Package: pooling (via r-universe)

September 16, 2024

Type Package

Title Fit Poolwise Regression Models

Version 1.1.2

Date 2020-02-12

Author Dane R. Van Domelen

Maintainer Dane R. Van Domelen <vandomed@gmail.com>

Description Functions for calculating power and fitting regression models in studies where a biomarker is measured in ``pooled'' samples rather than for each individual. Approaches for handling measurement error follow the framework of Schisterman et al. (2010) <doi:10.1002/sim.3823>.

License GPL-3

LazyData true

Encoding UTF-8

RoxygenNote 6.1.1

Imports cubature, data.table, dplyr, dvmisc, ggplot2, ggrepel, mvtnorm, numDeriv, stats

Repository https://vandomed.r-universe.dev

RemoteUrl https://github.com/vandomed/pooling

RemoteRef HEAD

RemoteSha 45132212f4a13fd21c62ef6405f8d773ab2f7414

Contents

`cond_logreg` *Conditional Logistic Regression with Measurement Error in One Covariate*

Description

Compatible with individual or pooled measurements. Assumes a normal linear model for exposure given other covariates, and additive normal errors.

Usage

```
cond_logreg(g = rep(1, length(xtilde1)), xtilde1, xtilde0, c1 = NULL,
  c0 = NULL, errors = "processing", approx_integral = TRUE,
  estimate_var = FALSE, start_nonvar_var = c(0.01, 1),
  lower_nonvar_var = c(-Inf, 1e-04), upper_nonvar_var = c(Inf, Inf),
  jitter_start = 0.01, hcubature_list = list(tol = 1e-08),
  nlminb_list = list(control = list(trace = 1, eval.max = 500, iter.max =
  500)), hessian_list = list(method.args = list(r = 4)),
  nlminb_object = NULL)
```

Arguments

g	Numeric vector with pool sizes, i.e. number of members in each pool.
xtilde1	Numeric vector (or list of numeric vectors, if some observations have replicates) with Xtilde values for cases.
xtilde0	Numeric vector (or list of numeric vectors, if some observations have replicates) with Xtilde values for controls.
c1	Numeric matrix with precisely measured covariates for cases.
c0	Numeric matrix with precisely measured covariates for controls.
errors	Character string specifying the errors that X is subject to. Choices are "none", "measurement" for measurement error, "processing" for processing error (only relevant for pooled data), and "both".
approx_integral	Logical value for whether to use the probit approximation for the logistic-normal integral, to avoid numerically integrating X's out of the likelihood function.
estimate_var	Logical value for whether to return variance-covariance matrix for parameter estimates.
start_nonvar_var	Numeric vector of length 2 specifying starting value for non-variance terms and variance terms, respectively.
lower_nonvar_var	Numeric vector of length 2 specifying lower bound for non-variance terms and variance terms, respectively.
upper_nonvar_var	Numeric vector of length 2 specifying upper bound for non-variance terms and variance terms, respectively.
jitter_start	Numeric value specifying standard deviation for mean-0 normal jitters to add to starting values for a second try at maximizing the log-likelihood, should the initial call to nlminb result in non-convergence. Set to NULL for no second try.
hcubature_list	List of arguments to pass to hcubature for numerical integration. Only used if approx_integral = FALSE.
nlminb_list	List of arguments to pass to nlminb for log-likelihood maximization.
hessian_list	List of arguments to pass to hessian for approximating the Hessian matrix. Only used if estimate_var = TRUE.
nlminb_object	Object returned from nlminb in a prior call. Useful for bypassing log-likelihood maximization if you just want to re-estimate the Hessian matrix with different options.

Value

List containing:

1. Numeric vector of parameter estimates.
2. Variance-covariance matrix (if estimate_var = TRUE).
3. Returned nlminb object from maximizing the log-likelihood function.
4. Akaike information criterion (AIC).

References

Saha-Chaudhuri, P., Umbach, D.M. and Weinberg, C.R. (2011) "Pooled exposure assessment for matched case-control studies." *Epidemiology* **22**(5): 704–712.

Schisterman, E.F., Vexler, A., Mumford, S.L. and Perkins, N.J. (2010) "Hybrid pooled-unpooled design for cost-efficient measurement of biomarkers." *Stat. Med.* **29**(5): 597–613.

Weinberg, C.R. and Umbach, D.M. (1999) "Using pooled exposure assessment to improve efficiency in case-control studies." *Biometrics* **55**: 718–726.

Weinberg, C.R. and Umbach, D.M. (2014) "Correction to 'Using pooled exposure assessment to improve efficiency in case-control studies' by Clarice R. Weinberg and David M. Umbach; 55, 718–726, September 1999." *Biometrics* **70**: 1061.

Examples

```
# Load simulated data for 150 case pools and 150 control pools
data(dat_cond_logreg)
dat <- dat_cond_logreg$dat
xtilde1 <- dat_cond_logreg$xtilde1
xtilde0 <- dat_cond_logreg$xtilde0

# Fit conditional logistic regression to estimate log-odds ratio for X and Y
# adjusted for C, using the precise poolwise summed exposure X. True log-OR
# for X is 0.5.
truth <- cond_logreg(
  g = dat$g,
  xtilde1 = dat$x1,
  xtilde0 = dat$x0,
  c1 = dat$c1.model,
  c0 = dat$c0.model,
  errors = "neither"
)
truth$theta.hat

# Suppose X is subject to additive measurement error and processing error,
# and we observe Xtilde1 and Xtilde0 rather than X1 and X0. Fit model with
# Xtilde's, accounting for errors (numerical integration avoided by using
# probit approximation).
## Not run:
corrected <- cond_logreg(
  g = dat$g,
  xtilde1 = xtilde1,
  xtilde0 = xtilde0,
  c1 = dat$c1.model,
  c0 = dat$c0.model,
  errors = "both",
  approx_integral = TRUE
)
corrected$theta.hat

## End(Not run)
```

dat_cond_logreg *Dataset for Examples in cond_logreg*

Description

List containing (1) data frame with poolwise (g, X1, X0, C1.model, C0.model, C1.match, C0.match) values, (2) list of replicate Xtilde values for case pools, and (3) list of replicate Xtilde values for control pools.

Source

Simulated data in R.

dat_p_gdfa *Dataset for Examples in p_gdfa*

Description

List containing (1) data frame with poolwise (g, Y, X, Xtilde) values, (2) list with replicate Xtilde values, and (3) list with C values for members of each pool.

Source

Simulated data in R.

dat_p_linreg_yerrors *Dataset for Examples in p_linreg_yerrors*

Description

List containing (1) data frame with poolwise (g, Y, X1, X2) values and (2) list with replicate Y values.

Source

Simulated data in R.

dat_p_ndfa *Dataset for Examples in p_ndfa*

Description

List containing (1) data frame with poolwise (g, Y*, Y, X, Xtilde, C) values and (2) list with replicate Xtilde values.

Source

Simulated data in R.

form_pools *Created a Pooled Dataset from a Subject-Specific One*

Description

Useful for simulation studies on biospecimen pooling designs.

Usage

```
form_pools(dat, pool_sizes, num_each = NULL,
  prop_each = rep(1/length(pool_sizes), length(pool_sizes)))
```

Arguments

dat	Data frame with individual level data.
pool_sizes	Integer vector of pool sizes ordered from largest to smallest.
num_each	Integer vector specifying number of pools of each size.
prop_each	Numeric vector specifying proportion of pools of each size.

Value

Data frame.

pdat1 *Dataset for Examples in p_dfa_xerrors and p_logreg_xerrors*

Description

Data frame with poolwise (g, Y*, Y, Xtilde, C) values.

Source

Simulated data in R.

pdat2 — *Dataset for Examples in p_dfa_xerrors2 and p_logreg_xerrors2*

Description

List containing (1) data frame with poolwise (g, Y, Xtilde, C) values and (2) list of C values for members of each pool.

Source

Simulated data in R.

plot_dfa — *Plot Log-OR vs. X for Normal Discriminant Function Approach*

Description

Archived on 7/23/2018. Please use `plot_ndfa` instead.

Usage

```
plot_dfa(estimates, varcov = NULL, xrange, xname = "X", cvals = NULL,
  set_labels = NULL, set_panels = TRUE)
```

Arguments

`estimates`	Numeric vector of point estimates for `(gamma_0, gamma_y, gamma_c^T, sigsq)`.
`varcov`	Numeric matrix with variance-covariance matrix for `estimates`. If `NULL`, 95% confidence bands are omitted.
`xrange`	Numeric vector specifying range of X values to plot.
`xname`	Character vector specifying name of X variable, for plot title and x-axis label.
`cvals`	Numeric vector or list of numeric vectors specifying covariate values to use in log-odds ratio calculations.
`set_labels`	Character vector of labels for the sets of covariate values. Only used if `cvals` is a list.
`set_panels`	Logical value for whether to use separate panels for each set of covariate values, as opposed to using different colors on a single plot.

Value

Plot of log-OR vs. `X` generated by `ggplot`.

Examples

```
# Fit discriminant function model for poolwise Xtilde vs. (Y, C), without
# assuming a constant log-OR. Ignoring processing errors for simplicity.
data(pdat1)
fit <- p_dfa_xerrors(g = pdat1$g, y = pdat1$numcases, xtilde = pdat1$xtilde,
                     c = pdat1$c, errors = "neither", constant_or = FALSE)

# Plot estimated log-OR vs. X at mean value for C
p <- plot_dfa(estimates = fit$estimates, varcov = fit$theta.var,
              xrange = range(pdat1$xtilde / pdat1$g),
              cvals = mean(pdat1$c / pdat1$g))
p
```

`plot_dfa2` *Plot Log-OR vs. X for Gamma Discriminant Function Approach*

Description

Archived on 7/23/2018. Please use `plot_gdfa` instead.

Usage

```
plot_dfa2(estimates, varcov = NULL, xrange, xname = "X",
  cvals = NULL, set_labels = NULL, set_panels = TRUE)
```

Arguments

`estimates`	Numeric vector of point estimates for `(gamma_0, gamma_y, gamma_c^T, b1, b0)`.
`varcov`	Numeric matrix with variance-covariance matrix for `estimates`. If `NULL`, 95% confidence bands are omitted.
`xrange`	Numeric vector specifying range of X values to plot.
`xname`	Character vector specifying name of X variable, for plot title and x-axis label.
`cvals`	Numeric vector or list of numeric vectors specifying covariate values to use in log-odds ratio calculations.
`set_labels`	Character vector of labels for the sets of covariate values. Only used if `cvals` is a list.
`set_panels`	Logical value for whether to use separate panels for each set of covariate values, as opposed to using different colors on a single plot.

Value

Plot of log-OR vs. `X` generated by `ggplot`.

Examples

```
# Fit Gamma discriminant function model for poolwise Xtilde vs. (Y, C),
# without assuming a constant log-OR. Ignoring processing errors for simplicity.
data(pdat2)
dat <- pdat2$dat
c.list <- pdat2$c.list
fit <- p_dfa_xerrors2(
  g = dat$g,
  y = dat$y,
  xtilde = dat$xtilde,
  c = c.list,
  errors = "neither",
  constant_or = FALSE
)

# Plot estimated log-OR vs. X at mean value for C
p <- plot_dfa2(
  estimates = fit$estimates,
  varcov = fit$theta.var,
  xrange = range(dat$xtilde / dat$g),
  cvals = mean(unlist(c.list))
)
p
```

`plot_gdfa` *Plot Log-OR vs. X for Gamma Discriminant Function Approach*

Description

When `p_gdfa` is fit with `constant_or = FALSE`, the log-OR for X depends on the value of X (and covariates, if any). This function plots the log-OR vs. X for one or several sets of covariate values.

Usage

```
plot_gdfa(estimates, varcov = NULL, xrange, xname = "X",
  cvals = NULL, set_labels = NULL, set_panels = TRUE, ncol = 1)
```

Arguments

`estimates`	Numeric vector of point estimates for `(gamma_0, gamma_y, gamma_c^T, b1, b0)`.
`varcov`	Numeric matrix with variance-covariance matrix for `estimates`. If `NULL`, 95% confidence bands are omitted.
`xrange`	Numeric vector specifying range of X values to plot.
`xname`	Character vector specifying name of X variable, for plot title and x-axis label.

cvals	Numeric vector or list of numeric vectors specifying covariate values to use in log-odds ratio calculations.
set_labels	Character vector of labels for the sets of covariate values. Only used if cvals is a list.
set_panels	Logical value for whether to use separate panels for each set of covariate values, as opposed to using different colors on a single plot.
ncol	Integer value specifying number of columns for multi-panel figure. Only used if there are multiple sets of covariate values (i.e. cvals is a list).

Value

Plot of log-OR vs. X generated by ggplot.

Examples

```
# Fit Gamma discriminant function model for poolwise X vs. (Y, C), without
# assuming a constant log-OR. Note that data were generated with a constant
# log-OR of 0.5.
data(dat_p_gdfa)
dat <- dat_p_gdfa$dat
c.list <- dat_p_gdfa$c.list
fit <- p_gdfa(
  g = dat$g,
  y = dat$y,
  xtilde = dat$x,
  c = c.list,
  errors = "neither",
  constant_or = FALSE
)

# Plot estimated log-OR vs. X, holding C fixed at the sample mean.
p <- plot_gdfa(
  estimates = fit$estimates,
  varcov = fit$theta.var,
  xrange = range(dat$x[dat$g == 1]),
  cvals = mean(unlist(c.list))
)
p
```

---

plot_ndfa *Plot Log-OR vs. X for Normal Discriminant Function Approach*

Description

When p_ndfa is fit with constant_or = FALSE, the log-OR for X depends on the value of X (and covariates, if any). This function plots the log-OR vs. X for one or several sets of covariate values.

Usage

```
plot_ndfa(estimates, varcov = NULL, xrange, xname = "X",
  cvals = NULL, set_labels = NULL, set_panels = TRUE)
```

Arguments

estimates	Numeric vector of point estimates for (gamma_0, gamma_y, gamma_c^T, sigsq).
varcov	Numeric matrix with variance-covariance matrix for estimates. If NULL, 95% confidence bands are omitted.
xrange	Numeric vector specifying range of X values to plot.
xname	Character vector specifying name of X variable, for plot title and x-axis label.
cvals	Numeric vector or list of numeric vectors specifying covariate values to use in log-odds ratio calculations.
set_labels	Character vector of labels for the sets of covariate values. Only used if cvals is a list.
set_panels	Logical value for whether to use separate panels for each set of covariate values, as opposed to using different colors on a single plot.

Value

Plot of log-OR vs. X generated by ggplot.

Examples

```
# Fit discriminant function model for poolwise X vs. (Y, C), without assuming
# a constant log-OR. Note that data were generated with a constant log-OR of
# 0.5.
data(dat_p_ndfa)
dat <- dat_p_ndfa$dat
fit <- p_ndfa(
  g = dat$g,
  y = dat$numcases,
  xtilde = dat$x,
  c = dat$c,
  errors = "neither",
  constant_or = FALSE
)

# Plot estimated log-OR vs. X, holding C fixed at the sample mean.
p <- plot_ndfa(
  estimates = fit$estimates,
  varcov = fit$theta.var,
  xrange = range(dat$x[dat$g == 1]),
  cvals = mean(dat$c / dat$g)
)
p
```

poolcost_t *Visualize Total Costs for Pooling Design as a Function of Pool Size*

Description

Useful for determining whether pooling is a good idea, what pool size minimizes costs, and how many assays are needed for a target power.

Usage

```
poolcost_t(g = 1:10, d = NULL, mu1 = NULL, mu2 = NULL,
  sigsq = NULL, sigsq1 = sigsq, sigsq2 = sigsq, sigsq_p = 0,
  sigsq_m = 0, multiplicative = FALSE, alpha = 0.05, beta = 0.2,
  assay_cost = 100, other_costs = 0, labels = TRUE, ylim = NULL)
```

Arguments

g	Numeric vector of pool sizes to include.
d	Numeric value specifying true difference in group means.
mu1, mu2	Numeric value specifying group means. Required if multiplicative = TRUE.
sigsq	Numeric value specifying the variance of observations.
sigsq1, sigsq2	Numeric value specifying the variance of observations for each group.
sigsq_p	Numeric value specifying the variance of processing errors.
sigsq_m	Numeric value specifying the variance of measurement errors.
multiplicative	Logical value for whether to assume multiplicative rather than additive errors.
alpha	Numeric value specifying type-1 error rate.
beta	Numeric value specifying type-2 error rate.
assay_cost	Numeric value specifying cost of each assay.
other_costs	Numeric value specifying other per-subject costs.
labels	Logical value.
ylim	Numeric vector.

Value

Plot of total costs vs. pool size generated by ggplot.

Examples

```
# Plot total study costs vs. pool size for d = 0.25, sigsq = 1, and costs of
# $100 per assay and $0 in other per-subject costs.
poolcost_t(d = 0.25, sigsq = 1)

# Repeat but with additive processing error and $10 in per-subject costs.
poolcost_t(d = 0.25, sigsq = 1, sigsq_p = 0.5, other_costs = 10)
```

poolcushion_t *Visualize T-test Power for Pooling Design as Function of Processing Error Variance*

Description

Useful for choosing a sample size such that power will be adequate even if the processing errors are larger than anticipated.

Usage

```
poolcushion_t(g = NULL, n = NULL, d = NULL, mu1 = NULL,
  mu2 = NULL, sigsq = NULL, sigsq1 = sigsq, sigsq2 = sigsq,
  sigsq_p_predicted = 0, sigsq_p_range = NULL, sigsq_m = 0,
  multiplicative = FALSE, alpha = 0.05, beta = 0.2, labels = TRUE)
```

Arguments

g	Numeric value specifying the pool size.
n	Numeric value specifying the number of assays per group. If unspecified, function figures out n required for 100 (1 - beta)% power when sigsq_p = 0.
d	Numeric value specifying true difference in group means.
mu1, mu2	Numeric value specifying group means. Required if multiplicative = TRUE.
sigsq	Numeric value specifying the variance of observations.
sigsq1, sigsq2	Numeric value specifying the variance of observations for each group.
sigsq_p_predicted	Numeric value specifying predicted processing error variance. Used to calculate n if n is unspecified.
sigsq_p_range	Numeric vector specifying range of processing error variances to consider.
sigsq_m	Numeric value specifying the variance of measurement errors.
multiplicative	Logical value for whether to assume multiplicative rather than additive errors.
alpha	Numeric value specifying type-1 error rate.
beta	Numeric value specifying type-2 error rate. Only used if n = NULL.
labels	Logical value.

Value

Plot generated by ggplot.

Examples

```
# Determine optimal pool size and number of assays to detect a difference in
# group means of 0.5, with a common variance of 1, processing errors with
# variance of 0.1, and measurement errors with variance of 0.2. Assume costs
# of $100 per assay and $10 per subject.
poolcost_t(
  g = 1: 10,
  d = 0.5,
  sigsq = 1,
  sigsq_p = 0.1,
  sigsq_m = 0.2,
  assay_cost = 100,
  other_costs = 10
)

# Visualize how power of the study will be affected if the true processing
# error variance is not exactly 0.1.
poolcushion_t(
  g = 7,
  n = 29,
  d = 0.5,
  sigsq = 1,
  sigsq_p_predicted = 0.1,
  sigsq_m = 0.2
)
```

---

pooling *Fit Poolwise Regression Models*

Description

Functions for calculating power and fitting regression models in studies where a biomarker is measured in "pooled" samples rather than for each individual. Approaches for handling measurement error follow the framework of Schisterman et al. (2010) <doi:10.1002/sim.3823>.

Details

Package:	pooling
Type:	Package
Version:	1.1.2
Date:	2020-02-12
License:	GPL-3

Author(s)

Dane R. Van Domelen
<vandomed@gmail.com>

References

Acknowledgment: This material is based upon work supported by the National Science Foundation Graduate Research Fellowship under Grant No. DGE-0940903.

`poolpower_t` *Visualize T-test Power for Pooling Design*

Description

Useful for assessing efficiency gains that might be achieved with a pooling design.

Usage

```
poolpower_t(g = c(1, 3, 10), d = NULL, mu1 = NULL, mu2 = NULL,
  sigsq = NULL, sigsq1 = sigsq, sigsq2 = sigsq, sigsq_p = 0,
  sigsq_m = 0, multiplicative = FALSE, alpha = 0.05, beta = 0.2,
  assay_cost = 100, other_costs = 0, labels = TRUE)
```

Arguments

`g`	Numeric vector of pool sizes to include.
`d`	Numeric value specifying true difference in group means.
`mu1`, `mu2`	Numeric value specifying group means. Required if `multiplicative = TRUE`.
`sigsq`	Numeric value specifying the variance of observations.
`sigsq1`, `sigsq2`	Numeric value specifying the variance of observations for each group.
`sigsq_p`	Numeric value specifying the variance of processing errors.
`sigsq_m`	Numeric value specifying the variance of measurement errors.
`multiplicative`	Logical value for whether to assume multiplicative rather than additive errors.
`alpha`	Numeric value specifying type-1 error rate.
`beta`	Numeric value specifying type-2 error rate.
`assay_cost`	Numeric value specifying cost of each assay.
`other_costs`	Numeric value specifying other per-subject costs.
`labels`	Logical value.

Value

Plot of power vs. total costs generated by `ggplot`.

Examples

```
# Plot power vs. total study costs for d = 0.25, sigsq = 1, and costs of $100
# per assay and $0 in other per-subject costs.
poolpower_t(d = 0.5, sigsq = 1, assay_cost = 100, other_costs = 0)

# Repeat but with $10 in per-subject costs.
poolpower_t(d = 0.5, sigsq = 1, assay_cost = 100, other_costs = 10)

# Back to no per-subject costs, but with processing and measurement error
poolpower_t(d = 0.5, sigsq = 1, sigsq_p = 0.2, sigsq_m = 0.1,
            assay_cost = 100, other_costs = 0)
```

`poolvar_t` *Visualize Ratio of Variance of Each Pooled Measurement to Variance of Each Unpooled Measurement as Function of Pool Size*

Description

Useful for determining whether pooling is a good idea, and finding the optimal pool size if it is.

Usage

```
poolvar_t(g = 1:10, mu1 = NULL, mu2 = NULL, sigsq = NULL,
  sigsq1 = sigsq, sigsq2 = sigsq, sigsq_p = 0, sigsq_m = 0,
  multiplicative = FALSE, assay_cost = 100, other_costs = 0,
  labels = TRUE, ylim = NULL)
```

Arguments

`g`	Numeric vector of pool sizes to include.
`mu1`, `mu2`	Numeric value specifying group means. Required if `multiplicative = TRUE`.
`sigsq`	Numeric value specifying the variance of observations.
`sigsq1`, `sigsq2`	Numeric value specifying the variance of observations for each group.
`sigsq_p`	Numeric value specifying the variance of processing errors.
`sigsq_m`	Numeric value specifying the variance of measurement errors.
`multiplicative`	Logical value for whether to assume multiplicative rather than additive errors.
`assay_cost`	Numeric value specifying cost of each assay.
`other_costs`	Numeric value specifying other per-subject costs.
`labels`	Logical value.
`ylim`	Numeric vector.

Value

Plot generated by `ggplot`.

Examples

```
# Plot ratio of variances vs. pool size with default settings
poolvar_t(sigsq = 1)

# Add processing error and other per-subject costs
poolvar_t(sigsq = 1, sigsq_p = 0.2, other_costs = 0.1)
```

`p_dfa_xerrors` *Discriminant Function Approach for Estimating Odds Ratio with Normal Exposure Measured in Pools and Potentially Subject to Errors*

Description

Archived on 7/23/18. Please use `p_ndfa` instead.

Usage

```
p_dfa_xerrors(g, y, xtilde, c = NULL, constant_or = TRUE,
  errors = "both", ...)
```

Arguments

`g`	Numeric vector of pool sizes, i.e. number of members in each pool.
`y`	Numeric vector of poolwise `Y` values (number of cases in each pool).
`xtilde`	Numeric vector (or list of numeric vectors, if some pools have replicates) with `Xtilde` values.
`c`	Numeric matrix with poolwise `C` values (if any), with one row for each pool. Can be a vector if there is only 1 covariate.
`constant_or`	Logical value for whether to assume a constant OR for `X`, which means that `sigsq_1` = `sigsq_0`. If `NULL`, model is fit with and without this assumption, and likelihood ratio test is performed to test it.
`errors`	Character string specifying the errors that `X` is subject to. Choices are `"neither"`, `"processing"` for processing error only, `"measurement"` for measurement error only, and `"both"`.
`...`	Additional arguments to pass to `nlminb`.

Value

List of point estimates, variance-covariance matrix, object returned by `nlminb`, and AIC, for one or two models depending on `constant_or`. If `constant_or` = `NULL`, also returns result of a likelihood ratio test for `H0: sigsq_1 = sigsq_0`, which is equivalent to `H0: log-OR is constant`. If `constant_or` = `NULL`, returned objects with names ending in 1 are for model that does not assume constant log-OR, and those ending in 2 are for model that assumes constant log-OR.

References

Lyles, R.H., Van Domelen, D.R., Mitchell, E.M. and Schisterman, E.F. (2015) "A discriminant function approach to adjust for processing and measurement error When a biomarker is assayed in pooled samples." *Int. J. Environ. Res. Public Health* **12**(11): 14723–14740.

Schisterman, E.F., Vexler, A., Mumford, S.L. and Perkins, N.J. (2010) "Hybrid pooled-unpooled design for cost-efficient measurement of biomarkers." *Stat. Med.* **29**(5): 597–613.

Examples

```
# Load dataset containing poolwise (Y, Xtilde, C) values for pools of size
# 1, 2, and 3. Xtilde values are affected by processing error.
data(pdat1)

# Estimate log-OR for X and Y adjusted for C, ignoring processing error
fit1 <- p_dfa_xerrors(g = pdat1$g, y = pdat1$numcases, xtilde = pdat1$xtilde,
                      c = pdat1$c, errors = "neither")
fit1$estimates

# Repeat, but accounting for processing error. Closer to true log-OR of 0.5.
fit2 <- p_dfa_xerrors(g = pdat1$g, y = pdat1$numcases, xtilde = pdat1$xtilde,
                      c = pdat1$c, errors = "processing")
fit2$estimates
```

`p_dfa_xerrors2` *Discriminant Function Approach for Estimating Odds Ratio with Gamma Exposure Measured in Pools and Potentially Subject to Errors*

Description

Archived on 7/23/18. Please use `p_gdfa` instead.

Usage

```
p_dfa_xerrors2(g, y, xtilde, c = NULL, constant_or = TRUE,
  errors = "both", integrate_tol = 1e-08,
  integrate_tol_hessian = integrate_tol, estimate_var = TRUE,
  fix_posdef = FALSE, ...)
```

Arguments

`g`	Numeric vector with pool sizes, i.e. number of members in each pool.
`y`	Numeric vector with poolwise `Y` values, coded 0 if all members are controls and 1 if all members are cases.
`xtilde`	Numeric vector (or list of numeric vectors, if some pools have replicates) with `Xtilde` values.

c	List where each element is a numeric matrix containing the C values for members of a particular pool (1 row for each member).
constant_or	Logical value for whether to assume a constant OR for X, which means that gamma_y = 0. If NULL, model is fit with and without this assumption, and likelihood ratio test is performed to test it.
errors	Character string specifying the errors that X is subject to. Choices are "neither", "processing" for processing error only, "measurement" for measurement error only, and "both".
integrate_tol	Numeric value specifying the tol input to hcubature.
integrate_tol_hessian	Same as integrate_tol, but for use when estimating the Hessian matrix only. Sometimes more precise integration (i.e. smaller tolerance) helps prevent cases where the inverse Hessian is not positive definite.
estimate_var	Logical value for whether to return variance-covariance matrix for parameter estimates.
fix_posdef	Logical value for whether to repeatedly reduce integrate_tol_hessian by factor of 5 and re-estimate Hessian to try to avoid non-positive definite variance-covariance matrix.
...	Additional arguments to pass to nlminb.

Value

List of point estimates, variance-covariance matrix, objects returned by nlminb, and AICs, for one or two models depending on constant_or. If constant_or = NULL, also returns result of a likelihood ratio test for H0: gamma_y = 0, which is equivalent to H0: log-OR is constant. If constant_or = NULL, returned objects with names ending in 1 are for model that does not assume constant log-OR, and those ending in 2 are for model that assumes constant log-OR.

References

Lyles, R.H., Van Domelen, D.R., Mitchell, E.M. and Schisterman, E.F. (2015) "A discriminant function approach to adjust for processing and measurement error When a biomarker is assayed in pooled samples." *Int. J. Environ. Res. Public Health* **12**(11): 14723–14740.

Mitchell, E.M, Lyles, R.H., and Schisterman, E.F. (2015) "Positing, fitting, and selecting regression models for pooled biomarker data." *Stat. Med* **34**(17): 2544–2558.

Schisterman, E.F., Vexler, A., Mumford, S.L. and Perkins, N.J. (2010) "Hybrid pooled-unpooled design for cost-efficient measurement of biomarkers." *Stat. Med.* **29**(5): 597–613.

Whitcomb, B.W., Perkins, N.J., Zhang, Z., Ye, A., and Lyles, R. H. (2012) "Assessment of skewed exposure in case-control studies with pooling." *Stat. Med.* **31**: 2461–2472.

Examples

```
# Load dataset with (g, Y, Xtilde, C) values for 248 pools and list of C
# values for members of each pool. Xtilde values are affected by processing
# error.
data(pdat2)
dat <- pdat2$dat
```

```
c.list <- pdat2$c.list

# Estimate log-OR for X and Y adjusted for C, ignoring processing error
fit1 <- p_dfa_xerrors2(
  g = dat$g,
  y = dat$y,
  xtilde = dat$xtilde,
  c = c.list,
  errors = "neither"
)
fit1$estimates

# Repeat, but accounting for processing error.
## Not run:
fit2 <- p_dfa_xerrors2(
  g = dat$g,
  y = dat$y,
  xtilde = dat$xtilde,
  c = c.list,
  errors = "processing",
  control = list(trace = 1)
)
fit2$estimates

## End(Not run)
```

---

p_gdfa *Gamma Discriminant Function Approach for Estimating Odds Ratio with Exposure Measured in Pools and Potentially Subject to Multiplicative Lognormal Errors*

Description

Assumes exposure given covariates and outcome is a constant-scale Gamma regression. Pooled exposure measurements can be assumed precise or subject to multiplicative lognormal processing error and/or measurement error. Parameters are estimated using maximum likelihood.

Usage

```
p_gdfa(g, y, xtilde, c = NULL, constant_or = TRUE,
  errors = "processing", estimate_var = TRUE,
  start_nonvar_var = c(0.01, 1), lower_nonvar_var = c(-Inf, 1e-04),
  upper_nonvar_var = c(Inf, Inf), jitter_start = 0.01,
  hcubature_list = list(tol = 1e-08), nlminb_list = list(control =
  list(trace = 1, eval.max = 500, iter.max = 500)),
  hessian_list = list(method.args = list(r = 4)), nlminb_object = NULL)
```

Arguments

g	Numeric vector with pool sizes, i.e. number of members in each pool.
y	Numeric vector with poolwise Y values, coded 0 if all members are controls and 1 if all members are cases.
xtilde	Numeric vector (or list of numeric vectors, if some pools have replicates) with Xtilde values.
c	List where each element is a numeric matrix containing the **C** values for members of a particular pool (1 row for each member).
constant_or	Logical value for whether to assume a constant OR for X, which means that gamma_y = 0. If NULL, model is fit with and without this assumption, and a likelihood ratio test is performed to test it.
errors	Character string specifying the errors that X is subject to. Choices are "neither", "processing" for processing error only, "measurement" for measurement error only, and "both".
estimate_var	Logical value for whether to return variance-covariance matrix for parameter estimates.
start_nonvar_var	Numeric vector of length 2 specifying starting value for non-variance terms and variance terms, respectively.
lower_nonvar_var	Numeric vector of length 2 specifying lower bound for non-variance terms and variance terms, respectively.
upper_nonvar_var	Numeric vector of length 2 specifying upper bound for non-variance terms and variance terms, respectively.
jitter_start	Numeric value specifying standard deviation for mean-0 normal jitters to add to starting values for a second try at maximizing the log-likelihood, should the initial call to nlminb result in non-convergence. Set to NULL for no second try.
hcubature_list	List of arguments to pass to hcubature for numerical integration.
nlminb_list	List of arguments to pass to nlminb for log-likelihood maximization.
hessian_list	List of arguments to pass to hessian for approximating the Hessian matrix. Only used if estimate_var = TRUE.
nlminb_object	Object returned from nlminb in a prior call. Useful for bypassing log-likelihood maximization if you just want to re-estimate the Hessian matrix with different options.

Value

List containing:

1. Numeric vector of parameter estimates.
2. Variance-covariance matrix.
3. Returned nlminb object from maximizing the log-likelihood function.
4. Akaike information criterion (AIC).

If `constant_or = NULL`, two such lists are returned (one under a constant odds ratio assumption and one not), along with a likelihood ratio test for `H0: gamma_y = 0`, which is equivalent to `H0: odds ratio is constant`.

References

Lyles, R.H., Van Domelen, D.R., Mitchell, E.M. and Schisterman, E.F. (2015) "A discriminant function approach to adjust for processing and measurement error When a biomarker is assayed in pooled samples." *Int. J. Environ. Res. Public Health* **12**(11): 14723–14740.

Mitchell, E.M, Lyles, R.H., and Schisterman, E.F. (2015) "Positing, fitting, and selecting regression models for pooled biomarker data." *Stat. Med* **34**(17): 2544–2558.

Schisterman, E.F., Vexler, A., Mumford, S.L. and Perkins, N.J. (2010) "Hybrid pooled-unpooled design for cost-efficient measurement of biomarkers." *Stat. Med.* **29**(5): 597–613.

Whitcomb, B.W., Perkins, N.J., Zhang, Z., Ye, A., and Lyles, R. H. (2012) "Assessment of skewed exposure in case-control studies with pooling." *Stat. Med.* **31**: 2461–2472.

Examples

```
# Load data frame with (g, Y, X, Xtilde) values for 496 pools, list of C
# values for members of each pool, and list of Xtilde values where 25
# single-specimen pools have replicates. Xtilde values are affected by
# processing error and measurement error. True log-OR = 0.5, sigsq_p = 0.25,
# sigsq_m = 0.1.
data(dat_p_gdfa)
dat <- dat_p_gdfa$dat
reps <- dat_p_gdfa$reps
c.list <- dat_p_gdfa$c.list

# Unobservable truth estimator - use precise X's
fit.unobservable <- p_gdfa(
  g = dat$g,
  y = dat$y,
  xtilde = dat$x,
  c = c.list,
  errors = "neither"
)
fit.unobservable$estimates

# Naive estimator - use imprecise Xtilde's, but treat as precise
fit.naive <- p_gdfa(
  g = dat$g,
  y = dat$y,
  xtilde = dat$xtilde,
  c = c.list,
  errors = "neither"
)
fit.naive$estimates

# Corrected estimator - use Xtilde's and account for errors (not using
# replicates here)
## Not run:
```

```
fit.noreps <- p_gdfa(
  g = dat$g,
  y = dat$y,
  xtilde = dat$xtilde,
  c = c.list,
  errors = "both"
)
fit.noreps$estimates

# Corrected estimator - use Xtilde's including 25 replicates
fit.reps <- p_gdfa(
  g = dat$g,
  y = dat$y,
  xtilde = reps,
  c = c.list,
  errors = "both"
)
fit.reps$estimates

# Same as previous, but allowing for non-constant odds ratio.
fit.nonconstant <- p_gdfa(
  g = dat$g,
  y = dat$y,
  xtilde = reps,
  c = c.list,
  constant_or = FALSE,
  errors = "both",
  hcubature_list = list(tol = 1e-4)
)
fit.nonconstant$estimates

# Visualize estimated log-OR vs. X based on previous model fit
p <- plot_gdfa(
  estimates = fit.nonconstant$estimates,
  varcov = fit.nonconstant$theta.var,
  xrange = range(dat$xtilde[dat$g == 1]),
  cvals = mean(unlist(c))
)
p

## End(Not run)
```

---

p_gdfa_constant *Gamma Discriminant Function Approach for Estimating Odds Ratio with Exposure Measured in Pools and Potentially Subject to Multiplicative Lognormal Errors (Constant Odds Ratio Version)*

Description

See p_gdfa.

Usage

```
p_gdfa_constant(g, y, xtilde, c = NULL, errors = "processing",
  estimate_var = TRUE, start_nonvar_var = c(0.01, 1),
  lower_nonvar_var = c(-Inf, 1e-04), upper_nonvar_var = c(Inf, Inf),
  jitter_start = 0.01, hcubature_list = list(tol = 1e-08),
  nlminb_list = list(control = list(trace = 1, eval.max = 500, iter.max =
  500)), hessian_list = list(method.args = list(r = 4)),
  nlminb_object = NULL)
```

Arguments

g	Numeric vector with pool sizes, i.e. number of members in each pool.
y	Numeric vector with poolwise Y values, coded 0 if all members are controls and 1 if all members are cases.
xtilde	Numeric vector (or list of numeric vectors, if some pools have replicates) with Xtilde values.
c	List where each element is a numeric matrix containing the **C** values for members of a particular pool (1 row for each member).
errors	Character string specifying the errors that X is subject to. Choices are `"neither"`, `"processing"` for processing error only, `"measurement"` for measurement error only, and `"both"`.
estimate_var	Logical value for whether to return variance-covariance matrix for parameter estimates.
start_nonvar_var	Numeric vector of length 2 specifying starting value for non-variance terms and variance terms, respectively.
lower_nonvar_var	Numeric vector of length 2 specifying lower bound for non-variance terms and variance terms, respectively.
upper_nonvar_var	Numeric vector of length 2 specifying upper bound for non-variance terms and variance terms, respectively.
jitter_start	Numeric value specifying standard deviation for mean-0 normal jitters to add to starting values for a second try at maximizing the log-likelihood, should the initial call to `nlminb` result in non-convergence. Set to `NULL` for no second try.
hcubature_list	List of arguments to pass to `hcubature` for numerical integration.
nlminb_list	List of arguments to pass to `nlminb` for log-likelihood maximization.
hessian_list	List of arguments to pass to `hessian` for approximating the Hessian matrix. Only used if `estimate_var = TRUE`.
nlminb_object	Object returned from `nlminb` in a prior call. Useful for bypassing log-likelihood maximization if you just want to re-estimate the Hessian matrix with different options.

Value

List containing:

1. Numeric vector of parameter estimates.
2. Variance-covariance matrix.
3. Returned nlminb object from maximizing the log-likelihood function.
4. Akaike information criterion (AIC).

References

Lyles, R.H., Van Domelen, D.R., Mitchell, E.M. and Schisterman, E.F. (2015) "A discriminant function approach to adjust for processing and measurement error When a biomarker is assayed in pooled samples." *Int. J. Environ. Res. Public Health* **12**(11): 14723–14740.

Mitchell, E.M, Lyles, R.H., and Schisterman, E.F. (2015) "Positing, fitting, and selecting regression models for pooled biomarker data." *Stat. Med* **34**(17): 2544–2558.

Schisterman, E.F., Vexler, A., Mumford, S.L. and Perkins, N.J. (2010) "Hybrid pooled-unpooled design for cost-efficient measurement of biomarkers." *Stat. Med.* **29**(5): 597–613.

Whitcomb, B.W., Perkins, N.J., Zhang, Z., Ye, A., and Lyles, R. H. (2012) "Assessment of skewed exposure in case-control studies with pooling." *Stat. Med.* **31**: 2461–2472.

---

p_gdfa_nonconstant *Gamma Discriminant Function Approach for Estimating Odds Ratio with Exposure Measured in Pools and Potentially Subject to Multiplicative Lognormal Errors (Non-constant Odds Ratio Version)*

Description

See p_gdfa.

Usage

```
p_gdfa_nonconstant(g, y, xtilde, c = NULL, errors = "processing",
  estimate_var = TRUE, start_nonvar_var = c(0.01, 1),
  lower_nonvar_var = c(-Inf, 1e-04), upper_nonvar_var = c(Inf, Inf),
  jitter_start = 0.01, hcubature_list = list(tol = 1e-08),
  nlminb_list = list(control = list(trace = 1, eval.max = 500, iter.max =
  500)), hessian_list = list(method.args = list(r = 4)),
  nlminb_object = NULL)
```

Arguments

g	Numeric vector with pool sizes, i.e. number of members in each pool.
y	Numeric vector with poolwise Y values, coded 0 if all members are controls and 1 if all members are cases.

xtilde	Numeric vector (or list of numeric vectors, if some pools have replicates) with Xtilde values.
c	List where each element is a numeric matrix containing the **C** values for members of a particular pool (1 row for each member).
errors	Character string specifying the errors that X is subject to. Choices are "neither", "processing" for processing error only, "measurement" for measurement error only, and "both".
estimate_var	Logical value for whether to return variance-covariance matrix for parameter estimates.
start_nonvar_var	Numeric vector of length 2 specifying starting value for non-variance terms and variance terms, respectively.
lower_nonvar_var	Numeric vector of length 2 specifying lower bound for non-variance terms and variance terms, respectively.
upper_nonvar_var	Numeric vector of length 2 specifying upper bound for non-variance terms and variance terms, respectively.
jitter_start	Numeric value specifying standard deviation for mean-0 normal jitters to add to starting values for a second try at maximizing the log-likelihood, should the initial call to nlminb result in non-convergence. Set to NULL for no second try.
hcubature_list	List of arguments to pass to hcubature for numerical integration.
nlminb_list	List of arguments to pass to nlminb for log-likelihood maximization.
hessian_list	List of arguments to pass to hessian for approximating the Hessian matrix. Only used if estimate_var = TRUE.
nlminb_object	Object returned from nlminb in a prior call. Useful for bypassing log-likelihood maximization if you just want to re-estimate the Hessian matrix with different options.

Value

List containing:

1. Numeric vector of parameter estimates.
2. Variance-covariance matrix.
3. Returned nlminb object from maximizing the log-likelihood function.
4. Akaike information criterion (AIC).

References

Lyles, R.H., Van Domelen, D.R., Mitchell, E.M. and Schisterman, E.F. (2015) "A discriminant function approach to adjust for processing and measurement error When a biomarker is assayed in pooled samples." *Int. J. Environ. Res. Public Health* **12**(11): 14723–14740.

Mitchell, E.M, Lyles, R.H., and Schisterman, E.F. (2015) "Positing, fitting, and selecting regression models for pooled biomarker data." *Stat. Med* **34**(17): 2544–2558.

Schisterman, E.F., Vexler, A., Mumford, S.L. and Perkins, N.J. (2010) "Hybrid pooled-unpooled design for cost-efficient measurement of biomarkers." *Stat. Med.* **29**(5): 597–613.

Whitcomb, B.W., Perkins, N.J., Zhang, Z., Ye, A., and Lyles, R. H. (2012) "Assessment of skewed exposure in case-control studies with pooling." *Stat. Med.* **31**: 2461–2472.

`p_linreg_yerrors` *Linear Regression of Y vs. Covariates with Y Measured in Pools and (Potentially) Subject to Additive Normal Errors*

Description

Assumes outcome given covariates is a normal-errors linear regression. Pooled outcome measurements can be assumed precise or subject to additive normal processing error and/or measurement error. Replicates are supported.

Usage

```
p_linreg_yerrors(g, ytilde, x = NULL, errors = "processing",
  estimate_var = TRUE, start_nonvar_var = c(0.01, 1),
  lower_nonvar_var = c(-Inf, 1e-04), upper_nonvar_var = c(Inf, Inf),
  nlminb_list = list(control = list(trace = 1, eval.max = 500, iter.max =
  500)), hessian_list = list(method.args = list(r = 4)))
```

Arguments

`g`	Numeric vector with pool sizes, i.e. number of members in each pool.
`ytilde`	Numeric vector (or list of numeric vectors, if some pools have replicates) with poolwise sum `Ytilde` values.
`x`	Numeric matrix with poolwise `X` values (if any), with one row for each pool. Can be a vector if there is only 1 covariate.
`errors`	Character string specifying the errors that `Y` is subject to. Choices are `"neither"`, `"processing"` for processing error only, `"measurement"` for measurement error only, and `"both"`.
`estimate_var`	Logical value for whether to return variance-covariance matrix for parameter estimates.
`start_nonvar_var`	Numeric vector of length 2 specifying starting value for non-variance terms and variance terms, respectively.
`lower_nonvar_var`	Numeric vector of length 2 specifying lower bound for non-variance terms and variance terms, respectively.
`upper_nonvar_var`	Numeric vector of length 2 specifying upper bound for non-variance terms and variance terms, respectively.
`nlminb_list`	List of arguments to pass to `nlminb` for log-likelihood maximization.
`hessian_list`	List of arguments to pass to `hessian`.

Details

The individual-level model of interest for Y|**X** is:

Y = beta_0 + **beta_x**^T **X** + e, e ~ N(0, sigsq)

The implied model for summed Y*|**X*** in a pool with g members is:

Y* = g beta_0 + **beta_x**^T **X*** + e*, e* ~ N(0, g sigsq)

The assay targets Ybar, the mean Y value for each pool, from which the sum Y* can be calculated as Y* = g Ybar. But the Ybar's may be subject to processing error and/or measurement error. Suppose Ybartilde is the imprecise version of Ybar from the assay. If both errors are present, the assumed error structure is:

Ybartilde = Ybar + e_p I(g > 1) + e_m, e_p ~ N(0, sigsq_p), e_m ~ N(0, sigsq_m)

with the processing error e_p and measurement error e_m assumed independent of each other. This motivates a maximum likelihood analysis for estimating **theta** = (beta_0, **beta_x**^T)^T based on observed (Ytilde*, **X***) values, where Ytilde* = g Ytildebar.

Value

List containing:

1. Numeric vector of parameter estimates.
2. Variance-covariance matrix (if `estimate_var = TRUE`).
3. Returned `nlminb` object from maximizing the log-likelihood function.
4. Akaike information criterion (AIC).

References

Schisterman, E.F., Vexler, A., Mumford, S.L. and Perkins, N.J. (2010) "Hybrid pooled-unpooled design for cost-efficient measurement of biomarkers." *Stat. Med.* **29**(5): 597–613.

Examples

```
# Load dataset containing data frame with (g, X1*, X2*, Y*, Ytilde*) values
# for 500 pools each of size 1, 2, and 3, and list of Ytilde values where 20
# of the single-specimen pools have replicates. Ytilde values are affected by
# processing error and measurement error; true parameter values are
# beta_0 = 0.25, beta_x1 = 0.5, beta_x2 = 0.25, sigsq = 1.
data(dat_p_linreg_yerrors)
dat <- dat_p_linreg_yerrors$dat
reps <- dat_p_linreg_yerrors$reps

# Fit Ytilde* vs. (X1*, X2*) ignoring errors in Ytilde (leads to loss of
# precision and overestimated sigsq, but no bias).
fit.naive <- p_linreg_yerrors(
  g = dat$g,
  y = dat$y,
  x = dat[, c("x1", "x2")],
  errors = "neither"
)
fit.naive$theta.hat
```

```
# Account for errors in Ytilde*, without using replicates
fit.corrected.noreps <- p_linreg_yerrors(
  g = dat$g,
  y = dat$ytilde,
  x = dat[, c("x1", "x2")],
  errors = "both"
)
fit.corrected.noreps$theta.hat

# Account for errors in Ytilde*, incorporating the 20 replicates
fit.corrected.reps <- p_linreg_yerrors(
  g = dat$g,
  y = reps,
  x = dat[, c("x1", "x2")],
  errors = "both"
)
fit.corrected.reps$theta.hat

# In this trial, incorporating replicates resulted in much better estimates
# of sigsq (truly 1), sigsq_p (truly 0.4), and sigsq_m (truly = 0.2) but very
# similar regression coefficient estimates.
fit.corrected.noreps$theta.hat
fit.corrected.reps$theta.hat
```

`p_logreg` *Poolwise Logistic Regression*

Description

Fit homogeneous-pools logistic regression model described by Weinberg & Umbach (1999).

Usage

```
p_logreg(g, y, x, method = "glm", prev = NULL, samp_y1y0 = NULL,
  estimate_var = TRUE, start = 0.01, lower = -Inf, upper = Inf,
  nlminb_list = list(control = list(trace = 1, eval.max = 500, iter.max =
  500)), hessian_list = list(method.args = list(r = 4)))
```

Arguments

`g`	Numeric vector with pool sizes, i.e. number of members in each pool.
`y`	Numeric vector with poolwise `Y` values, coded 0 if all members are controls and 1 if all members are cases.
`x`	Numeric matrix with poolwise `X` values, with one row for each pool. Can be a vector if there is only 1 predictor.

method	Character string specifying method to use for estimation. Choices are "glm" for glm function and "ml" for maximum likelihood.
prev	Numeric value specifying disease prevalence, allowing for valid estimation of the intercept with case-control sampling. Can specify samp_y1y0 instead if sampling rates are known.
samp_y1y0	Numeric vector of length 2 specifying sampling probabilities for cases and controls, allowing for valid estimation of the intercept with case-control sampling. Can specify prev instead if it's easier.
estimate_var	Logical value for whether to return variance-covariance matrix for parameter estimates.
start	Numeric value specifying starting values for parameters. Only used if method = "ml".
lower	Numeric value specifying lower bounds for parameters. Only used if method = "ml".
upper	Numeric value specifying upper bounds for parameters. Only used if method = "ml".
nlminb_list	List of arguments to pass to nlminb for log-likelihood maximization.
hessian_list	List of arguments to pass to hessian for approximating the Hessian matrix. Only used if method = "ml" and estimate_var = TRUE.

Value

List containing:

1. Numeric vector of parameter estimates.
2. Variance-covariance matrix (if estimate_var = TRUE).
3. Fitted glm object (if method = "glm") or returned nlminb object (if method = "ml").
4. Akaike information criterion (AIC).

References

Weinberg, C.R. and Umbach, D.M. (1999) "Using pooled exposure assessment to improve efficiency in case-control studies." *Biometrics* **55**: 718–726.

Weinberg, C.R. and Umbach, D.M. (2014) "Correction to 'Using pooled exposure assessment to improve efficiency in case-control studies' by Clarice R. Weinberg and David M. Umbach; 55, 718–726, September 1999." *Biometrics* **70**: 1061.

Examples

```
# Load dataset containing (Y, Xtilde, C) values for pools of size 1, 2, and 3
data(pdat1)

# Estimate log-OR for Xtilde and Y adjusted for C
fit <- p_logreg(g = pdat1$g, y = pdat1$allcases, x = pdat1[, c("xtilde", "c")])
fit$theta.hat
```

p_logreg_xerrors *Poolwise Logistic Regression with Normal Exposure Subject to Errors*

Description

Assumes normal linear model for exposure given covariates, and additive normal processing errors and measurement errors acting on the poolwise mean exposure. Manuscript fully describing the approach is under review.

Usage

```
p_logreg_xerrors(g, y, xtilde, c = NULL, errors = "processing",
  nondiff_pe = TRUE, nondiff_me = TRUE, constant_pe = TRUE,
  prev = NULL, samp_y1y0 = NULL, approx_integral = TRUE,
  estimate_var = TRUE, start_nonvar_var = c(0.01, 1),
  lower_nonvar_var = c(-Inf, 1e-04), upper_nonvar_var = c(Inf, Inf),
  jitter_start = 0.01, hcubature_list = list(tol = 1e-08),
  nlminb_list = list(control = list(trace = 1, eval.max = 500, iter.max =
  500)), hessian_list = list(method.args = list(r = 4)),
  nlminb_object = NULL)
```

Arguments

g	Numeric vector with pool sizes, i.e. number of members in each pool.
y	Numeric vector with poolwise Y values, coded 0 if all members are controls and 1 if all members are cases.
xtilde	Numeric vector (or list of numeric vectors, if some pools have replicates) with Xtilde values.
c	Numeric matrix with poolwise **C** values (if any), with one row for each pool. Can be a vector if there is only 1 covariate.
errors	Character string specifying the errors that X is subject to. Choices are `"neither"`, `"processing"` for processing error only, `"measurement"` for measurement error only, and `"both"`.
nondiff_pe	Logical value for whether to assume the processing error variance is non-differential, i.e. the same in case pools and control pools.
nondiff_me	Logical value for whether to assume the measurement error variance is non-differential, i.e. the same in case pools and control pools.
constant_pe	Logical value for whether to assume the processing error variance is constant with pool size. If `FALSE`, assumption is that processing error variance increase with pool size such that, for example, the processing error affecting a pool 2x as large as another has 2x the variance.
prev	Numeric value specifying disease prevalence, allowing for valid estimation of the intercept with case-control sampling. Can specify `samp_y1y0` instead if sampling rates are known.

`samp_y1y0` Numeric vector of length 2 specifying sampling probabilities for cases and controls, allowing for valid estimation of the intercept with case-control sampling. Can specify `prev` instead if it's easier.

`approx_integral` Logical value for whether to use the probit approximation for the logistic-normal integral, to avoid numerically integrating X's out of the likelihood function.

`estimate_var` Logical value for whether to return variance-covariance matrix for parameter estimates.

`start_nonvar_var` Numeric vector of length 2 specifying starting value for non-variance terms and variance terms, respectively.

`lower_nonvar_var` Numeric vector of length 2 specifying lower bound for non-variance terms and variance terms, respectively.

`upper_nonvar_var` Numeric vector of length 2 specifying upper bound for non-variance terms and variance terms, respectively.

`jitter_start` Numeric value specifying standard deviation for mean-0 normal jitters to add to starting values for a second try at maximizing the log-likelihood, should the initial call to `nlminb` result in non-convergence. Set to `NULL` for no second try.

`hcubature_list` List of arguments to pass to `hcubature` for numerical integration. Only used if `approx_integral = FALSE`.

`nlminb_list` List of arguments to pass to `nlminb` for log-likelihood maximization.

`hessian_list` List of arguments to pass to `hessian` for approximating the Hessian matrix. Only used if `estimate_var = TRUE`.

`nlminb_object` Object returned from `nlminb` in a prior call. Useful for bypassing log-likelihood maximization if you just want to re-estimate the Hessian matrix with different options.

Value

List containing:

1. Numeric vector of parameter estimates.
2. Variance-covariance matrix (if `estimate_var = TRUE`).
3. Returned `nlminb` object from maximizing the log-likelihood function.
4. Akaike information criterion (AIC).

References

Schisterman, E.F., Vexler, A., Mumford, S.L. and Perkins, N.J. (2010) "Hybrid pooled-unpooled design for cost-efficient measurement of biomarkers." *Stat. Med.* **29**(5): 597–613.

Weinberg, C.R. and Umbach, D.M. (1999) "Using pooled exposure assessment to improve efficiency in case-control studies." *Biometrics* **55**: 718–726.

Weinberg, C.R. and Umbach, D.M. (2014) "Correction to 'Using pooled exposure assessment to improve efficiency in case-control studies' by Clarice R. Weinberg and David M. Umbach; 55, 718–726, September 1999." *Biometrics* **70**: 1061.

Examples

```
# Load dataset containing (Y, Xtilde, C) values for pools of size 1, 2, and
# 3. Xtilde values are affected by processing error.
data(pdat1)

# Estimate log-OR for X and Y adjusted for C, ignoring processing error
fit1 <- p_logreg_xerrors(
  g = pdat1$g,
  y = pdat1$allcases,
  xtilde = pdat1$xtilde,
  c = pdat1$c,
  errors = "neither"
)
fit1$theta.hat

# Repeat, but accounting for processing error. Closer to true log-OR of 0.5.
fit2 <- p_logreg_xerrors(
  g = pdat1$g,
  y = pdat1$allcases,
  xtilde = pdat1$xtilde,
  c = pdat1$c,
  errors = "processing"
)
fit2$theta.hat
```

---

p_logreg_xerrors2 *Poolwise Logistic Regression with Gamma Exposure Subject to Errors*

Description

Assumes constant-scale Gamma model for exposure given covariates, and multiplicative lognormal processing errors and measurement errors acting on the poolwise mean exposure. Manuscript fully describing the approach is under review.

Usage

```
p_logreg_xerrors2(g = NULL, y, xtilde, c = NULL,
  errors = "processing", nondiff_pe = TRUE, nondiff_me = TRUE,
  constant_pe = TRUE, prev = NULL, samp_y1y0 = NULL,
  estimate_var = TRUE, start_nonvar_var = c(0.01, 1),
  lower_nonvar_var = c(-Inf, 1e-04), upper_nonvar_var = c(Inf, Inf),
  jitter_start = 0.01, hcubature_list = list(tol = 1e-08),
  nlminb_list = list(control = list(trace = 1, eval.max = 500, iter.max =
  500)), hessian_list = list(method.args = list(r = 4)),
  nlminb_object = NULL)
```

Arguments

g	Numeric vector with pool sizes, i.e. number of members in each pool.
y	Numeric vector with poolwise Y values, coded 0 if all members are controls and 1 if all members are cases.
xtilde	Numeric vector (or list of numeric vectors, if some pools have replicates) with Xtilde values.
c	List where each element is a numeric matrix containing the **C** values for members of a particular pool (1 row for each member).
errors	Character string specifying the errors that X is subject to. Choices are "neither", "processing" for processing error only, "measurement" for measurement error only, and "both".
nondiff_pe	Logical value for whether to assume the processing error variance is non-differential, i.e. the same in case pools and control pools.
nondiff_me	Logical value for whether to assume the measurement error variance is non-differential, i.e. the same in case pools and control pools.
constant_pe	Logical value for whether to assume the processing error variance is constant with pool size. If FALSE, assumption is that processing error variance increase with pool size such that, for example, the processing error affecting a pool 2x as large as another has 2x the variance.
prev	Numeric value specifying disease prevalence, allowing for valid estimation of the intercept with case-control sampling. Can specify samp_y1y0 instead if sampling rates are known.
samp_y1y0	Numeric vector of length 2 specifying sampling probabilities for cases and controls, allowing for valid estimation of the intercept with case-control sampling. Can specify prev instead if it's easier.
estimate_var	Logical value for whether to return variance-covariance matrix for parameter estimates.
start_nonvar_var	Numeric vector of length 2 specifying starting value for non-variance terms and variance terms, respectively.
lower_nonvar_var	Numeric vector of length 2 specifying lower bound for non-variance terms and variance terms, respectively.
upper_nonvar_var	Numeric vector of length 2 specifying upper bound for non-variance terms and variance terms, respectively.
jitter_start	Numeric value specifying standard deviation for mean-0 normal jitters to add to starting values for a second try at maximizing the log-likelihood, should the initial call to nlminb result in non-convergence. Set to NULL for no second try.
hcubature_list	List of arguments to pass to hcubature for numerical integration.
nlminb_list	List of arguments to pass to nlminb for log-likelihood maximization.
hessian_list	List of arguments to pass to hessian for approximating the Hessian matrix. Only used if estimate_var = TRUE.

`nlminb_object` Object returned from `nlminb` in a prior call. Useful for bypassing log-likelihood maximization if you just want to re-estimate the Hessian matrix with different options.

Value

List containing:

1. Numeric vector of parameter estimates.
2. Variance-covariance matrix (if `estimate_var = TRUE`).
3. Returned `nlminb` object from maximizing the log-likelihood function.
4. Akaike information criterion (AIC).

References

Mitchell, E.M, Lyles, R.H., and Schisterman, E.F. (2015) "Positing, fitting, and selecting regression models for pooled biomarker data." *Stat. Med* **34**(17): 2544–2558.

Schisterman, E.F., Vexler, A., Mumford, S.L. and Perkins, N.J. (2010) "Hybrid pooled-unpooled design for cost-efficient measurement of biomarkers." *Stat. Med.* **29**(5): 597–613.

Weinberg, C.R. and Umbach, D.M. (1999) "Using pooled exposure assessment to improve efficiency in case-control studies." *Biometrics* **55**: 718–726.

Weinberg, C.R. and Umbach, D.M. (2014) "Correction to 'Using pooled exposure assessment to improve efficiency in case-control studies' by Clarice R. Weinberg and David M. Umbach; 55, 718–726, September 1999." *Biometrics* **70**: 1061.

Whitcomb, B.W., Perkins, N.J., Zhang, Z., Ye, A., and Lyles, R. H. (2012) "Assessment of skewed exposure in case-control studies with pooling." *Stat. Med.* **31**: 2461–2472.

Examples

```
# Load dataset with (g, Y, Xtilde, C) values for 248 pools and list of C
# values for members of each pool. Xtilde values are affected by processing
# error.
data(pdat2)
dat <- pdat2$dat
c.list <- pdat2$c.list

# Estimate log-OR for X and Y adjusted for C, ignoring processing error
fit1 <- p_logreg_xerrors2(
  g = dat$g,
  y = dat$y,
  xtilde = dat$xtilde,
  c = c.list,
  errors = "neither"
)
fit1$theta.hat

# Repeat, but accounting for processing error.
## Not run:
fit2 <- p_logreg_xerrors2(
```

```
  g = dat$g,
  y = dat$y,
  xtilde = dat$xtilde,
  c = c.list,
  errors = "processing"
)
fit2$theta.hat

## End(Not run)
```

---

p_ndfa *Normal Discriminant Function Approach for Estimating Odds Ratio with Exposure Measured in Pools and Potentially Subject to Additive Normal Errors*

Description

Assumes exposure given covariates and outcome is a normal-errors linear regression. Pooled exposure measurements can be assumed precise or subject to additive normal processing error and/or measurement error. Parameters are estimated using maximum likelihood.

Usage

```
p_ndfa(g, y, xtilde, c = NULL, constant_or = TRUE,
  errors = "processing", start_nonvar_var = c(0.01, 1),
  lower_nonvar_var = c(-Inf, 1e-04), upper_nonvar_var = c(Inf, Inf),
  jitter_start = 0.01, nlminb_list = list(control = list(trace = 1,
  eval.max = 500, iter.max = 500)), hessian_list = list(method.args =
  list(r = 4)), nlminb_object = NULL)
```

Arguments

g	Numeric vector of pool sizes, i.e. number of members in each pool.
y	Numeric vector of poolwise Y values (number of cases in each pool).
xtilde	Numeric vector (or list of numeric vectors, if some pools have replicates) with Xtilde values.
c	Numeric matrix with poolwise **C** values (if any), with one row for each pool. Can be a vector if there is only 1 covariate.
constant_or	Logical value for whether to assume a constant odds ratio for X, which means that sigsq_1 = sigsq_0. If NULL, model is fit with and without this assumption, and a likelihood ratio test is performed to test it.
errors	Character string specifying the errors that X is subject to. Choices are "neither", "processing" for processing error only, "measurement" for measurement error only, and "both".

start_nonvar_var
: Numeric vector of length 2 specifying starting value for non-variance terms and variance terms, respectively.

lower_nonvar_var
: Numeric vector of length 2 specifying lower bound for non-variance terms and variance terms, respectively.

upper_nonvar_var
: Numeric vector of length 2 specifying upper bound for non-variance terms and variance terms, respectively.

jitter_start
: Numeric value specifying standard deviation for mean-0 normal jitters to add to starting values for a second try at maximizing the log-likelihood, should the initial call to nlminb result in non-convergence. Set to NULL for no second try.

nlminb_list
: List of arguments to pass to nlminb for log-likelihood maximization.

hessian_list
: List of arguments to pass to hessian for approximating the Hessian matrix. Only used if estimate_var = TRUE.

nlminb_object
: Object returned from nlminb in a prior call. Useful for bypassing log-likelihood maximization if you just want to re-estimate the Hessian matrix with different options.

Value

List containing:

1. Numeric vector of parameter estimates.
2. Variance-covariance matrix.
3. Returned nlminb object from maximizing the log-likelihood function.
4. Akaike information criterion (AIC).

If constant_or = NULL, two such lists are returned (one under a constant odds ratio assumption and one not), along with a likelihood ratio test for H0: sigsq_1 = sigsq_0, which is equivalent to H0: odds ratio is constant.

References

Lyles, R.H., Van Domelen, D.R., Mitchell, E.M. and Schisterman, E.F. (2015) "A discriminant function approach to adjust for processing and measurement error When a biomarker is assayed in pooled samples." *Int. J. Environ. Res. Public Health* **12**(11): 14723–14740.

Schisterman, E.F., Vexler, A., Mumford, S.L. and Perkins, N.J. (2010) "Hybrid pooled-unpooled design for cost-efficient measurement of biomarkers." *Stat. Med.* **29**(5): 597–613.

Examples

```
# Load data frame with (g, Y, X, Xtilde, C) values for 4,996 pools and list
# of Xtilde values where 25 subjects have replicates. Xtilde values are
# affected by processing error and measurement error. True log-OR = 0.5,
# sigsq = 1, sigsq_p = 0.5, sigsq_m = 0.1.
data(dat_p_ndfa)
dat <- dat_p_ndfa$dat
```

```
reps <- dat_p_ndfa$reps

# Unobservable truth estimator - use precise X's
fit.unobservable <- p_ndfa(
  g = dat$g,
  y = dat$numcases,
  xtilde = dat$x,
  c = dat$c,
  errors = "neither"
)
fit.unobservable$estimates

# Naive estimator - use imprecise Xtilde's, but treat as precise
fit.naive <- p_ndfa(
  g = dat$g,
  y = dat$numcases,
  xtilde = dat$xtilde,
  c = dat$c,
  errors = "neither"
)
fit.naive$estimates

# Corrected estimator - use Xtilde's and account for errors (not using
# replicates here)
## Not run:
fit.noreps <- p_ndfa(
  g = dat$g,
  y = dat$numcases,
  xtilde = dat$xtilde,
  c = dat$c,
  errors = "both"
)
fit.noreps$estimates

# Corrected estimator - use Xtilde's including 25 replicates
fit.reps <- p_ndfa(
  g = dat$g,
  y = dat$numcases,
  xtilde = reps,
  c = dat$c,
  errors = "both"
)
fit.reps$estimates

# Same as previous, but allowing for non-constant odds ratio.
fit.nonconstant <- p_ndfa(
  g = dat$g,
  y = dat$numcases,
  xtilde = reps,
  c = dat$c,
  constant_or = FALSE,
  errors = "both"
)
```

```
fit.nonconstant$estimates

# Visualize estimated log-OR vs. X based on previous model fit
p <- plot_ndfa(
  estimates = fit.nonconstant$estimates,
  varcov = fit.nonconstant$theta.var,
  xrange = range(dat$xtilde[dat$g == 1]),
  cvals = mean(dat$c / dat$g)
)
p

# Likelihood ratio test for H0: odds ratio is constant.
test.constantOR <- p_ndfa(
  g = dat$g,
  y = dat$numcases,
  xtilde = reps,
  c = dat$c,
  constant_or = NULL,
  errors = "both"
)
test.constantOR$lrt

## End(Not run)
```

---

p_ndfa_constant *Normal Discriminant Function Approach for Estimating Odds Ratio with Exposure Measured in Pools and Potentially Subject to Additive Normal Errors (Constant Odds Ratio Version)*

Description

Assumes exposure given covariates and outcome is a normal-errors linear regression. Pooled exposure measurements can be assumed precise or subject to additive normal processing error and/or measurement error. Parameters are estimated using maximum likelihood.

Usage

```
p_ndfa_constant(g, y, xtilde, c = NULL, errors = "processing",
  start_nonvar_var = c(0.01, 1), lower_nonvar_var = c(-Inf, 1e-04),
  upper_nonvar_var = c(Inf, Inf), jitter_start = 0.01,
  nlminb_list = list(control = list(trace = 1, eval.max = 500, iter.max =
  500)), hessian_list = list(method.args = list(r = 4)),
  nlminb_object = NULL)
```

Arguments

g	Numeric vector of pool sizes, i.e. number of members in each pool.

y	Numeric vector of poolwise Y values (number of cases in each pool).
xtilde	Numeric vector (or list of numeric vectors, if some pools have replicates) with Xtilde values.
c	Numeric matrix with poolwise **C** values (if any), with one row for each pool. Can be a vector if there is only 1 covariate.
errors	Character string specifying the errors that X is subject to. Choices are `"neither"`, `"processing"` for processing error only, `"measurement"` for measurement error only, and `"both"`.
start_nonvar_var	Numeric vector of length 2 specifying starting value for non-variance terms and variance terms, respectively.
lower_nonvar_var	Numeric vector of length 2 specifying lower bound for non-variance terms and variance terms, respectively.
upper_nonvar_var	Numeric vector of length 2 specifying upper bound for non-variance terms and variance terms, respectively.
jitter_start	Numeric value specifying standard deviation for mean-0 normal jitters to add to starting values for a second try at maximizing the log-likelihood, should the initial call to `nlminb` result in non-convergence. Set to `NULL` for no second try.
nlminb_list	List of arguments to pass to `nlminb` for log-likelihood maximization.
hessian_list	List of arguments to pass to `hessian` for approximating the Hessian matrix. Only used if `estimate_var = TRUE`.
nlminb_object	Object returned from `nlminb` in a prior call. Useful for bypassing log-likelihood maximization if you just want to re-estimate the Hessian matrix with different options.

Value

List containing:

1. Numeric vector of parameter estimates.
2. Variance-covariance matrix.
3. Returned `nlminb` object from maximizing the log-likelihood function.
4. Akaike information criterion (AIC).

References

Lyles, R.H., Van Domelen, D.R., Mitchell, E.M. and Schisterman, E.F. (2015) "A discriminant function approach to adjust for processing and measurement error When a biomarker is assayed in pooled samples." *Int. J. Environ. Res. Public Health* **12**(11): 14723–14740.

Schisterman, E.F., Vexler, A., Mumford, S.L. and Perkins, N.J. (2010) "Hybrid pooled-unpooled design for cost-efficient measurement of biomarkers." *Stat. Med.* **29**(5): 597–613.

p_ndfa_nonconstant *Normal Discriminant Function Approach for Estimating Odds Ratio with Exposure Measured in Pools and Potentially Subject to Additive Normal Errors (Non-constant Odds Ratio Version)*

Description

Assumes exposure given covariates and outcome is a normal-errors linear regression. Pooled exposure measurements can be assumed precise or subject to additive normal processing error and/or measurement error. Parameters are estimated using maximum likelihood.

Usage

```
p_ndfa_nonconstant(g, y, xtilde, c = NULL, errors = "processing",
  start_nonvar_var = c(0.01, 1), lower_nonvar_var = c(-Inf, 1e-04),
  upper_nonvar_var = c(Inf, Inf), jitter_start = 0.01,
  nlminb_list = list(control = list(trace = 1, eval.max = 500, iter.max =
  500)), hessian_list = list(method.args = list(r = 4)),
  nlminb_object = NULL)
```

Arguments

g	Numeric vector of pool sizes, i.e. number of members in each pool.
y	Numeric vector of poolwise Y values (number of cases in each pool).
xtilde	Numeric vector (or list of numeric vectors, if some pools have replicates) with Xtilde values.
c	Numeric matrix with poolwise **C** values (if any), with one row for each pool. Can be a vector if there is only 1 covariate.
errors	Character string specifying the errors that X is subject to. Choices are "neither", "processing" for processing error only, "measurement" for measurement error only, and "both".
start_nonvar_var	Numeric vector of length 2 specifying starting value for non-variance terms and variance terms, respectively.
lower_nonvar_var	Numeric vector of length 2 specifying lower bound for non-variance terms and variance terms, respectively.
upper_nonvar_var	Numeric vector of length 2 specifying upper bound for non-variance terms and variance terms, respectively.
jitter_start	Numeric value specifying standard deviation for mean-0 normal jitters to add to starting values for a second try at maximizing the log-likelihood, should the initial call to nlminb result in non-convergence. Set to NULL for no second try.
nlminb_list	List of arguments to pass to nlminb for log-likelihood maximization.

`hessian_list` List of arguments to pass to `hessian` for approximating the Hessian matrix. Only used if `estimate_var = TRUE`.

`nlminb_object` Object returned from `nlminb` in a prior call. Useful for bypassing log-likelihood maximization if you just want to re-estimate the Hessian matrix with different options.

Value

List containing:

1. Numeric vector of parameter estimates.
2. Variance-covariance matrix.
3. Returned `nlminb` object from maximizing the log-likelihood function.
4. Akaike information criterion (AIC).

References

Lyles, R.H., Van Domelen, D.R., Mitchell, E.M. and Schisterman, E.F. (2015) "A discriminant function approach to adjust for processing and measurement error When a biomarker is assayed in pooled samples." *Int. J. Environ. Res. Public Health* **12**(11): 14723–14740.

Schisterman, E.F., Vexler, A., Mumford, S.L. and Perkins, N.J. (2010) "Hybrid pooled-unpooled design for cost-efficient measurement of biomarkers." *Stat. Med.* **29**(5): 597–613.

`simdata` *Dataset for a Paper Under Review*

Description

Simulated data intended to mimic the motivating example from a paper under review. Generated under GLR with true log-OR = 0.05.

Source

Simulated data in R.

test_pe *Test for Underestimated Processing Error Variance in Pooling Studies*

Description

In studies where a biomarker is measured in combined samples from multiple subjects rather than for each individual, design parameters (e.g. optimal pool size, sample size for 80% power) are very sensitive to the magnitude of processing errors. This function provides a test that can be used midway through data collection to test whether the processing error variance is larger than initially assumed, in which case the pool size may need to be adjusted.

Usage

```
test_pe(xtilde, g, sigsq, sigsq_m = 0, multiplicative = FALSE,
  mu = NULL, alpha = 0.05, boots = 1000, seed = NULL)
```

Arguments

xtilde	Numeric vector of pooled measurements.
g	Numeric value specifying the pool size.
sigsq	Numeric value specifying the variance of observations.
sigsq_m	Numeric value specifying the variance of measurement errors.
multiplicative	Logical value for whether to assume multiplicative rather than additive errors.
mu	Numeric value specifying the mean of observations. Only used if multiplicative = TRUE.
alpha	Numeric value specifying significance level for bootstrap confidence interval.
boots	Numeric value specifying the number of bootstrap samples to take.
seed	Numeric value specifying the random number seed, in case it is important to be able to reproduce the lower bound.

Details

The method is fully described in a manuscript currently under review. Briefly, the test of interest is H0: sigsq_p <= c, where sigsq_p is the processing error variance and c is the value assumed during study design. Under additive errors, a point estimate for sigsq_p is given by:

sigsq_p.hat = s2 - sigsq / g - sigsq_m

where s2 is the sample variance of poolwise measurements, g is the pool size, and sigsq_m is the measurement error variance which may be 0 if the assay is known to be precise.

Under multiplicative errors, the estimator is:

sigsq_p.hat = [(s2 - sigsq / g) / (mu^2 + sigsq / g) - sigsq_m] / (1 + sigsq_m).

In either case, bootstrapping can be used to obtain a lower bound for a one-sided confidence interval. If the lower bound is greater than c, H0 is rejected.

Value

List containing point estimate and lower bound of confidence interval.

Examples

```
# Generate data for hypothetical study designed assuming sigsq_p = 0.1, but
# truly sigsq_p = 0.25. Have data collected for 40 pools of size 5, and wish
# to test H0: sigsq_p <= 0.1. In this instance, a false negative occurs.
set.seed(123)
xtilde <- replicate(n = 40, expr = mean(rnorm(5)) + rnorm(n = 1, sd = sqrt(0.25)))
(fit <- test_pe(xtilde = xtilde, g = 5, sigsq = 1, sigsq_m = 0))
```

Index