Package 'miceafter'

October 13, 2022

Type Package

Depends R (>= 4.0.0),

Imports survival (>= 3.1-12), pROC (>= 1.16.2), rms (>= 6.1-0), mice (>= 3.12.0), mitml (>= 0.3-7), mitools (>= 2.4), dplyr (>= 1.0.2), purrr (>= 0.3.4), tidyr (>= 1.1.2), tibble (>= 3.0.4), stringr (>= 1.4.0), car (>= 3.0-10), rlang, magrittr

Suggests foreign (>= 0.8-80), knitr, rmarkdown, testthat (>= 3.0.0), bookdown, readr

Title Data and Statistical Analyses after Multiple Imputation

Version 0.5.0

Description Statistical Analyses and Pooling after Multiple Imputation. A large variety of repeated statistical analysis can be performed and finally pooled. Statistical analysis that are available are, among others, Levene's test, Odds and Risk Ratios, One sample proportions, difference between proportions and linear and logistic regression models. Functions can also be used in combination with the Pipe operator.
More and more statistical analyses and pooling functions will be added over time.
Heymans (2007) <doi:10.1186/1471-2288-7-33>.
Eekhout (2017) <doi:10.1186/s12874-017-0404-7>.
Wiel (2009) <doi:10.1093/biostatistics/kxp011>.
Marshall (2009) <doi:10.1186/1471-2288-9-57>.
Sidi (2021) <doi:10.1080/00031305.2021.1898468>.
Lott (2018) <doi:10.1080/00031305.2018.1473796>.
Grund (2021) <doi:10.31234/osf.io/d459g>.

Encoding UTF-8

LazyData true

RoxygenNote 7.2.0

License GPL (>= 2)

URL https://mwheymans.github.io/miceafter/

BugReports https://github.com/mwheymans/miceafter/issues

VignetteBuilder knitr

Config/testthat/edition 3

NeedsCompilation no

Author Martijn Heymans [cre, aut] (<https://orcid.org/0000-0002-3889-0921>), Jaap Brand [ctb]

Maintainer Martijn Heymans <mw.heymans@amsterdamumc.nl>

Repository CRAN

Date/Publication 2022-10-02 13:30:02 UTC

R topics documented:

`bf_test` *Calculates the Brown-Forsythe test.*

Description

`bf_test` Calculates the Brown-Forsythe test for homogeneity of variance across groups, coefficients, variance-covariance matrix, and degrees of freedom.

Usage

```
bf_test(y, x, formula, data)
```

Arguments

`y`	numeric response variable.
`x`	categorical variable.
`formula`	A formula object to specify the model as normally used by glm. Use 'factor' to define the grouping variable.
`data`	An objects of class `milist`, created by `df2milist`, `list2milist` or `mids2milist`.

Details

The Levene's test centers around means to calculate outcome residuals, the Brown-Forsythe test around the median.

Value

An object containing:

- `fstats` F-test value, including numerator and denominator degrees of freedom.
- `qhat` pooled coefficients from fit.
- `vcov` variance-covariance matrix.
- `dfcom` degrees of freedom obtained from `df.residual`.

Author(s)

Martijn Heymans, 2021

See Also

with.milist

Examples

```
imp_dat <- df2milist(lbpmilr, impvar="Impnr")
ra <- with(imp_dat, expr=bf_test(Pain ~ factor(Carrying)))
```

---

cindex *Calculates the c-index and standard error*

Description

cindex Calculates the c-index and standard error for logistic and Cox regression models and the degrees of freedom to be further used in function with.milist.

Usage

```
cindex(formula, data)
```

Arguments

formula	A formula object to specify the model as normally used by glm or coxph.
data	An object of class milist, created by df2milist, list2milist or mids2milist.

Value

The c-index, related standard error and complete data degrees of freedom (dfcom) as n-1.

Author(s)

Martijn Heymans, 2021

See Also

with.milist, pool_cindex

Examples

```
imp_dat <- df2milist(lbpmilr, impvar="Impnr")
ra <- with(data=imp_dat,
expr = cindex(glm(Chronic ~ Gender + Radiation, family=binomial)))
```

cor2fz *Fisher z transformation of correlation coefficient*

Description

cor2fz Fisher z transformation of correlation coefficient

Usage

```
cor2fz(r)
```

Arguments

r	value for the correlation coefficient.

Value

correlation coefficient on z scale.

Author(s)

Martijn Heymans, 2022

Examples

```
cor2fz(r=0.65)
```

cor_est *Calculates the correlation coefficient*

Description

cor_est Calculates the correlation coefficient and standard error to be used in function with.miceafter.

Usage

```
cor_est(y, x, data, method = "pearson", se_method = "normal")
```

Arguments

y	name of numeric vector variable.
x	name of numeric vector variable.
data	An objects of class milist, created by df2milist, list2milist or mids2milist.
method	a character string indicating which correlation coefficient is used for the test. One of "pearson" (default), "kendall", or "spearman".
se_method	Method to calculate standard error. See details.

Details

The basic method to calculate the standard error is by:

$$se = \sqrt{(\frac{1}{n-3})}$$

For the Spearman correlation coefficients se_method "fieller" is calculated as:

$$se = \sqrt{(\frac{1.06}{n-3})}$$

For the Kendall correlation coefficients se_method "fieller" is calculated as:

$$se = \sqrt{(\frac{0.437}{n-4})}$$

Value

The correlation coefficient, standard error and complete data degrees of freedom (dfcom).

Author(s)

Martijn Heymans, 2022

See Also

`with.milist`, `pool_cor`

Examples

```
imp_dat <- df2milist(lbpmilr, impvar="Impnr")
ra <- with(imp_dat, expr=cor_est(y=BMI, x=Age))
```

`df2milist` *Turns a data frame with multiply imputed data into an object of class 'milist'*

Description

`df2milist` Turns a data frame of class 'data.frame', 'tbl_df' or 'tbl' (tibble) into an object of class 'milist' to be further used by 'miceafter::with'

Usage

```
df2milist(data, impvar, keep = FALSE)
```

Arguments

data	an object of class 'data.frame', 'tbl_df' or 'tbl' (tibble).
impvar	A character vector. Name of the variable that distinguishes the imputed datasets.
keep	if TRUE the grouping column is kept, if FALSE (default) the grouping column is not kept.

Value

an object of class 'milist' (Multiply Imputed Data list)

Author(s)

Martijn Heymans, 2021

---

f2chi *Converts F-values into Chi Square values*

Description

`f2chi` convert F to Chi-square values.

Usage

```
f2chi(f, df_num)
```

Arguments

f	a vector of F values.
df_num	single value for the numerator degrees of freedom of the F test.

Value

The Chi square values.

Author(s)

Martijn Heymans, 2021

Examples

```
f2chi(c(5.83, 4.95, 3.24, 6.27, 4.81), 5)
```

fz2cor *Fisher z back transformation of correlation coefficient*

Description

fz2cor Fisher z back transformation of correlation coefficient

Usage

```
fz2cor(z)
```

Arguments

z	value of the correlation coefficient on z scale.

Value

correlation coefficient on correlation scale.

Author(s)

Martijn Heymans, 2022

Examples

```
fz2cor(z=0.631)
```

glm_mi *Direct Pooling and model selection of Linear and Logistic regression models across multiply imputed data.*

Description

glm_mi Pooling and backward or forward selection of Linear and Logistic regression models across multiply imputed data using selection methods RR, D1, D2, D3, D4 and MPR (without use of with function).

Usage

```
glm_mi(
  data,
  formula = NULL,
  nimp = 5,
  impvar = NULL,
  keep.predictors = NULL,
  p.crit = 1,
  method = "RR",
  direction = NULL,
  model_type = NULL
)
```

Arguments

data	Data frame with stacked multiple imputed datasets. The original dataset that contains missing values must be excluded from the dataset. The imputed datasets must be distinguished by an imputation variable, specified under impvar, and starting by 1.
formula	A formula object to specify the model as normally used by glm. See under "Details" and "Examples" how these can be specified. If a formula object is used set predictors, cat.predictors, spline.predictors or int.predictors at the default value of NULL.
nimp	A numerical scalar. Number of imputed datasets. Default is 5.
impvar	A character vector. Name of the variable that distinguishes the imputed datasets.
keep.predictors	A single string or a vector of strings including the variables that are forced in the model during predictor selection. All type of variables are allowed.
p.crit	A numerical scalar. P-value selection criterium. A value of 1 provides the pooled model without selection.
method	A character vector to indicate the pooling method for p-values to pool the total model or used during predictor selection. This can be "RR", D1", "D2", "D3", "D4", or "MPR". See details for more information. Default is "RR".
direction	The direction of predictor selection, "BW" means backward selection and "FW" means forward selection.
model_type	A character vector for type of model, "binomial" is for logistic regression and "linear" is for linear regression models.

Details

The basic pooling procedure to derive pooled coefficients, standard errors, 95 confidence intervals and p-values is Rubin's Rules (RR). However, RR is only possible when the model includes continuous and dichotomous variables. Specific procedures are available when the model also included categorical (> 2 categories) or restricted cubic spline variables. These pooling methods are: "D1" is pooling of the total covariance matrix, "D2" is pooling of Chi-square values, "D3" and "D4" is

pooling Likelihood ratio statistics (method of Meng and Rubin) and "MPR" is pooling of median p-values (MPR rule). Spline regression coefficients are defined by using the rcs function for restricted cubic splines of the rms package. A minimum number of 3 knots as defined under knots is required.

A typical formula object has the form `Outcome ~ terms`. Categorical variables has to be defined as `Outcome ~ factor(variable)`, restricted cubic spline variables as `Outcome ~ rcs(variable, 3)`. Interaction terms can be defined as `Outcome ~ variable1*variable2` or `Outcome ~ variable1 + variable2 + variable1:variable2`. All variables in the terms part have to be separated by a "+". If a formula object is used set predictors, cat.predictors, spline.predictors or int.predictors at the default value of NULL.

Value

An object of class `pmods` (multiply imputed models) from which the following objects can be extracted:

- `data` imputed datasets
- `RR_model` pooled model at each selection step
- `RR_model_final` final selected pooled model
- `multiparm` pooled p-values at each step according to pooling method
- `multiparm_final` pooled p-values at final step according to pooling method
- `multiparm_out` (only when direction = "FW") pooled p-values of removed predictors
- `formula_step` formula object at each step
- `formula_final` formula object at final step
- `formula_initial` formula object at final step
- `predictors_in` predictors included at each selection step
- `predictors_out` predictors excluded at each step
- `impvar` name of variable used to distinguish imputed datasets
- `nimp` number of imputed datasets
- `Outcome` name of the outcome variable
- `method` selection method
- `p.crit` p-value selection criterium
- `call` function call
- `model_type` type of regression model used
- `direction` direction of predictor selection
- `predictors_final` names of predictors in final selection step
- `predictors_initial` names of predictors in start model
- `keep.predictors` names of predictors that were forced in the model

Author(s)

Martijn Heymans, 2021

References

Eekhout I, van de Wiel MA, Heymans MW. Methods for significance testing of categorical covariates in logistic regression models after multiple imputation: power and applicability analysis. BMC Med Res Methodol. 2017;17(1):129.

Enders CK (2010). Applied missing data analysis. New York: The Guilford Press.

Meng X-L, Rubin DB. Performing likelihood ratio tests with multiply-imputed data sets. Biometrika.1992;79:103-11.

van de Wiel MA, Berkhof J, van Wieringen WN. Testing the prediction error difference between 2 predictors. Biostatistics. 2009;10:550-60.

Marshall A, Altman DG, Holder RL, Royston P. Combining estimates of interest in prognostic modelling studies after multiple imputation: current practice and guidelines. BMC Med Res Methodol. 2009;9:57.

Van Buuren S. (2018). Flexible Imputation of Missing Data. 2nd Edition. Chapman & Hall/CRC Interdisciplinary Statistics. Boca Raton.

EW. Steyerberg (2019). Clinical Prediction MOdels. A Practical Approach to Development, Validation, and Updating (2nd edition). Springer Nature Switzerland AG.

http://missingdatasolutions.rbind.io/

Examples

```
pool_lr <- glm_mi(data=lbpmilr, formula = Chronic ~ Pain +
factor(Satisfaction) + rcs(Tampascale,3) + Radiation +
Radiation*factor(Satisfaction) + Age + Duration + BMI,
p.crit = 0.05, direction="FW", nimp=5, impvar="Impnr",
keep.predictors = c("Radiation*factor(Satisfaction)", "Age"),
method="D1", model_type="binomial")

pool_lr$RR_model_final
```

---

invlogit *Takes the inverse of a logit transformed value*

Description

`invlogit` Takes the inverse of a logit transformed value

Usage

```
invlogit(est)
```

Arguments

`est`	A parameter estimate on the logit scale.

Value

back transformed value.

Author(s)

Martijn Heymans, 2021

Examples

```
invlogit(est=1.39)
```

`invlogit_ci` *Takes the inverse of logit transformed parameters and calculates the confidence intervals*

Description

`invlogit_ci` Takes the inverse of logit transformed parameters and calculates the confidence interval by using the critical value.

Usage

```
invlogit_ci(est, se, crit.value)
```

Arguments

`est`	A parameter estimate on the logit scale.
`se`	A standard error value on the logit scale.
`crit.value`	Critical value of any distribution.

Details

Takes the inverse of logit transformed parameter estimates. The confidence interval is calculated by taking the inverse of $est +/- crit.value1 - \alpha/2 * se$.

Value

Parameter, critical value and confidence intervals on original scale.

Author(s)

Martijn Heymans, 2021

Examples

```
invlogit_ci(est=1.39, se=0.25, crit.value=1.96)
```

lbpmicox *Survival data of 265 Low Back Pain Patients*

Description

A data frame with 10 multiply imputed datasets of 265 observations each on 17 variables related to low back pain.

Usage

```
lbpmicox
```

Format

A data frame with 2650 observations on the following 18 variables.

Impnr a numeric vector

patnr a numeric vector

Status dichotomous event

Time continuous follow up time variable

Duration continuous

Previous dichotomous

Radiation dichotomous

Onset dichotomous

Age continuous

Tampascale continuous

Pain continuous

Function continuous

Satisfaction categorical

JobControl continuous

JobDemand continuous

Social continuous

Expectation a numeric vector

Expect_cat categorical

Examples

```
data(lbpmicox)
## maybe str(lbpmicox)
```

`lbpmilr` *Data of 159 Low Back Pain Patients*

Description

A data frame with 10 multiply imputed datasets of 159 observations each on 17 variables related to low back pain.

Usage

```
lbpmilr
```

Format

A data frame with 1590 observations on the following 17 variables.

Impnr a numeric vector

ID a numeric vector

Chronic dichotomous

Gender dichotomous

Carrying categorical

Pain continuous

Tampascale continuous

Function continuous

Radiation dichotomous

Age continuous

Smoking dichotomous

Satisfaction categorical

JobControl continuous

JobDemands continuous

SocialSupport continuous

Duration continuous

BMI continuous

Examples

```
data(lbpmilr)
## maybe str(lbpmilr)
```

lbp_orig *Dataset of 159 Low Back Pain Patients with missing values*

Description

A data frame with 159 observations of 15 variables related to low back pain.

Usage

```
lbp_orig
```

Format

A data frame with 159 observations on the following 15 variables.

Chronic dichotomous

Gender dichotomous

Carrying categorical

Pain continuous

Tampascale continuous

Function continuous

Radiation dichotomous

Age continuous

Smoking dichotomous

Satisfaction categorical

JobControl continuous

JobDemands continuous

SocialSupport continuous

Duration continuous

BMI continuous

Examples

```
data(lbp_orig)
## maybe str(lbp_orig)
```

`levene_test` *Calculates the Levene's test*

Description

`levene_test` Calculates the Levene's test for homogeneity of variance across groups, model coefficients, the variance-covariance matrix and the degrees of freedom.

Usage

```
levene_test(y, x, formula, data)
```

Arguments

`y`	numeric (continuous) response variable.
`x`	categorical group variable.
`formula`	A formula object to specify the model as normally used by glm. Use 'factor' to define the grouping x variable. Only one variable is allowed.
`data`	An objects of class `milist`, created by `df2milist`, `list2milist` or `mids2milist`.

Details

The Levene's test centers on group means to calculate outcome residuals, the Brown-Forsythe test on the median.

Value

An object from which the following objects are extracted:

- `fstats` F-test value, including numerator and denominator degrees of freedom.
- `qhat` model coefficients.
- `vcov` variance-covariance matrix.
- `dfcom` degrees of freedom obtained from `df.residual`.

Author(s)

Martijn Heymans, 2021

See Also

`with.milist`, `pool_levenetest`, `bf_test`

Examples

```
imp_dat <- df2milist(lbpmilr, impvar="Impnr")
ra <- with(imp_dat, expr=levene_test(Pain ~ factor(Carrying)))
```

list2milist *Turns a list object with multiply imputed datasets into an object of class 'milist'.*

Description

`list2milist` Turns a list with multiply imputed datasets into an object of class 'milist' to be further used by 'with.milist'

Usage

```
list2milist(data)
```

Arguments

`data`	an object of class 'list'.

Value

an object of class 'milist'

Author(s)

Martijn Heymans, 2021

logit_trans *Logit transformation of parameter estimates*

Description

`logit_trans` Logit transformation of parameter estimate and standard error.

Usage

```
logit_trans(est, se)
```

Arguments

`est`	A numeric vector of values.
`se`	A numeric vector of standard error values.

Details

Function is used to logit transform parameters and standard errors. For the standard error the Delta method is used.

Value

The logit transformed values.

Author(s)

Martijn Heymans, 2021

`mids2milist` *Turns a 'mice::mids' object into an object of class 'milist' to be further used by 'miceafter::with'*

Description

`mids2milist` Turns a 'mice::mids' object into an object with multiply imputed datasets of class 'milist' to be further used by 'miceafter::with'

Usage

```
mids2milist(data, keep = FALSE)
```

Arguments

`data`	a 'mice::mids' object
`keep`	if TRUE the grouping column is kept, if FALSE (default) the grouping column is not kept.

Value

an object of class 'milist'

Author(s)

Martijn Heymans, 2021

odds_ratio *Calculates the odds ratio (OR) and standard error.*

Description

odds_ratio Calculates the odds ratio and standard error and degrees of freedom to be used in function with.milist.

Usage

```
odds_ratio(y, x, formula, data)
```

Arguments

y	0-1 binary response variable.
x	0-1 binary independent variable.
formula	A formula object to specify the model as normally used by glm.
data	An objects of class milist, created by df2milist, df2milist or mids2milist.

Details

Note that the standard error of the OR is in fact the standard error of the (natural) log odds ratio.

Value

The odds ratio, related standard error and complete data degrees of freedom (dfcom) as n-2.

Author(s)

Martijn Heymans, 2021

See Also

with.milist, pool_odds_ratio

Examples

```
imp_dat <- df2milist(lbpmilr, impvar="Impnr")
ra <- with(imp_dat, expr=odds_ratio(Chronic ~ Radiation))
```

pool_bftest *Calculates the pooled Brown-Forsythe test.*

Description

pool_levenetest Calculates the pooled F-statistic of the Brown-Forsythe test.

Usage

```
pool_bftest(object, method = "D1")
```

Arguments

object	An object of class 'mistats' ('Multiply Imputed Statistical Analysis').
method	A character vector to choose the pooling method, 'D1' (default) or 'D2'.

Value

The (combined) F-statistic, p-value and degrees of freedom.

Author(s)

Martijn Heymans, 2021

References

Eekhout I, van de Wiel MA, Heymans MW. Methods for significance testing of categorical covariates in logistic regression models after multiple imputation: power and applicability analysis. BMC Med Res Methodol. 2017;17(1):129.

Enders CK (2010). Applied missing data analysis. New York: The Guilford Press.

Van Buuren S. (2018). Flexible Imputation of Missing Data. 2nd Edition. Chapman & Hall/CRC Interdisciplinary Statistics. Boca Raton.

See Also

with.milist, bf_test

Examples

```
imp_dat <- df2milist(lbpmilr, impvar="Impnr")
ra <- with(imp_dat, expr=bf_test(Pain ~ factor(Carrying)))
res <- pool_bftest(ra)
res
```

pool_cindex *Calculates the pooled C-index and Confidence intervals*

Description

pool_cindex Calculates the pooled C-index and Confidence intervals.

Usage

```
pool_cindex(data, conf.level = 0.95, dfcom = NULL)
```

Arguments

data	An object of class 'mistats' ('Multiply Imputed Statistical Analysis'.) or a m x 2 matrix with correlation coefficients and standard errors in the first and second column. For the latter option dfcom has to be provided.
conf.level	conf.level Confidence level of the confidence intervals.
dfcom	Number of completed-data analysis degrees of freedom. Default number is taken from function cindex

Details

Rubin's Rules are used for pooling. The C-index values are log transformed before pooling and finally back transformed.

Value

The pooled c-index value and the confidence intervals.

Vignettes

https://mwheymans.github.io/miceafter/articles/pooling_cindex.html

Author(s)

Martijn Heymans, 2021

See Also

with.milist, cindex

Examples

```
# Logistic Regression
imp_dat <- df2milist(lbpmilr, impvar="Impnr")
res_stats <- with(data=imp_dat,
 expr = cindex(glm(Chronic ~ Gender + Radiation,
 family=binomial)))
res <- pool_cindex(res_stats)
res

# Cox regression
library(survival)
imp_dat <- df2milist(lbpmicox, impvar="Impnr")
res_stats <- with(data=imp_dat,
  expr = cindex(coxph(Surv(Time, Status) ~ Pain + Radiation)))
res <- pool_cindex(res_stats)
res
```

`pool_cor` *Calculates the pooled correlation coefficient and Confidence intervals*

Description

`pool_cor` Calculates the pooled correlation coefficient and Confidence intervals.

Usage

```
pool_cor(
  data,
  conf.level = 0.95,
  dfcom = NULL,
  statistic = TRUE,
  df_small = TRUE,
  approxim = "tdistr"
)
```

Arguments

`data`	An object of class 'mistats' ('Multiply Imputed Statistical Analysis'.) or a m x 2 matrix with C-index values and standard errors in the first and second column. For the latter option dfcom has to be provided.
`conf.level`	conf.level Confidence level of the confidence intervals.
`dfcom`	Number of completed-data analysis degrees of freedom. Default number is taken from function `cindex`
`statistic`	if TRUE (default) the test statistic and p-value are provided, if FALSE these are not shown. See details.

df_small	if TRUE (default) the (Barnard & Rubin) small sample correction for the degrees of freedom is applied, if FALSE the old number of degrees of freedom is calculated.
approxim	if "tdistr" a t-distribution is used (default), if "zdistr" a z-distribution is used to derive a p-value for the test statistic.

Details

Rubin's Rules are used for pooling. The correlation coefficient is first transformed using Fisher z transformation (function cor2fz) before pooling and finally back transformed (function fz2cor). The test statistic and p-values are obtained using the Fisher z transformation.

Value

An object of class mipool from which the following objects can be extracted:

- cor correlation coefficient
- SE standard error
- t t-value (for confidence interval)
- low_r lower limit of confidence interval
- high_r upper limit of confidence interval
- statistic test statistic
- pval p-value

Author(s)

Martijn Heymans, 2022

See Also

with.milist, cor_est

Examples

```
imp_dat <- df2milist(lbpmilr, impvar="Impnr")
res_stats <- with(data=imp_dat,
 expr = cor_est(y=BMI, x=Age))
res <- pool_cor(res_stats)
res
```

pool_D2 *Combines the Chi Square statistics across Multiply Imputed datasets*

Description

pool_D2 The D2 statistic to combine the Chi square values across Multiply Imputed datasets.

Usage

```
pool_D2(dw, v)
```

Arguments

dw	a vector of chi square values obtained after multiple imputation.
v	single value for the degrees of freedom of the chi square statistic.

Value

The pooled chi square values as the D2 statistic, the p-value, the numerator, df1 and denominator, df2 degrees of freedom for the F-test.

Author(s)

Martijn Heymans, 2021

References

Eekhout I, van de Wiel MA, Heymans MW. Methods for significance testing of categorical covariates in logistic regression models after multiple imputation: power and applicability analysis. BMC Med Res Methodol. 2017;17(1):129.

Van Buuren S. (2018). Flexible Imputation of Missing Data. 2nd Edition. Chapman & Hall/CRC Interdisciplinary Statistics. Boca Raton.

Examples

```
pool_D2(c(2.25, 3.95, 6.24, 5.27, 2.81), 4)
```

pool_D4 *Pools the Likelihood Ratio tests across Multiply Imputed datasets (method D4)*

Description

pool_D4 The D4 statistic to combine the likelihood ratio tests (LRT) across Multiply Imputed datasets according method D4.

Usage

```
pool_D4(data, nimp, impvar, fm0, fm1, robust = TRUE, model_type = "binomial")
```

Arguments

data	Data frame with stacked multiple imputed datasets. The original dataset that contains missing values must be excluded from the dataset. The imputed datasets must be distinguished by an imputation variable, specified under impvar, and starting by 1.
nimp	A numerical scalar. Number of imputed datasets. Default is 5.
impvar	A character vector. Name of the variable that distinguishes the imputed datasets.
fm0	the null model.
fm1	the (nested) model to compare. Must be larger than the null model.
robust	if TRUE a robust LRT is used (algorithm 1 in Chan and Meng), otherwise algorithm 2 is used.
model_type	if TRUE (default) a logistic regression model is fitted, otherwise a linear regression model is used

Value

The D4 statistic, the numerator, df1 and denominator, df2 degrees of freedom for the F-test.

Author(s)

Martijn Heymans, 2021

References

Chan, K. W., & Meng, X.-L. (2019). Multiple improvements of multiple imputation likelihood ratio tests. https://arxiv.org/abs/1711.08822

Grund, Simon, Oliver Lüdtke, and Alexander Robitzsch. 2021. "Pooling Methods for Likelihood Ratio Tests in Multiply Imputed Data Sets." PsyArXiv. January 29. doi:10.31234/osf.io/d459g.

Examples

```
fm0 <- Chronic ~ BMI + factor(Carrying) +
  Satisfaction + SocialSupport + Smoking
fm1 <- Chronic ~ BMI + factor(Carrying) +
  Satisfaction +  SocialSupport + Smoking +
  Radiation

miceafter::pool_D4(data=lbpmilr, nimp=10, impvar="Impnr",
               fm0=fm0, fm1=fm1, robust = TRUE)
```

`pool_glm` *Pools and selects Linear and Logistic regression models across multiply imputed data.*

Description

`pool_glm` Pools and selects Linear and Logistic regression models across multiply imputed data, using pooling methods RR, D1, D2, D3, D4 and MPR (in combination with 'with' function).

Usage

```
pool_glm(
  object,
  method = "D1",
  p.crit = 1,
  keep.predictors = NULL,
  direction = NULL
)
```

Arguments

`object`	An object of class 'mistats' ('Multiply Imputed Statistical Analyses').
`method`	A character vector to indicate the multiparameter pooling method to pool the total model or used during model selection. This can be "RR", D1", "D2", "D3", "D4", or "MPR". See details for more information. Default is "RR".
`p.crit`	A numerical scalar. P-value selection criterium. A value of 1 provides the pooled model without selection.
`keep.predictors`	A single string or a vector of strings including the variables that are forced in the model during model selection. All type of variables are allowed.
`direction`	The direction for model selection, "BW" means backward selection and "FW" means forward selection.

Details

The basic pooling procedure to derive pooled coefficients, standard errors, 95 confidence intervals and p-values is Rubin's Rules (RR). However, RR is only possible when the model includes continuous and dichotomous variables. Multiparameter pooling methods are available when the model also included categorical (> 2 categories) variables. These pooling methods are: "D1" is pooling of the total covariance matrix, "D2" is pooling of Chi-square values, "D3" and "D4" is pooling Likelihood ratio statistics (method of Meng and Rubin) and "MPR" is pooling of median p-values (MPR rule). For pooling restricted cubic splines using the 'rcs' function of of the rms package, use function 'glm_mi'.

A typical formula object has the form `Outcome ~ terms`. Categorical variables has to be defined as `Outcome ~ factor(variable)`. Interaction terms can be defined as `Outcome ~ variable1*variable2` or `Outcome ~ variable1 + variable2 + variable1:variable2`. All variables in the terms part have to be separated by a "+".

Value

An object of class `mipool` (multiply imputed pooled models) from which the following objects can be extracted:

- `pmodel` pooled model (at last selection step)
- `pmultiparm` pooled p-values according to multiparameter test method (at last selection step)
- `pmodel_step` pooled model (at each selection step)
- `pmultiparm_step` pooled p-values according to multiparameter test method (at each selection step)
- `multiparm_final` pooled p-values at final step according to pooling method
- `multiparm_out` (only when direction = "FW") pooled p-values of removed predictors
- `formula_final` formula object at final step
- `formula_initial` formula object at final step
- `predictors_in` predictors included at each selection step
- `predictors_out` predictors excluded at each step
- `impvar` name of variable used to distinguish imputed datasets
- `nimp` number of imputed datasets
- `Outcome` name of the outcome variable
- `method` selection method
- `p.crit` p-value selection criterium
- `call` function call
- `model_type` type of regression model used
- `direction` direction of predictor selection
- `predictors_final` names of predictors in final selection step
- `predictors_initial` names of predictors in start model
- `keep.predictors` names of predictors that were forced in the model

Vignettes

https://mwheymans.github.io/miceafter/articles/regression_modelling.html

Author(s)

Martijn Heymans, 2021

References

Eekhout I, van de Wiel MA, Heymans MW. Methods for significance testing of categorical covariates in logistic regression models after multiple imputation: power and applicability analysis. BMC Med Res Methodol. 2017;17(1):129.

Enders CK (2010). Applied missing data analysis. New York: The Guilford Press.

Meng X-L, Rubin DB. Performing likelihood ratio tests with multiply-imputed data sets. Biometrika.1992;79:103-11.

van de Wiel MA, Berkhof J, van Wieringen WN. Testing the prediction error difference between 2 predictors. Biostatistics. 2009;10:550-60.

Marshall A, Altman DG, Holder RL, Royston P. Combining estimates of interest in prognostic modelling studies after multiple imputation: current practice and guidelines. BMC Med Res Methodol. 2009;9:57.

Van Buuren S. (2018). Flexible Imputation of Missing Data. 2nd Edition. Chapman & Hall/CRC Interdisciplinary Statistics. Boca Raton.

Examples

```
dat_list <- df2milist(lbpmilr, impvar="Impnr")
ra <- with(data=dat_list, expr = glm(Chronic ~ factor(Carrying) + Radiation + Age))
poolm <- pool_glm(ra, method="D1")
poolm$pmodel
poolm$pmultiparm
```

---

pool_levenetest *Calculates the pooled Levene test.*

Description

`pool_levenetest` Calculates the pooled F-statistic of the Levenene test.

Usage

```
pool_levenetest(object, method = "D1")
```

Arguments

object	An object of class 'mistats' ('Multiply Imputed Statistical Analysis').
method	A character vector to choose the pooling method, 'D1' (default) or 'D2'.

Value

The (combined) F-statistic, p-value and degrees of freedom.

Vignettes

https://mwheymans.github.io/miceafter/articles/levene_test.html

Author(s)

Martijn Heymans, 2021

References

Eekhout I, van de Wiel MA, Heymans MW. Methods for significance testing of categorical covariates in logistic regression models after multiple imputation: power and applicability analysis. BMC Med Res Methodol. 2017;17(1):129.

Enders CK (2010). Applied missing data analysis. New York: The Guilford Press.

Van Buuren S. (2018). Flexible Imputation of Missing Data. 2nd Edition. Chapman & Hall/CRC Interdisciplinary Statistics. Boca Raton.

See Also

with.milist, levene_test

Examples

```
library(magrittr)
lbpmilr %>%
   df2milist(impvar="Impnr") %>%
      with(expr=levene_test(Pain ~ factor(Carrying))) %>%
        pool_levenetest(method="D1")

# Same as
imp_dat <- df2milist(lbpmilr, impvar="Impnr")
ra <- with(imp_dat, expr=levene_test(Pain ~ factor(Carrying)))
res <- pool_levenetest(ra, method="D1")
```

pool_odds_ratio *Calculates the pooled odds ratio (OR) and related confidence interval.*

Description

pool_odds_ratio Calculates the pooled odds ratio and confidence interval.

Usage

```
pool_odds_ratio(object, conf.level = 0.95, dfcom = NULL)
```

Arguments

object	An object of class 'mistats' ('Multiply Imputed Statistical Analysis')
conf.level	Confidence level of the confidence intervals.
dfcom	Complete data degrees of freedom. Default number is taken from function odds_ratio

Value

The pooled OR and confidence intervals.

Author(s)

Martijn Heymans, 2021

See Also

with.milist, odds_ratio

Examples

```
library(magrittr)
lbpmilr %>%
   df2milist(impvar="Impnr") %>%
     with(expr=odds_ratio(Chronic ~ Radiation)) %>%
       pool_odds_ratio()

# Same as
imp_dat <- df2milist(lbpmilr, impvar="Impnr")
ra <- with(imp_dat, expr=odds_ratio(Chronic ~ Radiation))
res <- pool_odds_ratio(ra)
```

pool_propdiff_ac *Calculates the pooled difference between proportions and standard error according to Agresti-Caffo across multiply imputed datasets.*

Description

pool_propdiff_ac Calculates the pooled difference between proportions and standard error according to Agresti-Caffo across multiply imputed datasets.

Usage

```
pool_propdiff_ac(object, conf.level = 0.95, dfcom = NULL)
```

Arguments

object	An object of class 'mistats' ('Multiply Imputed Statistical Analysis').
conf.level	Confidence level of the confidence intervals.
dfcom	Complete data degrees of freedom. Default number is taken from function propdiff_ac

Details

For the pooled difference between proportions the difference between proportions according to Wald are used. The Agresti-Caffo difference is used to derive the Agresti-Caffo confidence intervals.

Value

The proportion, the Confidence intervals, the standard error and statistic.

Author(s)

Martijn Heymans, 2021

References

Agresti, A. and Caffo, B. Simple and Effective Confidence Intervals for Proportions and Differences of Proportions Result from Adding Two Successes and Two Failures. The American Statistician. 2000;54:280-288.

Fagerland MW, Lydersen S, Laake P. Recommended confidence intervals for two independent binomial proportions. Stat Methods Med Res. 2015 Apr;24(2):224-54.

See Also

with.milist, propdiff_ac

Examples

```
imp_dat <- df2milist(lbpmilr, impvar="Impnr")
ra <- with(imp_dat, expr=propdiff_ac(Chronic ~ Radiation))
res <- pool_propdiff_ac(ra)
res
```

`pool_propdiff_nw` *Calculates the pooled difference between proportions and confidence intervals according to Newcombe-Wilson (NW) across multiply imputed datasets.*

Description

`pool_propdiff_nw` Calculates the pooled difference between proportions and confidence intervals according to Newcombe-Wilson (NW) across multiply imputed datasets.

Usage

```
pool_propdiff_nw(object, conf.level = 0.95)
```

Arguments

`object`	An object of class 'mistats' ('Multiply Imputed Statistical Analysis'.).
`conf.level`	Confidence level of the confidence intervals. Mostly set at 0.95.

Details

The `pool_propdiff_nw` function uses information from separate exposure groups. It is therefore important to first use the `propdiff_wald` function and to set strata = TRUE in that function.

Value

The Proportion and the Confidence intervals according to Newcombe-Wilson.

Author(s)

Martijn Heymans, 2021

References

Yulia Sidi & Ofer Harel (2021): Difference Between Binomial Proportions Using Newcombe's Method With Multiple Imputation for Incomplete Data, The American Statistician, DOI:10.1080/00031305.2021.1898468

See Also

`with.milist`, `propdiff_wald`

Examples

```
library(magrittr)
lbpmilr %>%
  df2milist(impvar="Impnr") %>%
    with(expr=propdiff_wald(Chronic ~ Radiation, strata = TRUE)) %>%
      pool_propdiff_nw()

# Same as
imp_dat <- df2milist(lbpmilr, impvar="Impnr")
res <- with(imp_dat, expr=propdiff_wald(Chronic ~ Radiation, strata = TRUE))
res <- pool_propdiff_nw(res)
```

`pool_propdiff_wald` *Calculates the pooled difference between proportions and standard error according to Wald across multiply imputed datasets.*

Description

`pool_propdiff_wald` Calculates the pooled difference between proportions and standard error according to Wald across multiply imputed datasets.

Usage

```
pool_propdiff_wald(object, conf.level = 0.95, dfcom = NULL)
```

Arguments

`object`	An object of class 'mistats' ('Multiply Imputed Statistical Analysis').
`conf.level`	Confidence level of the confidence intervals.
`dfcom`	Complete data degrees of freedom. Default number is taken from function `propdiff_wald`

Value

The proportion, the Confidence intervals, the standard error and statistic.

Author(s)

Martijn Heymans, 2021

See Also

`with.milist`, `propdiff_wald`

Examples

```
imp_dat <- df2milist(lbpmilr, impvar="Impnr")
ra <- with(imp_dat, expr=propdiff_wald(Chronic ~ Gender))
res <- pool_propdiff_wald(ra)
res
```

`pool_prop_nna` *Calculates the pooled proportion and confidence intervals using an approximate Beta distribution.*

Description

`pool_prop_nna` Calculates the pooled proportion and confidence intervals using an approximate Beta distribution.

Usage

```
pool_prop_nna(object, conf.level = 0.95)
```

Arguments

`object`	An object of class 'mistats' ('Multiply Imputed Statistical Analysis').
`conf.level`	Confidence level of the confidence intervals.

Details

The parameters for the Beta distribution are calculated using the method of moments (Gelman et al. p. 582).

Value

The pooled proportion and the 95% Confidence interval.

Author(s)

Martijn Heymans, 2021

References

Raghunathan, T. (2016). Missing Data Analysis in Practice. Boca Raton, FL: Chapman and Hall/CRC. (paragr 4.6.2)

Andrew Gelman, John B. Carlin, Hal S. Stern, David B. Dunson, Aki Vehtari, Donald B. Rubin. (2003). Bayesian Data Analysis (2nd ed). Chapman and Hall/CRC.

See Also

with.milist, prop_nna

Examples

```
imp_dat <- df2milist(lbpmilr, impvar='Impnr')
ra <- with(imp_dat, expr=prop_nna(Radiation))
res <- pool_prop_nna(ra)
res
```

---

pool_prop_wald *Calculates the pooled proportion and standard error according to Wald across multiply imputed datasets.*

Description

pool_prop_wald Calculates the pooled proportion and standard error according to Wald across multiply imputed datasets and using Rubin's Rules.

Usage

```
pool_prop_wald(object, conf.level = 0.95, dfcom = NULL)
```

Arguments

object	An object of class 'mistats' (repeated statistical analysis across multiply imputed datasets).
conf.level	Confidence level of the confidence intervals.
dfcom	Complete data degrees of freedom. Default number is taken from function prop_wald

Details

Before pooling, the proportions will be naturally log transformed and the pooled estimates back transformed to the original scale.

Value

The proportion, the Confidence intervals, the standard error and the statistic.

Author(s)

Martijn Heymans, 2021

See Also

with.milist, prop_wald

Examples

```
imp_dat <- df2milist(lbpmilr, impvar="Impnr")
ra <- with(imp_dat, expr=prop_wald(Radiation ~ 1))
res <- pool_prop_wald(ra)
res
```

---

pool_prop_wilson *Calculates the pooled single proportion confidence intervals according to Wilson across multiply imputed datasets.*

Description

pool_prop_wilson Calculates the pooled single proportion and confidence intervals according to Wald across multiply imputed datasets.

Usage

```
pool_prop_wilson(object, conf.level = 0.95)
```

Arguments

object	An object of class 'mistats' ('Multiply Imputed Statistical Analysis').
conf.level	Confidence level of the confidence intervals.

Value

The proportion and the 95% Confidence interval according to Wilson.

Author(s)

Martijn Heymans, 2021

References

Anne Lott & Jerome P. Reiter (2020) Wilson Confidence Intervals for Binomial Proportions With Multiple Imputation for Missing Data, The American Statistician, 74:2, 109-115, DOI: 10.1080/00031305.2018.1473796.

See Also

with.milist, prop_wald

Examples

```
library(magrittr)
lbpmilr %>%
  df2milist(impvar="Impnr") %>%
    with(expr=prop_wald(Radiation ~ 1)) %>%
      pool_prop_wilson()

# Same as
imp_dat <- df2milist(lbpmilr, impvar="Impnr")
ra <- with(imp_dat, expr=prop_wald(Radiation ~ 1))
res <- pool_prop_wilson(ra)
```

`pool_risk_ratio` *Calculates the pooled risk ratio (RR) and related confidence interval.*

Description

`pool_risk_ratio` Calculates the pooled risk ratio and confidence interval.

Usage

```
pool_risk_ratio(object, conf.level = 0.95, dfcom = NULL)
```

Arguments

`object`	An object of class 'mistats' ('Multiply Imputed Statistical Analysis').
`conf.level`	Confidence level of the confidence intervals.
`dfcom`	Complete data degrees of freedom. Default number is taken from function `risk_ratio`

Value

The pooled RR and confidence intervals.

Author(s)

Martijn Heymans, 2021

See Also

`with.milist`, `risk_ratio`

Examples

```
library(magrittr)
lbpmilr %>%
 df2milist(impvar="Impnr") %>%
   with(expr=risk_ratio(Chronic ~ Radiation)) %>%
     pool_risk_ratio()

# Same as
imp_dat <- df2milist(lbpmilr, impvar="Impnr")
ra <- with(imp_dat, expr=risk_ratio(Chronic ~ Radiation))
res <- pool_risk_ratio(ra)
```

`pool_scalar_RR` *Rubin's Rules for scalar estimates*

Description

`pool_scalar_RR` Applies Rubin's pooling Rules for scalar estimates

Usage

```
pool_scalar_RR(
  est,
  se,
  logit_trans = FALSE,
  conf.level = 0.95,
  statistic = FALSE,
  dfcom = NULL,
  df_small = TRUE,
  approxim = "tdistr"
)
```

Arguments

`est`	a numerical vector of parameter estimates.
`se`	a numerical vector of standard error estimates.
`logit_trans`	If TRUE logit transformation of parameter values is applied before pooling, if FALSE (default), pooling is done on the original parameter scale.
`conf.level`	Confidence level of the confidence intervals.
`statistic`	if TRUE the test statistic and confidence interval are provided, if FALSE (default) these are not shown.
`dfcom`	The complete data analysis degrees of freedom.

df_small	if TRUE (default) the (Barnard & Rubin) small sample correction for the degrees of freedom is applied, if FALSE the old number of degrees of freedom is calculated.
approxim	if "tdistr" a t-distribution is used (default), if "zdistr" a z-distribution is used to derive a p-value according to the test statistic.

Details

The t-value is the quantile value of the t-distribution that can be used to calculate confidence intervals according to $est_{pooled} + / - t_{1-\alpha/2} * se_{pooled}$. When statistic is TRUE the test statistic is calculated as $statistic = estpooled/sepooled$. The p-value is than derived using the t-distribution and adjusted degrees of freedom.

Value

A list object from which the following objects are extracted:

- pool_est the pooled parameter value.
- pool_se the pooled standard error value.
- t quantile of the t-distribution (to calculate confidence intervals).
- r the relative increase in variance due to missing data.
- dfcom complete data degrees of freedom.
- v_adj adjusted degrees of freedom (according to Barnard and Rubin 1999)

Author(s)

Martijn Heymans, 2021

Examples

```
est <- c(0.4, 0.6, 0.8)
se <- c(0.02, 0.05, 0.03)
res <- pool_scalar_RR(est, se, dfcom=500)
res
```

---

pool_t_test *Calculates the pooled t-test and Confidence intervals*

Description

pool_t_test Calculates the pooled t-test, confidence intervals and p-value.

Usage

```
pool_t_test(object, conf.level = 0.95, dfcom = NULL, statistic = FALSE)
```

Arguments

object	An object of class 'mistats' ('Multiply Imputed Statistical Analysis'.)
conf.level	conf.level Confidence level of the confidence intervals.
dfcom	Number of completed-data analysis degrees of freedom. Default number is taken from function cindex.
statistic	if TRUE (default) the test statistic and p-value are provided, if FALSE these are not shown.

Value

An object of class mipool from which the following objects can be extracted:

- Mean diff Difference between means
- SE standard error
- t t-value (for confidence interval)
- low_r lower limit of confidence interval
- high_r upper limit of confidence interval
- statistic test statistic
- pval p-value

Author(s)

Martijn Heymans, 2022

See Also

with.milist, t_test

Examples

```
imp_dat <- df2milist(lbpmilr, impvar="Impnr")
res_stats <- with(data=imp_dat,
 expr = t_test(Pain ~ Gender, var_equal=TRUE, paired=FALSE))
res <- pool_t_test(res_stats)
res
```

propdiff_ac *Calculates the difference between proportions and standard error according to method Agresti-Caffo*

Description

propdiff_ac Calculates the difference between proportions and standard error according to method Agresti-Caffo.

Usage

```
propdiff_ac(y, x, formula, data)
```

Arguments

y	0-1 binary response variable.
x	0-1 binary independent variable.
formula	A formula object to specify the model as normally used by glm.
data	An objects of class milist, created by df2milist, list2milist or mids2milist.

Details

As output the differences between proportions according to Agresti-Caffo and Wald are provided. The Agresti-Caffo difference is used in the function pool_propdiff_ac to derive the Agresti-Caffo confidence intervals. For the pooled difference between proportions the difference between proportions according to Wald are used.

Value

The difference between proportions, the standard error according to Agresti-Caffo and complete data degrees of freedom (dfcom) as n-1.

Author(s)

Martijn Heymans, 2021

References

Agresti, A. and Caffo, B. Simple and Effective Confidence Intervals for Proportions and Differences of Proportions Result from Adding Two Successes and Two Failures. The American Statistician. 2000;54:280-288.

Fagerland MW, Lydersen S, Laake P. Recommended confidence intervals for two independent binomial proportions. Stat Methods Med Res. 2015 Apr;24(2):224-54.

See Also

with.milist, pool_propdiff_ac

Examples

```
imp_dat <- df2milist(lbpmilr, impvar="Impnr")
ra <- with(imp_dat, expr=propdiff_ac(Chronic ~ Radiation))

# same as
ra <- with(imp_dat, expr=propdiff_ac(y=Chronic, x=Radiation))
```

`propdiff_wald` *Calculates the difference between proportions and standard error according to Wald*

Description

`propdiff_wald` Calculates the difference between proportions and standard error according to Wald and degrees of freedom to be used in function `with.miceafter`.

Usage

```
propdiff_wald(y, x, formula, data, strata = FALSE)
```

Arguments

`y`	0-1 binary response variable.
`x`	0-1 binary independent variable.
`formula`	A formula object to specify the model as normally used by glm.
`data`	An objects of class `milist`, created by `df2milist`, `list2milist` or `mids2milist`.
`strata`	If TRUE the proportion, se and n of each group is provided. Default is FALSE. Has to be used in combination with function `pool_propdiff_wilson`

Value

The difference between proportions, standard error and complete data degrees of freedom (dfcom) as n-1.

Author(s)

Martijn Heymans, 2021

See Also

`with.milist`, `pool_propdiff_nw`

Examples

```
imp_dat <- df2milist(lbpmilr, impvar="Impnr")
ra <- with(imp_dat, expr=propdiff_wald(Chronic ~ Radiation))

# proportions in each subgroup
imp_dat <- df2milist(lbpmilr, impvar="Impnr")
ra <- with(imp_dat, expr=propdiff_wald(Chronic ~ Radiation, strata=TRUE))
```

---

prop_nna *Calculates the posterior beta components for a single proportion*

Description

prop_nna Calculates the posterior beta components for a single proportion (assuming noninformative prior).

Usage

```
prop_nna(x, data)
```

Arguments

x	name of variable to calculate proportion.
data	An object of class 'mistats' ('Multiply Imputed Statistical Analysis').

Value

The posterior beta components.

Author(s)

Martijn Heymans, 2021

References

Raghunathan, T. (2016). Missing Data Analysis in Practice. Boca Raton, FL: Chapman and Hall/CRC. (paragr 4.6.2)

See Also

with.milist, pool_prop_nna

Examples

```
imp_dat <- df2milist(lbpmilr, impvar='Impnr')
ra <- with(imp_dat, expr=prop_nna(Radiation))
```

prop_wald *Calculates a single proportion and related standard error according to Wald*

Description

prop_wald Calculates a single proportion and related standard error according to Wald and provides degrees of freedom to be used in function with.miceafter.

Usage

```
prop_wald(x, formula, data)
```

Arguments

x	name of variable to calculate proportion.
formula	A formula object to specify the model as normally used by glm.
data	An objects of class milist, created by df2milist, list2milist or mids2milist.

Value

The proportion, standard error and complete data degrees of freedom (dfcom) as n-1.

Author(s)

Martijn Heymans, 2021

See Also

with.milist, pool_prop_wald

Examples

```
imp_dat <- df2milist(lbpmilr, impvar="Impnr")
ra <- with(imp_dat, expr=prop_wald(Chronic ~ 1))
```

`risk_ratio` *Calculates the risk ratio (RR) and standard error.*

Description

`risk_ratio` Calculates the risk ratio and standard error.

Usage

```
risk_ratio(y, x, formula, data)
```

Arguments

`y`	0-1 binary response variable.
`x`	0-1 binary independent variable.
`formula`	A formula object to specify the model as normally used by glm.
`data`	An objects of class `milist`, created by `df2milist`, `list2milist` or `mids2milist`.

Details

Note that the standard error of the RR is in fact the standard error of the (natural) risk ratio.

Value

The risk ratio, related standard error and complete data degrees of freedom (dfcom) as n-2.

Author(s)

Martijn Heymans, 2021

See Also

`with.milist`

Examples

```
imp_dat <- df2milist(lbpmilr, impvar="Impnr")
ra <- with(imp_dat, expr=risk_ratio(Chronic ~ Radiation))
```

t_test *Calculates the one, two and paired sample t-test*

Description

t_test Calculates the one, two and paired sample t-test.

Usage

```
t_test(y, x, formula, data, paired = FALSE, var_equal = TRUE)
```

Arguments

y	numeric response variable.
x	categorical variable with 2 groups.
formula	A formula object to specify the model as normally used by glm.
data	An objects of class milist, created by df2milist, list2milist or mids2milist.
paired	a logical indicating whether you want a paired t-test (TRUE) or not (FALSE, default).
var_equal	a logical, if TRUE equal variances are assumed, if FALSE (default) equal variances are not assumed and Welch correction is applied for the number of degrees of freedom. See detail.

Details

For all t-tests the dataset must be in long format (i.e. group data under each other). For the paired t-test x and y must have the same length. When variances between groups are unequal, the Welch df correction formula is used and eventually averaged across multiply imputed datasets in the pool_t_test function.

Value

An object containing the following objects are extracted:

- mdiff the mean difference.
- se the standard error.
- dfcom the complete data degrees of freedom.

Author(s)

Martijn Heymans, 2022

See Also

with.milist, pool_t_test

Examples

```
imp_dat <- df2milist(lbpmilr, impvar="Impnr")
ra <- with(imp_dat, expr=t_test(Pain ~ Gender))
```

`with.milist` *Evaluate an Expression across a list of multiply imputed datasets*

Description

`with.milist` Evaluate an expression in the form of a statistical test procedure across a list of multiply imputed datasets

Usage

```
## S3 method for class 'milist'
with(data, expr = NULL, ...)
```

Arguments

`data`	data that is used to evaluate the expression in, an objects of class `milist` after a call to function `df2milist`, `list2milist` or `mids2milist`. For 'df2milist' the original dataset (normally indicated as dataset 0) must be exluded and the imputed datasets must be distinguished by an imputation variable, specified under impvar and starting by 1.
`expr`	expression to evaluate.
`...`	Not required.

Value

The value of the evaluated expression with class `mistats` 'Multiply Imputed Statistical Analysis'.

Author(s)

Martijn Heymans, 2021

Index