note that the "minlikelihood" method is available only for discrete distributions.
locpar	a double representing a generic location parameter chosen to separate the tails of the distribution. Note that if locpar corresponds to the median of CDF, there is no difference between the two methods, in the continuous case. If locpar is not specified, the function attempts to compute the expectation of CDF using numerical integration and, if successful, uses this as locpar. However, this may yield unexpected results, especially if CDF is not one of the cumulative distribution functions of well-known distributions included in the stats package.
supportlim	A numeric vector of length 2, giving the minimum and maximum values in the support of the distribution whose cumulative distribution function is CDF. This argument is only used if the distribution is discrete (i.e. if continuous is FALSE) and if method is "minlikelihood" or locpar is not specified. If supportlim is not supplied, the function assumes that the support is -1e6:1e6. Values of -Inf and Inf may be supplied, but if so, the support is truncated at -1e6 and 1e6 respectively.
...	Optional arguments to pass to CDF.

Details

Let T be a statistic that, under the null hypothesis, has cumulative distribution function F and probability density or mass function f. Denote by A a generic location parameter chosen to separate the two tails of the distribution. Particular examples include the mean $E(T\vert\mathrm{H}_0)$, the mode $\arg\sup_t f(t)$, or the median $F^{-1}\left(\frac{1}{2}\right)$. Let q be the observed value of T.

In the continuous case, the conditional two-sided p-value centered at A is defined as

$$P_C^A(q) = \frac{F(q)}{F(A)} 1_{q \leq A} + \frac{1 - F(q)}{1 - F(A)} 1_{q > A}$$

where $1.$ is the indicator function. In the discrete case, P_C^A depends on whether A is an attainable value within the support of T. If A is not attainable, the conditional two-sided p-value centred at A is defined as

$$P_C^A(q) = \frac{\Pr(T \leq q)}{\Pr(T < A)} 1_{q < A} + \frac{\Pr(T \geq q)}{\Pr(T > A)} 1_{q > A}$$

If A is attainable, the conditional two-sided p-value centred at A is defined as

$$P_C^A(q) = \frac{\Pr(T \leq q)}{\Pr(T \leq A)/\left(1 + \Pr(T = A)\right)} 1_{q < A} + 1_{q = A} + \frac{\Pr(T \geq q)}{\Pr(T \geq A)/\left(1 + \Pr(T = A)\right)} 1_{q > A}$$

Value

A double.

References

Kulinskaya E (2008). "On Two-Sided p-Values for Non-Symmetric Distributions." 0810.2124,
0810.2124.

Examples

```
# Computation of two-sided p-value for F test for equality of variances
n1 <- 10
n2 <- 20
set.seed(1234)
x1 <- stats::rnorm(n1, mean = 0, sd = 1)
x2 <- stats::rnorm(n2, mean = 0, sd = 3)
# 'Conventional' two-sided p-value obtained by doubling one-sided p-value:
stats::var.test(x1, x2, alternative = "two.sided")$p.value
# This is replicated in `twosidedpval` by setting `method` argument to `"doubled"`
twosidedpval(q = var(x1) / var(x2), CDF = stats::pf, continuous = TRUE,
 method = "doubled", locpar = 1, df1 = n1 - 1, df2 = n2 - 1)
# Conditional two-sided p-value centered at df (mean of chi-squared r.v.):
twosidedpval(q = var(x1) / var(x2), CDF = stats::pf, continuous = TRUE,
 method = "kulinskaya", locpar = 1, df1 = n1 - 1, df2 = n2 - 1)
```

T_alpha   *Pseudorandom numbers from Asymptotic Null Distribution of Test*
                                 Statistic for Method of Rackauskas and Zuokas (2007)

Description

A matrix of 2^14 rows and 16 columns. Each column contains 2^14 Monte Carlo replicates from the
distribution of T_α for a particular value of α, $\alpha = i/32$, $i = 0, 1, \ldots, 15$. The values were generated
by first generating a Brownian Bridge using $m = 2^17$ standard normal variates and then applying
Equation (11) from Rackauskas and Zuokas (2007). It can be used to compute empirical approx-
imate p-values for implementation of the Rackauskas-Zuokas Test for heteroskedasticity. This is
a time-saving measure because, while rackauskas_zuokas contains an option for simulating the
p-value directly, this would be computationally intensive for the authors' recommended m of 2^17.
Passing the argument pvalmethod = "data" to rackauskas_zuokas instructs the function to use
the pre-generated values in this data set to compute the empirical approximate p-value for the test.

Usage

T_alpha

Format

An object of class matrix (inherits from array) with 16384 rows and 16 columns.

---

verbyla   *Verbyla's Test for Heteroskedasticity in a Linear Regression Model*

Description

This function implements the residual maximum likelihood test of Verbyla (1993) for testing for heteroskedasticity in a linear regression model.

Usage

```
verbyla(mainlm, auxdesign = NA, statonly = FALSE)
```

Arguments

mainlm   Either an object of [class](#) "lm" (e.g., generated by [lm](#)), or a list of two objects: a response vector and a design matrix. The objects are assumed to be in that order, unless they are given the names "X" and "y" to distinguish them. The design matrix passed in a list must begin with a column of ones if an intercept is to be included in the linear model. The design matrix passed in a list should not contain factors, as all columns are treated 'as is'. For tests that use ordinary least squares residuals, one can also pass a vector of residuals in the list, which should either be the third object or be named "e".

auxdesign   A [data.frame](#) or [matrix](#) representing an auxiliary design matrix of containing exogenous variables that (under alternative hypothesis) are related to error variance, or a character "fitted.values" indicating that the fitted $\hat{y}_i$ values from OLS should be used. If set to NA (the default), the design matrix of the original regression model is used. An intercept is included in the auxiliary regression even if the first column of auxdesign is not a vector of ones.

statonly   A logical. If TRUE, only the test statistic value is returned, instead of an object of [class](#) "htest". Defaults to FALSE.

Details

Verbyla's Test entails fitting a generalised auxiliary regression model in which the response variable is the vector of standardised squared residuals $e_i^2/\hat{\omega}$ from the original OLS model and the design matrix is some function of Z, an $n \times q$ matrix consisting of q exogenous variables, appended to a column of ones. The test statistic is half the residual sum of squares from this generalised auxiliary regression. Under the null hypothesis of homoskedasticity, the distribution of the test statistic is asymptotically chi-squared with q degrees of freedom. The test is right-tailed.

Value

An object of [class](#) "htest". If object is not assigned, its attributes are displayed in the console as a [tibble](#) using [tidy](#).

References

Verbyla AP (1993). "Modelling Variance Heterogeneity: Residual Maximum Likelihood and Diagnostics." *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, **55**(2), 493–508.

Examples

```
mtcars_lm <- lm(mpg ~ wt + qsec + am, data = mtcars)
verbyla(mtcars_lm)
verbyla(mtcars_lm, auxdesign = "fitted.values")
```

---

white | *White's Test for Heteroskedasticity in a Linear Regression Model*

Description

This function implements the popular method of White (1980) for testing for heteroskedasticity in a linear regression model.

Usage

```
white(mainlm, interactions = FALSE, statonly = FALSE)
```

Arguments

mainlm   Either an object of [class] "lm" (e.g., generated by [lm]), or a list of two objects: a response vector and a design matrix. The objects are assumed to be in that order, unless they are given the names "X" and "y" to distinguish them. The design matrix passed in a list must begin with a column of ones if an intercept is to be included in the linear model. The design matrix passed in a list should not contain factors, as all columns are treated 'as is'. For tests that use ordinary least squares residuals, one can also pass a vector of residuals in the list, which should either be the third object or be named "e".

interactions A logical. Should two-way interactions between explanatory variables be included in the auxiliary regression? Defaults to FALSE, since when interaction terms are present the test is not a pure test of heteroskedasticity but also of model specification.

statonly   A logical. If TRUE, only the test statistic value is returned, instead of an object of [class] "htest". Defaults to FALSE.

Details

White's Test entails fitting an auxiliary regression model in which the response variable is the vector of squared residuals from the original model and the design matrix includes the original explanatory variables, their squares, and (optionally) their two-way interactions. The test statistic is the number of observations multiplied by the coefficient of determination from the auxiliary regression model:

$$T = nr_{\text{aux}}^2$$

White's Test is thus a special case of the method of Breusch and Pagan (1979). Under the null hypothesis of homoskedasticity, the distribution of the test statistic is asymptotically chi-squared with `parameter` degrees of freedom. The test is right-tailed.

Value

An object of [class](#) `"htest"`. If object is not assigned, its attributes are displayed in the console as a [tibble](#) using [tidy](#).

References

Breusch TS, Pagan AR (1979). "A Simple Test for Heteroscedasticity and Random Coefficient Variation." *Econometrica*, **47**(5), 1287–1294.

White H (1980). "A Heteroskedasticity-Consistent Covariance Matrix Estimator and a Direct Test for Heteroskedasticity." *Econometrica*, **48**(4), 817–838.

See Also

This function should not be confused with [tseries::white.test](#), which does *not* implement the method of White (1980) for testing for heteroskedasticity in a linear model.

Examples

```
mtcars_lm <- lm(mpg ~ wt + qsec + am, data = mtcars)
white(mtcars_lm)
white(mtcars_lm, interactions = TRUE)
```

---

wilcox_keselman	*Wilcox and Keselman's Test for Heteroskedasticity in a Linear Regression Model*

Description

This function implements the nonparametric test of Wilcox and Keselman (2006) for testing for heteroskedasticity in a simple linear regression model, and extends it to the multiple linear regression model.

Usage

```
wilcox_keselman(
  mainlm,
  gammapar = 0.2,
  B = 500L,
  p.adjust.method = "none",
  seed = NA,
  rqwarn = FALSE,
  matchWRS = FALSE,
  statonly = FALSE
)
```

Arguments

mainlm	Either an object of [class] "lm" (e.g., generated by [lm]), or a list of two objects: a response vector and a design matrix. The objects are assumed to be in that order, unless they are given the names "X" and "y" to distinguish them. The design matrix passed in a list must begin with a column of ones if an intercept is to be included in the linear model. The design matrix passed in a list should not contain factors, as all columns are treated 'as is'. For tests that use ordinary least squares residuals, one can also pass a vector of residuals in the list, which should either be the third object or be named "e".
gammapar	A double value between 0 and 0.5 exclusive specifying the quantile value $gamma$. Defaults to 0.2.
B	An integer specifying the number of nonparametric bootstrap samples to use to estimate standard error(s) of the quantile difference(s). Defaults to 500L.
p.adjust.method	
	A character specifying the family-wise error rate method to use in adjusting p-values (if it is a multiple linear regression model). The value is passed to [p.adjust]. By default no adjustment is made.
seed	An integer specifying a seed to pass to [set.seed] for random number generation. This allows reproducibility of bootstrap results. The value NA results in not setting a seed.
rqwarn	A logical specifying whether warnings generated by [rq.fit] (such as 'Solution may be nonunique') should be printed (TRUE) or suppressed (FALSE). Defaults to FALSE.
matchWRS	A logical specifying whether bootstrap samples should be generated in the exact same manner as in the qhomtv2 function in [WRS] package. If TRUE, and seed is set to 2 and B to 100 and p.adjust.method to "none", results will be identical to those of the default settings of qhomtv2.
statonly	A logical. If TRUE, only the test statistic value is returned, instead of an object of [class] "htest". Defaults to FALSE.

Value

An object of [class] "htest". If object is not assigned, its attributes are displayed in the console as a [tibble] using [tidy].

References

Wilcox RR, Keselman HJ (2006). "Detecting Heteroscedasticity in a Simple Regression Model via Quantile Regression Slopes." *Journal of Statistical Computation and Simulation*, **76**(8), 705–712.

See Also

Rand R. Wilcox's package WRS on Github; in particular the functions qhomt and qhomtv2, which implement this method for simple and multiple linear regression respectively.

Examples

```
mtcars_lm <- lm(mpg ~ wt + qsec + am, data = mtcars)
wilcox_keselman(mtcars_lm)
```

---

yuce   *Yüce's Test for Heteroskedasticity in a Linear Regression Model*

Description

This function implements the two methods of Yüce (2008) for testing for heteroskedasticity in a linear regression model.

Usage

```
yuce(mainlm, method = c("A", "B"), statonly = FALSE)
```

Arguments

mainlm   Either an object of class "lm" (e.g., generated by lm), or a list of two objects: a response vector and a design matrix. The objects are assumed to be in that order, unless they are given the names "X" and "y" to distinguish them. The design matrix passed in a list must begin with a column of ones if an intercept is to be included in the linear model. The design matrix passed in a list should not contain factors, as all columns are treated 'as is'. For tests that use ordinary least squares residuals, one can also pass a vector of residuals in the list, which should either be the third object or be named "e".

method   A character indicating which of the two tests presented in Yüce (2008) should be implemented. Possible values are "A" (the chi-squared test) and "B" (the t-test). Partial matching is used and the argument is not case-sensitive.

statonly   A logical. If TRUE, only the test statistic value is returned, instead of an object of class "htest". Defaults to FALSE.

Details

These two tests are straightforward to implement; in both cases the test statistic is a function only of the residuals of the linear regression model. The first test statistic has an asymptotic chi-squared distribution and the second has an asymptotic t-distribution. In both cases the degrees of freedom are $n - p$. The chi-squared test is right-tailed whereas the t-test is two-tailed.

Value

An object of class "htest". If object is not assigned, its attributes are displayed in the console as a tibble using tidy.

References

Yüce M (2008). "An Asymptotic Test for the Detection of Heteroskedasticity." *Istanbul University Econometrics and Statistics e-Journal*, **8**, 33–44.

Examples

```
mtcars_lm <- lm(mpg ~ wt + qsec + am, data = mtcars)
yuce(mtcars_lm, method = "A")
yuce(mtcars_lm, method = "B")
```

---

zhou_etal   *Zhou, Song, and Thompson's Test for Heteroskedasticity in a Linear Regression Model*

Description

This function implements the methods of Zhou et al. (2015) for testing for heteroskedasticity in a linear regression model.

Usage

```
zhou_etal(
  mainlm,
  auxdesign = NA,
  method = c("pooled", "covariate-specific", "hybrid"),
  Bperturbed = 500L,
  seed = 1234,
  statonly = FALSE
)
```

Arguments

mainlm   Either an object of class "lm" (e.g., generated by [lm](lm)), or a list of two objects:
                a response vector and a design matrix. The objects are assumed to be in that
                order, unless they are given the names "X" and "y" to distinguish them. The
                design matrix passed in a list must begin with a column of ones if an intercept
                is to be included in the linear model. The design matrix passed in a list should
                not contain factors, as all columns are treated 'as is'. For tests that use ordinary
                least squares residuals, one can also pass a vector of residuals in the list, which
                should either be the third object or be named "e".

auxdesign	A [data.frame](#) or [matrix](#) representing an auxiliary design matrix of containing exogenous variables that (under alternative hypothesis) are related to error variance, or a character "fitted.values" indicating that the fitted $\hat{y}_i$ values from OLS should be used. If set to NA (the default), the design matrix of the original regression model is used. An intercept is included in the auxiliary regression even if the first column of auxdesign is not a vector of ones.
method	A character specifying which of the three test methods to implement; one of "pooled", "covariate-specific", or "hybrid" (which combines the other two). Partial matching is used.
Bperturbed	An integer specifying the number of perturbation samples to generate when estimating the p-value. Defaults to 500L.
seed	An integer specifying a seed to pass to [set.seed](#) for random number generation. This allows for reproducibility of perturbation sampling. A value of NA results in not setting a seed.
statonly	A logical. If TRUE, only the test statistic value is returned, instead of an object of [class](#) "htest". Defaults to FALSE.

Details

Zhou et al. (2015) The authors propose three variations based on comparisons between sandwich and model-based esimators for the variances of individual regression coefficient esimators. The covariate-specific method computes a test statistic and p-value for each column of the auxiliary design matrix (which is, by default, the original design matrix with intercept omitted). The p-values undergo a Bonferroni correction to control overall test size. When the null hypothesis is rejected in this case, it also provides information about which auxiliary design variable is associated with the error variance. The pooled method computes a single test statistic and p-value and is thus an omnibus test. The hybrid method returns the minimum p-value between the Bonferroni-corrected covariate-specific p-values and the pooled p-value, multiplying it by 2 for a further Bonferroni correction. The test statistic returned is that which corresponds to the minimum p-value. The covariate-specific and pooled test statistics have null distributions that are asymptotically normal with mean 0. However, the variance is intractable and thus perturbation sampling is used to compute p-values empirically.

Value

An object of [class](#) "htest". If object is not assigned, its attributes are displayed in the console as a [tibble](#) using [tidy](#).

References

Zhou QM, Song PX, Thompson ME (2015). "Profiling Heteroscedasticity in Linear Regression Models." *The Canadian Journal of Statistics*, **43**(3), 358–377.

Examples

```
mtcars_lm <- lm(mpg ~ wt + qsec + am, data = mtcars)
zhou_etal(mtcars_lm, method = "pooled")
zhou_etal(mtcars_lm, method = "covariate-specific")
```

```
zhou_etal(mtcars_lm, method = "hybrid")
```

Index