Package ‘hettx’

February 24, 2026

Type Package

Title Fisherian and Neymanian Methods for Detecting and Measuring Treatment Effect Variation

Version 1.0.1

Date 2026-02-24

Description Implements methods developed by Ding, Feller, and Miratrix (2016) <doi:10.1111/rssb.12124> <doi:10.48550/arXiv.1412.5000>, and Ding, Feller, and Miratrix (2018) <doi:10.1080/01621459.2017.1407322> <doi:10.48550/arXiv.1605.06566> for testing whether there is unexplained variation in treatment effects across observations, and for characterizing the extent of the explained and unexplained variation in treatment effects. The package includes wrapper functions implementing the proposed methods, as well as helper functions for analyzing and visualizing the results of the test.

License GPL (>= 3)

Imports generics, quantreg, mvtnorm, MASS, foreach, parallel, doParallel, moments, ggplot2

Depends R (>= 4.0.0)

BugReports `https://github.com/bfifield/hettx/issues`

RoxygenNote 7.3.3

Suggests testthat, knitr, rmarkdown, dplyr, tidyr, purrr

Encoding UTF-8

VignetteBuilder knitr

LazyData true

NeedsCompilation no

Author Peng Ding [aut], Avi Feller [aut], Ben Fifield [aut, cre], Luke Miratrix [aut]

Maintainer Ben Fifield <`benfifield@gmail.com`>

Repository CRAN

Date/Publication 2026-02-24 15:20:02 UTC

Contents

`hettx-package` *Fisherian and Neymanian Methods for Detecting and Measuring Treatment Effect Variation*

Description

This package implements methods developed by Ding, Feller, and Miratrix (JRSS-B, 2016) "Randomization Inference for Treatment Effect Variation", for validly testing whether there is unexplained variation in treatment effects across observations. The package also implements methods introduced in Ding, Feller, and Miratrix (JASA, 2019) "Decomposing Treatment Effect Variation", for measuring the degree of treatment effect heterogeneity explained by covariates. The package includes wrapper functions implementing the proposed methods, as well as helper functions for analyzing and visualizing the results of the tests.

Details

This package partially supported by the Institute of Education Sciences, U.S. Department of Education, through Grant R305D150040. The opinions expressed are those of the authors and do not represent views of the Institute or the U.S. Department of Education.

Special thanks to Masha Bertling for some early work on documenting this project.

Author(s)

Peng Ding, Avi Feller, Ben Fifield, and Luke Miratrix

Maintainer: Ben Fifield `<benfifield@gmail.com>`

References

Ding, Peng, Avi Feller and Luke Miratrix. (2016) "Randomization Inference for Treatment Effect Variation", Journal of the Royal Statistical Society-Series B. Ding, Peng, Avi Feller and Luke Miratrix. (2019) "Decomposing Treatment Effect Variation", Journal of the American Statistical Association.

See Also

Useful links:

- Report bugs at `https://github.com/bfifield/hettx/issues`

`coef.RI.regression.result` *Extract coefficients of a fit RI regression model.*

Description

Extract coefficients of a fit RI regression model.

Usage

```
## S3 method for class 'RI.regression.result'
coef(object, ...)
```

Arguments

`object`	A RI.regression.result object.
`...`	Unused

Examples

```
df <- make_randomized_dat( 1000, gamma.vec=c(1,1,1,2), beta.vec=c(-1,-1,1,0) )
es <- estimate_systematic( Yobs ~ Z,  interaction.formula = ~ A + B, data = df )
coef(es)
```

`detect_idiosyncratic` *detect_idiosyncratic*

Description

Test for systematic treatment effect heterogeneity using Fisherian permutation inference methods.

Usage

```
detect_idiosyncratic(
  formula,
  data,
  interaction.formula = NULL,
  control.formula = NULL,
  plugin = FALSE,
  tau.hat = NULL,
  test.stat = ifelse(is.null(W) & is.null(X), "SKS_stat", ifelse(is.null(W),
    "SKS_stat_cov", "SKS_stat_int_cov")),
  te.vec = NULL,
  B = 500,
  gamma = 1e-04,
  grid.gamma = 100 * gamma,
  grid.size = 151,
  return.matrix = FALSE,
  na.rm = FALSE,
  n.cores = 1,
  verbose = TRUE,
  ...
)
```

Arguments

`formula`	An object of class formula, as in lm(), such as Y ~ Z with only the treatment variable on the right-hand side.
`data`	A data.frame, tbl_df, or data.table with the input data.

`interaction.formula`	A right-sided formula with pre-treatment covariates to model treatment effects for on the right hand side, such as ~ x1 + x2 + x3. Defaultis NULL (no interactions modeled)
`control.formula`	A right-sided formula with pre-treatment covariates to adjust for on the right hand side, such as ~ x1 + x2 + x3. Default is NULL (no variables adjusted for)
`plugin`	Whether to calculate the plug-in p-value without sweeping over range of possible treatment effect magnitudes. Default is FALSE.
`tau.hat`	The value of the plug-in treatment effect. Default is sample average treatment effect.
`test.stat`	Test statistic function to use on the data. Default is shifted Kolmogorov-Smirnov statistic, potentially with covariate adjustment depending on passed arguments. test.stat can be a string name for a test statistic function, or the function itself.
`te.vec`	Vector of taus to examine if you want to override generating ones automatically. Default is NULL.
`B`	Number of permutations to take. Default is 500.
`gamma`	How wide of a CI to make around tau-hat for search. Default is 0.0001.
`grid.gamma`	Parameter to govern where the grid points are sampled. Bigger values means more samples towards the estimated tau-hat. Default is 100*gamma.
`grid.size`	Number of points in the grid. Default is 151.
`return.matrix`	Whether to return the matrix of all the imputed statistics. Default is FALSE.
`na.rm`	A logical flag indicating whether to list-wise delete missing data. The function will report an error if missing data exist. Default is FALSE.
`n.cores`	Number of cores to use to parallelize permutation step. Default is 1.
`verbose`	Whether to print out progress bar when fitting and other diagnostics. Default is TRUE.
`...`	Extra arguments passed to the generate_permutations function and test.stat functions.

Value

If plug-in, the value of the test and the associated p-value. If not, a list with the value of the test statistic on the observed data, the value of the CI-adjusted p-value, the plug-in p-value, and other information on the test.

Examples

```
Z <- rep(c(0, 1), 100)
tau <- 4
Y <- ifelse(Z, rnorm(100, tau), rnorm(100, 0))
df <- data.frame(Y=Y, Z=Z)
tst <- detect_idiosyncratic(Y ~ Z, df, B = 50, grid.size = 50)
```

`estimate_systematic` *Calculate systematic effects model using LATE, ITT, or full potential outcomes.*

Description

Implements the systematic effects model proposed in Ding, Feller, and Miratrix (2018). Can estimate an ITT or LATE model, or the actual beta in cases where full potential outcomes schedule is available.

Usage

```
estimate_systematic(
  formula,
  data,
  interaction.formula,
  control.formula = NULL,
  method = c("RI", "OLS", "2SLS"),
  na.rm = FALSE
)
```

Arguments

`formula`	An object of class formula, as in lm(). For ITT estimation, specify as Y ~ Z with only the treatment variable on the right-hand side. For LATE estimation, specify as Y ~ D | Z with only the endogenous variable (D) and the instrument (Z) on the right-hand side separated by a vertical bar (|). For oracle estimation (where full potential outcome schedule is known), specify as Y(1) + Y(0) ~ Z with only the treatment variable on the right-hand side and the variables indicating the outcome under treatment and the outcome under control on the left-hand-side. The first variable on the left-hand-side will be treated as the outcome under treatment, and the second variable on the right-hand-side will be treated as the outcome under control.
`data`	A data.frame, tbl_df, or data.table with the input data.
`interaction.formula`	A right-sided formula with pre-treatment covariates to model treatment effects for on the right hand side, such as ~ x1 + x2 + x3.
`control.formula`	A right-sided formula with pre-treatment covariates to adjust for on the right hand side, such as ~ x1 + x2 + x3. Default is NULL (no variables adjusted for). Will be ignored for LATE estimation and oracle estimation. Default is NULL
`method`	RI or OLS (for ITT and oracle), RI or 2SLS (for LATE). method=OLS is shorthand for setting the empirical.Sxx variable to TRUE, nothing more.
`na.rm`	A logical flag indicating whether to list-wise delete missing data. The function will report an error if missing data exist. Default is FALSE.

Details

The OLS method differs from the RI method only by how the Sxx matrix is handled. In the OLS case, seperate Sxx for treatment and control are calculated for each treatment arm. For RI the known Sxx based on all units is used.

Examples

```
df <- make_randomized_dat( 1000, gamma.vec=c(1,1,1,2), beta.vec=c(-1,-1,1,0) )
es <- estimate_systematic( Yobs ~ Z,  interaction.formula = ~ A + B, data = df )
```

`get_p_value` *get p-value along with uncertainty on p-value*

Description

Give confidence bounds (from monte carlo simulation error) for the p-values returned by a test

Usage

```
get_p_value(tst)
```

Arguments

`tst`	A FRTCI.test object from detect_idiosyncratic()

Value

p-value and range of p-values due to monte carlo error.

Examples

```
Z <- rep(c(0, 1), 100)
tau <- 4
Y <- ifelse(Z, rnorm(100, tau), rnorm(100, 0))
df <- data.frame(Y=Y, Z=Z)
tst <- detect_idiosyncratic(Y ~ Z, df, B = 50, grid.size = 50)
get_p_value( tst )
```

`glance.FRTCI.test` *Glance at a FRTCI.test result*

Description

Glance at a FRTCI.test result

Usage

```
## S3 method for class 'FRTCI.test'
glance(x, ...)
```

Arguments

`x`	A `FRTCI.test` object from `detect_idiosyncratic()`.
`...`	Additional arguments (ignored).

Value

A one-row data.frame with model-level summary statistics.

`glance.RI.R2.result` *Glance at an RI.R2.result*

Description

Glance at an RI.R2.result

Usage

```
## S3 method for class 'RI.R2.result'
glance(x, ...)
```

Arguments

`x`	An `RI.R2.result` object from `R2()`.
`...`	Additional arguments (ignored).

Value

A one-row data.frame with model-level summary statistics.

glance.RI.regression.result — *Glance at an RI.regression.result*

Description

Glance at an RI.regression.result

Usage

```
## S3 method for class 'RI.regression.result'
glance(x, ...)
```

Arguments

`x`	An `RI.regression.result` object from `estimate_systematic()`.
`...`	Additional arguments (ignored).

Value

A one-row data.frame with model-level summary statistics.

KS_stat — *KS_stat*

Description

Calculate classic (not shifted) KS statistic; code is a modified version of R's ks.test().

Usage

```
KS_stat(Y, Z, tau = NULL, alternative = c("two.sided", "less", "greater"))
```

Arguments

`Y`	Observed outcome vector
`Z`	Treatment assigment vector
`tau`	Value of treatment effect for shifting Y1. Default is NULL (Y1 not shifted).
`alternative`	Direction of test ("two.sided", "less", "greater")

Details

If tau passed, Y1 will be shifted by tau.

Value

The value of the test.

See Also

detect_idiosyncratic

Examples

```
df <- make_randomized_dat( 1000, gamma.vec=c(1,1,1,2), beta.vec=c(-1,-1,1,0) )
KS_stat(df$Yobs, df$Z)
```

---

make_linear_data *Generate dataset according to a linear model.*

Description

Given the parameters, generate a dataset and return a potential outcomes schedule (science table) of synthetic potential outcomes.

Usage

```
make_linear_data(
  n,
  gamma.vec = c(1, 2, 2, 1),
  gamma2.vec = NULL,
  beta.vec = c(-1, -1, 1),
  ideo.sd = 0,
  quad.tx = FALSE,
  mu.X = FALSE,
  corr.X = TRUE
)

make_quadradic_data(n, beta.vec = c(-1, -1, 1))

make_skew_data(n, beta.vec = c(-1, -1, 1))
```

Arguments

`n`	Sample size
`gamma.vec`	Control outcome surface
`gamma2.vec`	Quadratic terms
`beta.vec`	Treatment effect surface
`ideo.sd`	Ideosyncratic residual variation
`quad.tx`	Quadratic treatment effects?

`mu.X`	Center of the X covariates (can be single number or vector of length equal to the max of the length of gamma.vec, gamma2.vec, and beta.vec)
`corr.X`	TRUE or FALSE. Have Xs correlated or no.

Details

The control outcome surface is either linear or quadratic, of the form:

$$Y_i = gamma_0 + sum_{k=1}^{J} gamma_k X_{ki} + sum_{k=1}^{J_2} gamma_k^{(2)} X_{ki}^2 + epsilon_i$$

The individual treatment effects are similarly a linear or quadratic model.

Value

List of elements of data (not data frame)

Functions

- `make_quadradic_data()`: Generate dataset according to a quadratic model
- `make_skew_data()`: Generate dataset with a skew

`make_randomized_compliance_dat`

Generate fake data with noncompliance.

Description

This will generate and randomize a science table to get observed outcomes and treatment assignment

Usage

```
make_randomized_compliance_dat(
  n,
  p = 0.6,
  science.table.generator = make_linear_data,
  include.POs = TRUE,
  ...
)
```

Arguments

`n`	Sample size
`p`	Proportion treated
`science.table.generator`	Method to generate potential outcomes
`include.POs`	Preserve potential outcomes in returned value
`...`	To be passed to science.table.generator

Value

Data frame with data randomized to tx and control, and compliers, etc.

See Also

make_randomized_dat

`make_randomized_dat` *Make fake data for simulations*

Description

Randomize a science table to get observed outcomes and treatment assignment

Usage

```
make_randomized_dat(
  n,
  p = 0.6,
  science.table.generator = make_linear_data,
  include.POs = TRUE,
  as.data.frame = TRUE,
  ...
)
```

Arguments

`n`	Sample size
`p`	Proportion treated
`science.table.generator`	Data generator
`include.POs`	TRUE/FALSE. Keep POs
`as.data.frame`	TRUE/FALSE. Return as dataframe or as list of elements.
`...`	Additional to be passed to science.table.generator

Value

Either a list of elements or a dataframe.

Penn46_ascii *Sample data set*

Description

This is a sample data set to illustrate the package methods.

Usage

```
Penn46_ascii
```

Format

A dataframe containing 6384 observations and 12 columns.

plot.FRTCI.test *plot.FRTCI.test*

Description

Plot curve from FRTCI.test object.

Usage

```
## S3 method for class 'FRTCI.test'
plot(
  x,
  true.tau = NULL,
  xlab = expression(tau),
  ylab = "p-value",
  true.tau.col = "red",
  plot.envelope = TRUE,
  ci.line.col = "blue",
  ...
)
```

Arguments

`x`	An object of class `FRTCI.test`
`true.tau`	The true value of tau, if known. Default is NULL.
`xlab`	X-axis label. Default is tau.
`ylab`	Y-axis label. Default is "p-value".
`true.tau.col`	Color to plot true tau value, if provided. Default is red.
`plot.envelope`	Plot envelope around tested values of tau. Default is TRUE.

`ci.line.col`	Color to plot confidence interval around estimated treatment effect. Default is blue.
`...`	Further arguments to be passed to `print.FRTCI.test()`

Examples

```
Z <- rep(c(0, 1), 100)
tau <- 4
Y <- ifelse(Z, rnorm(100, tau), rnorm(100, 0))
df <- data.frame(Y=Y, Z=Z)
tst <- detect_idiosyncratic(Y ~ Z, df, B = 50, grid.size = 50)
plot(tst)
```

`plot.RI.R2.result` *Make a plot of the treatment effect R2 estimates*

Description

Make a plot of the treatment effect R2 estimates

Usage

```
## S3 method for class 'RI.R2.result'
plot(
  x,
  main = paste("R2 for Het Tx (", x$type, ")", sep = ""),
  ADD = FALSE,
  ...
)
```

Arguments

`x`	Results from est.beta, etc.
`main`	Title for plot
`ADD`	TRUE if add to existing plot. FALSE make a new plot.
`...`	Arguments to pass to plotting of points.

See Also

calc_beta_oracle

Examples

```
df <- make_randomized_dat( 1000, gamma.vec=c(1,1,1,2), beta.vec=c(-1,-1,1,0) )
es <- estimate_systematic( Yobs ~ Z,  interaction.formula = ~ A + B, data = df )
r2_out <- R2(es)
plot(r2_out)
```

R2 *Estimate treatment variation R2*

Description

Bounds the R2 measure (how much of treatment variation is explained by given covariates) using either the OLS output for the ITT from est.beta, or the LATE estimation from est.beta.

Usage

```
R2(est.beta, rho.step = 0.05)
```

Arguments

`est.beta`	The output from ‘est.beta()‘. Either an estimate of overall systematic effect variation, or systematic effect variation for compliers.
`rho.step`	Grid size for sensitivity analysis on values of rho. Default is 0.05

Value

RI.R2.result object.

See Also

print.RI.R2.result

Examples

```
df <- make_randomized_dat( 1000, gamma.vec=c(1,1,1,2), beta.vec=c(-1,-1,1,0) )
es <- estimate_systematic( Yobs ~ Z,  interaction.formula = ~ A + B, data = df )
r2_out <- R2(es)
```

rq_stat *rq_stat*

Description

rq_stat is the Kolmogorov-smirnov statistic via quantile regression with covariates without further adjustment.

rq_stat_cond_cov does Kolmogorov-smirnov statistic via quantile regression with covariates, with a conditional approach; see Koenker and Xiao (2002).

rq_stat_uncond_cov implements a Kolmogorov-smirnov statistic via quantile regression with covariates, unconditional approach; see Firpo (2007).

Usage

```
rq_stat(Y, Z, rq.pts = seq(0.1, 0.9, by = 0.1))

rq_stat_cond_cov(Y, Z, X, rq.pts = seq(0.1, 0.9, by = 0.1))

rq_stat_uncond_cov(Y, Z, X, rq.pts = seq(0.1, 0.9, by = 0.1))
```

Arguments

Y	Observed outcome vector
Z	Treatment assigment vector
rq.pts	Sequence of quantile points at which to evaluate the test. Default is seq(.1, .9, by = .1). Should not go beyond 0 and 1.
X	Additional pre-treatment covariates to adjust for in estimation, but not to interact with treatment.

Details

Warning: This function supresses all warnings of the ‘rq()‘ method call.

Warning: This function supresses all warnings of the ‘rq()‘ method call.

Value

The value of the test.

Examples

```
df <- make_randomized_dat( 1000, gamma.vec=c(1,1,1,2), beta.vec=c(-1,-1,1,0) )
rq_stat(df$Yobs, df$Z)

df <- make_randomized_dat( 1000, gamma.vec=c(1,1,1,2), beta.vec=c(-1,-1,1,0) )
rq_stat_cond_cov(df$Yobs, df$Z, df$A)

df <- make_randomized_dat( 1000, gamma.vec=c(1,1,1,2), beta.vec=c(-1,-1,1,0) )
rq_stat_uncond_cov(df$Yobs, df$Z, df$A)
```

SE *Extract the standard errors from a var-cov matrix.*

Description

Extract the standard errors from a var-cov matrix.

Usage

```
SE(object, ...)
```

Arguments

`object`	est.beta object
`...`	unused

Examples

```
df <- make_randomized_dat( 1000, gamma.vec=c(1,1,1,2), beta.vec=c(-1,-1,1,0) )
es <- estimate_systematic( Yobs ~ Z,  interaction.formula = ~ A + B, data = df )
SE(es)
```

`SKS_pool_t` *SKS_pool_t*

Description

Subtract off group level treatment effect estimates and then look at KS statistic on residuals.

Usage

```
SKS_pool_t(Y, Z, W)
```

Arguments

`Y`	Observed outcome vector
`Z`	Treatment assigment vector
`W`	A a factor or categorical covariate.

Details

Distinct from the interacted lm in that the control units are not shifted and centered with respect to eachother.

Examples

```
df <- make_randomized_dat( 1000, gamma.vec=c(1,1,1,2), beta.vec=c(-1,-1,1,0) )
df$W <- sample(c("A", "B", "C"), nrow(df), replace = TRUE)
SKS_pool_t(df$Yobs, df$Z, df$W)
```

SKS_stat *SKS_stat*

Description

Shifted kolmogorov-smirnov statistic. Calculate KS distance between Y0 and Y1 shifted by sample tau.

Usage

```
SKS_stat(Y, Z)
```

Arguments

Y	Observed outcome vector
Z	Treatment assigment vector

Value

The value of the test.

See Also

KS_stat, SKS_stat_cov

detect_idiosyncratic

Examples

```
df <- make_randomized_dat( 1000, gamma.vec=c(1,1,1,2), beta.vec=c(-1,-1,1,0) )
SKS_stat(df$Yobs, df$Z)
```

SKS_stat_cov_pool *SKS_stat_cov_pool*

Description

SKS_stat_cov_pool is the shifted kolmogorov-smirnov statistic with covariates to increase precision. This is the test statistic used Ding, Feller, and Miratrix (2016), JRSS-B.

SKS_stat_cov is the shifted kolmogorov-smirnov statistic with covariates with model for outcomes calculated on control group only. This avoids "splitting" the treatment variation between tx and co groups. We recommend this method over the "pool" method.

Usage

```
SKS_stat_cov_pool(Y, Z, X)

SKS_stat_cov(Y, Z, X)
```

Arguments

`Y`	Observed outcome vector
`Z`	Treatment assigment vector
`X`	Additional pre-treatment covariates to adjust for in estimation, but not to interact with treatment.

Value

The value of the test.

Examples

```
df <- make_randomized_dat( 1000, gamma.vec=c(1,1,1,2), beta.vec=c(-1,-1,1,0) )
SKS_stat_cov_pool(df$Yobs, df$Z, df$A)

df <- make_randomized_dat( 1000, gamma.vec=c(1,1,1,2), beta.vec=c(-1,-1,1,0) )
SKS_stat_cov(df$Yobs, df$Z, df$A)
```

`SKS_stat_cov_rq` *SKS_stat_cov_rq*

Description

Shifted kolmogorov-smirnov statistic with covariates and quantile regression.

Usage

```
SKS_stat_cov_rq(Y, Z, X)
```

Arguments

`Y`	Observed outcome vector
`Z`	Treatment assigment vector
`X`	Additional pre-treatment covariates to adjust for in estimation, but not to interact with treatment.

Value

The test statistic value.

Examples

```
df <- make_randomized_dat( 1000, gamma.vec=c(1,1,1,2), beta.vec=c(-1,-1,1,0) )
SKS_stat_cov_rq(df$Yobs, df$Z, df$A)
```

SKS_stat_int_cov_pool *SKS_stat_int_cov_pool*

Description

SKS_stat_int_cov_pool is a shifted kolmogorov-smirnov statistic with a linear treatment effect model defined by W. It will attempt to remove any systematic variation corresponding to W and then return a SKS statistic on the residuals to measure any variation "left over".

SKS_stat_int_cov() is a Shifted kolmogorov-smirnov statistic with a linear treatment effect model defined by W. It will attempt to remove any systematic variation corresponding to W and then return a SKS statistic on the residuals to measure any variation "left over".

Usage

```
SKS_stat_int_cov_pool(Y, Z, W, X = NULL)

SKS_stat_int_cov(Y, Z, W, X = NULL)
```

Arguments

`Y`	Observed outcome vector
`Z`	Treatment assigment vector
`W`	Additional pre-treatment covariates to interact with T to define linear model of treatment effects.
`X`	Additional pre-treatment covariates to adjust for in estimation, but not to interact with treatment.

Details

X are _additional_ covariates to adjust for beyond those involved in treatment effect model. It will automatically ajust for W as well. Do not put a covariate in for both X and W.

This is the test statistic used in Ding, Feller, and Miratrix (2016), JRSS-B.

SKS_stat_int_cov first adjusts for baseline and then models treatment effect on the residuals to not split treatment effects (see the vignette for more information on this).

We recommend SKS_stat_int_cov over the "pool" method.

Examples

```
df <- make_randomized_dat( 1000, gamma.vec=c(1,1,1,2), beta.vec=c(-1,-1,1,0) )
SKS_stat_int_cov_pool(Y = df$Yobs, Z = df$Z, W = df$A, X = df$B)


df <- make_randomized_dat( 1000, gamma.vec=c(1,1,1,2), beta.vec=c(-1,-1,1,0) )
SKS_stat_int_cov(Y = df$Yobs, Z = df$Z, W = df$A, X = df$B)
```

`test_stat_info` *test_stat_info*

Description

A list of test statistics for detect.idiosyncratic(), and information on use cases when each is appropriate.

Usage

```
test_stat_info()
```

Examples

```
test_stat_info()
```

`tidy.FRTCI.test` *Tidy a FRTCI.test result*

Description

Tidy a FRTCI.test result

Usage

```
## S3 method for class 'FRTCI.test'
tidy(x, ...)
```

Arguments

`x`	A `FRTCI.test` object from `detect_idiosyncratic()`.
`...`	Additional arguments (ignored).

Value

A data.frame with columns: statistic, p.value, p.value.plug, method, test.stat, estimate, std.error.

`tidy.RI.R2.result` *Tidy an RI.R2.result*

Description

Tidy an RI.R2.result

Usage

```
## S3 method for class 'RI.R2.result'
tidy(x, ...)
```

Arguments

`x`	An `RI.R2.result` object from `R2()`.
`...`	Additional arguments (ignored).

Value

A data.frame with R-squared bound estimates. For ITT results, one row. For LATE results, three rows (Compliers, Noncompliers, Covariates and compliers).

`tidy.RI.regression.result` *Tidy an RI.regression.result*

Description

Tidy an RI.regression.result

Usage

```
## S3 method for class 'RI.regression.result'
tidy(x, ...)
```

Arguments

`x`	An `RI.regression.result` object from `estimate_systematic()`.
`...`	Additional arguments (ignored).

Value

A data.frame with one row per coefficient, containing columns: term, estimate, std.error.

`ToyData` *Toy data set*

Description

This is a toy data set to illustrate the package methods.

Usage

```
ToyData
```

Format

A dataframe containing 500 observations and 7 columns.

`variance_ratio_test` *Variance ratio test*

Description

Given vector of observed outcomes and treatment vector, test to see if there is evidence the variances are different (taking kurtosis into account).

Usage

```
variance_ratio_test(Yobs, Z, data = NULL)
```

Arguments

`Yobs`	Outcome
`Z`	Treatment assignment vector
`data`	Dataframe with variables listed in formula and control.formula

Examples

```
df <- make_randomized_dat( 1000, gamma.vec=c(1,1,1,2), beta.vec=c(-1,-1,1,0) )
variance_ratio_test(df$Yobs, df$Z)
```

`vcov.RI.regression.result` *Get vcov() from object.*

Description

Get vcov() from object.

Usage

```
## S3 method for class 'RI.regression.result'
vcov(object, ...)
```

Arguments

`object`	est.beta object
`...`	unused

Examples

```
df <- make_randomized_dat( 1000, gamma.vec=c(1,1,1,2), beta.vec=c(-1,-1,1,0) )
es <- estimate_systematic( Yobs ~ Z,  interaction.formula = ~ A + B, data = df )
vcov(es)
```

`WSKS_t` *WSKS_t*

Description

Weighted average of the group-level SKS statistics. This is useful for a blocked experiment.

Usage

```
WSKS_t(Y, Z, W)
```

Arguments

`Y`	Observed outcome vector
`Z`	Treatment assigment vector
`W`	A a factor or categorical covariate.

Value

The value of the test.

Examples

```
df <- make_randomized_dat( 1000, gamma.vec=c(1,1,1,2), beta.vec=c(-1,-1,1,0) )
df$W <- sample(c("A", "B", "C"), nrow(df), replace = TRUE)
WSKS_t(df$Yobs, df$Z, df$W)
```

Index