Package 'pooling'

April 26, 2018

Type Package

Title Fit Poolwise Regression Models

Version 1.1.1

Date 2018-04-25

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

RoxygenNote 6.0.1

Imports cubature, dplyr, dvmisc, ggplot2, ggrepel, mvtnorm, pracma, stats

NeedsCompilation no

Repository CRAN

Date/Publication 2018-04-26 10:49:06 UTC

R topics documented:

`dat1` *Dataset with Simulated (Y, C) Values for Examples in dfa_xerrors and logreg_xerrors*

Description

Includes 5,000 observations. The `Xtilde` values are stored as a separate list called `dat1_xtilde`. The data were generated with a true log-odds ratio of 0.5 for `X` and `Y`, adjusted for `C`. The `Xtilde` measurements are subject to measurement error.

Source

Simulated data in R.

`dat1_xtilde` *Dataset with Simulated Xtilde Values for Examples in dfa_xerrors and logreg_xerrors*

Description

Includes 5,000 observations, 30 of which have replicates. The `(Y, C)` values are stored as a separate data frame called `dat1`. The data were generated with a true log-odds ratio of 0.5 for `X` and `Y`, adjusted for `C`. The `Xtilde` measurements are subject to measurement error.

Source

Simulated data in R.

dfa_xerrors *Discriminant Function Approach for Estimating Odds Ratio with Normal Exposure Subject to Measurement Error*

Description

Assumes exposure measurements are subject to additive normal measurement error, and exposure given covariates and outcome is a normal-errors linear regression. Some replicates are required for identifiability.

Usage

```
dfa_xerrors(y, xtilde, c = NULL, merror = TRUE, ...)
```

Arguments

y	Numeric vector of Y values.
xtilde	List of numeric vectors with Xtilde values.
c	Numeric matrix with C values (if any), with one row for each pool. Can be a vector if there is only 1 covariate.
merror	Logical value for whether there is measurement error.
...	Additional arguments to pass to nlminb.

Value

List containing:

1. Numeric vector with point estimates for gamma_y, sigsq, and the covariate-adjusted log-odds ratio, and the estimated variance for the log-odds ratio estimate if estimate_var = TRUE.
2. Object returned by nlminb containing information about likelihood maximization.
3. 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.

Examples

```
# Load datasets - dat1 has (Y, C) values and dat1_xtilde is list with 1 or 2
# Xtilde measurements for each subject. Xtilde values are affected by
# measurement error.
data(dat1)
data(dat1_xtilde)

# Estimate log-OR for X and Y adjusted for C, ignoring measurement error
```

```
fit1 <- dfa_xerrors(y = dat1$y, xtilde = dat1_xtilde, c = dat1$c, merror = FALSE)
fit1$estimates

# Repeat, but accounting for measurement error. Closer to true log-OR of 0.5.
fit2 <- dfa_xerrors(y = dat1$y, xtilde = dat1_xtilde, c = dat1$c, merror = TRUE)
fit2$estimates
```

---

gamma_constantscale *Fit Constant-Scale Gamma Model for Y vs. Covariates*

Description

Uses maximum likelihood to fit Y|**X** ~ Gamma(exp(beta_0 + **beta_x**^T **X**), b), with the shape-scale (as opposed to shape-rate) parameterization described in GammaDist.

Usage

```
gamma_constantscale(y, x = NULL, var = TRUE)
```

Arguments

y	Numeric vector.
x	Numeric vector or matrix. If NULL, model reduces to marginal Gamma model Y ~ Gamma(exp(beta_0), b).
var	Logical value for whether to return Hessian-based variance-covariance matrix.

Value

List of parameter estimates, variance-covariance matrix (if requested), AIC, and nlminb object.

Examples

```
# Generate data
set.seed(123)
x <- rnorm(1000)
y <- rgamma(1000, shape = exp(0.5 + 0.25 * x), scale = 0.25)

# Fit model
fit <- gamma_constantscale(y = y, x = x)
fit$theta.hat
fit$varcov
fit$aic

# Plot E(Y) vs. X according to model fit
plot(x, y, main = "Gamma Model for Y vs. X")
xvals <- seq(min(x), max(x), 0.01)
yvals <- exp(fit$theta.hat[1] + fit$theta.hat[2] * xvals) * fit$theta.hat[3]
```

```
points(xvals, yvals, type = "l")
```

`lognormal` *Fit Lognormal Regression for Y vs. Covariates*

Description

Uses maximum likelihood to fit Y|**X** ~ Lognormal(beta_0 + **beta_x**^T **X**, sigsq)

Usage

```
lognormal(y, x = NULL, var = TRUE)
```

Arguments

`y`	Numeric vector.
`x`	Numeric vector or matrix. If `NULL`, model reduces to marginal lognormal model Y ~ Lognormal(exp(beta_0), sigsq).
`var`	Logical value for whether to return Hessian-based variance-covariance matrix.

Value

List of parameter estimates, variance-covariance matrix (if requested), AIC, and `nlminb` object.

Examples

```
# Generate data
set.seed(123)
x <- rnorm(1000)
y <- rlnorm(1000, meanlog = 0.5 + 0.25 * x, sdlog = 0.5)

# Fit model
fit <- lognormal(y = y, x = x)
fit$theta.hat
fit$varcov
fit$aic

# Plot E(Y) vs. X according to model fit
plot(x, y, main = "Lognormal Model for Y vs. X")
xvals <- seq(min(x), max(x), 0.01)
yvals <- exp(fit$theta.hat[1] + fit$theta.hat[2] * xvals + fit$theta.hat[3] / 2)
points(xvals, yvals, type = "l")
```

logreg_xerrors *Logistic Regression with Normal Exposure Subject to Errors*

Description

Assumes exposure measurements are subject to additive normal measurement error, and exposure given covariates is a normal-errors linear regression. Some replicates are required for identifiability.

Usage

```
logreg_xerrors(y, xtilde, c = NULL, prev = NULL, samp_y1y0 = NULL,
  merror = TRUE, approx_integral = TRUE, integrate_tol = 1e-08,
  integrate_tol_start = integrate_tol,
  integrate_tol_hessian = integrate_tol, estimate_var = TRUE, ...)
```

Arguments

y	Numeric vector of Y values.
xtilde	List of numeric vectors with Xtilde values.
c	Numeric matrix with C values (if any), with one row for each pool. Can be a vector if there is only 1 covariate.
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.
merror	Logical value for whether there is measurement error.
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.
integrate_tol	Numeric value specifying the tol input to adaptIntegrate. Only used if approx_integral = FALSE.
integrate_tol_start	Same as integrate_tol, but applies only to the very first iteration of ML maximization. The first iteration tends to take much longer than subsequent ones, so less precise integration at the start can speed things up.
integrate_tol_hessian	Same as integrate_tol, but for use when estimating the Hessian matrix only. Sometimes more precise integration (i.e. smaller tolerance) than used for maximizing the likelihood 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.
...	Additional arguments to pass to nlminb.

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).

Examples

```
# Load dataset - dat1 has (Y, C) values and dat1_xtilde is list with 1 or 2
# Xtilde measurements for each subject.
data(dat1)
data(dat1_xtilde)

# Estimate log-OR for X and Y adjusted for C, ignoring measurement error
fit1 <- logreg_xerrors(y = dat1$y, xtilde = dat1_xtilde, c = dat1$c,
                       merror = FALSE)
fit1$theta.hat

# Repeat, but accounting for measurement error. Closer to true log-OR of 0.5.
fit2 <- logreg_xerrors(y = dat1$y, xtilde = dat1_xtilde, c = dat1$c,
                       merror = TRUE)
fit2$theta.hat
```

---

pdat1 *Dataset with Simulated (Y, Xtilde, C) Values for Examples in p_dfa_xerrors and p_logreg_xerrors*

Description

Includes 4,999 pooled observations, with a roughly equal number of pools of size 1, 2, and 3. The data were generated with a true log-odds ratio of 0.5 for `X` and `Y`, adjusted for `C`. The `Xtilde` measurements are subject to processing error.

Source

Simulated data in R.

pdat2 — *Dataset with Simulated (Y, Xtilde) Values for Examples in p_dfa_xerrors2 and p_logreg_xerrors2*

Description

Includes 248 pooled observations, with a roughly equal number of pools of size 1, 2, and 3. The individual-level `C` values are stored as a separate list called `pdat2_c`. The data were generated with a true log-odds ratio of 0.5 for `X` and `Y`, adjusted for `C`. The `Xtilde` measurements are subject to processing error.

Source

Simulated data in R.

pdat2_c — *Dataset with Simulated C Values for Examples in p_dfa_xerrors2 and p_logreg_xerrors2*

Description

Includes 248 sets of individual-level `C` values. The `(Y, Xtilde)` values are stored as a separate data frame called `pdat2`. The data were generated with a true log-odds ratio of 0.5 for `X` and `Y`, adjusted for `C`. The `Xtilde` measurements are subject to processing error.

Source

Simulated data in R.

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

Description

When `p_dfa_xerrors` is fit with `constant_or = FALSE`, the log-odds ratio for `X` depends on the value of `X`. This function plots the log-odds ratio vs. `X`.

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

When p_dfa_xerrors2 is fit with constant_or = FALSE, the log-odds ratio for X depends on the value of X (and covariates, if there are any). This function plots the log-odds ratio vs. X for one or several sets of covariate values.

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_1, 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)
data(pdat2_c)
fit <- p_dfa_xerrors2(g = pdat2$g, y = pdat2$y, xtilde = pdat2$xtilde,
                      c = pdat2_c, 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(pdat2$xtilde / pdat2$g),
               cvals = mean(unlist(pdat2_c)))
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 = 0.2, sigsq = 1, sigsq_p = 0, sigsq_m = 0,
  multiplicative = FALSE, mu = 1, alpha = 0.05, beta = 0.2,
  type = "two.sample", 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.
sigsq	Numeric value specifying the variance of observations.
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.
mu	Numeric value specifying the larger of the two suspected means. Only used if `multiplicative = TRUE`.
alpha	Numeric value specifying significance level.
beta	Numeric value specifying beta = 1 - power.
type	Character string specifying type of t-test. Choices are `"two.sample"`, `"one.sample"`, and `"paired"`.
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 study costs vs. pool size with default settings
poolcost_t()

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

---

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:	1.1.1
Date:	2018-04-25
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 = 0.5, sigsq = 1, sigsq_p = 0,
  sigsq_m = 0, multiplicative = FALSE, mu = 1, alpha = 0.05,
  beta = 0.2, type = "two.sample", 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.
`sigsq`	Numeric value specifying the variance of observations.
`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.
`mu`	Numeric value specifying the larger of the two suspected means. Only used if `multiplicative = TRUE`.
`alpha`	Numeric value specifying significance level.
`beta`	Numeric value specifying beta = 1 - power.

type	Character string specifying type of t-test. Choices are "two.sample", "one.sample", and "paired".
assay_cost	Numeric value specifying cost of each assay.
other_costs	Numeric value specifying other per-subject costs.
labels	Logical value.
ylim	Numeric vector.
...	Arguments to pass to power.t.test.

Value

Plot of power vs. total costs generated by ggplot.

Examples

```
# Power for two-sample t-test with d = 0.5, var = 1, and no "other" costs
# per subject
poolpower_t(d = 0.5, sigsq = 1)

# Repeat but for other costs per subject equal to 1/4 the assay cost
poolpower_t(d = 0.5, sigsq = 1, other_costs = 1/4)

# Back to no other costs, but with processing and measurement error
poolpower_t(d = 0.5, sigsq = 1, sigsq_p = 0.2, sigsq_m = 0.1)
```

---

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, sigsq = 1, sigsq_p = 0, sigsq_m = 0,
  multiplicative = FALSE, mu = 1, type = "two.sample", assay_cost = 100,
  other_costs = 0, labels = TRUE, ylim = NULL)
```

Arguments

g	Numeric vector of pool sizes to include.
sigsq	Numeric value specifying the variance of observations.
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.
mu	Numeric value specifying the larger of the two suspected means. Only used if multiplicative = TRUE.
type	Character string specifying type of t-test. Choices are "two.sample", "one.sample", and "paired".
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()

# Add processing error and other per-subject costs
poolvar_t(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 Subject to Errors*

Description

Assumes exposure measurements are subject to additive normal processing error and measurement error, and exposure given covariates and outcome is a normal-errors linear regression.

Usage

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

Arguments

g	Numeric vector with 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, objects returned by nlminb, and AICs, for one or two models depending on constant_or.

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: 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 Subject to Errors*

Description

Assumes exposure measurements are subject to multiplicative lognormal processing error and measurement error, and exposure given covariates and outcome is a Gamma regression.

Usage

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

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.
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 adaptIntegrate.
integrate_tol_start	Same as integrate_tol, but applies only to the very first iteration of ML maximization. The first iteration tends to take much longer than subsequent ones, so less precise integration at the start can speed things up.
integrate_tol_hessian	Same as integrate_tol, but for use when estimating the Hessian matrix only. Sometimes more precise integration (i.e. smaller tolerance) than used for maximizing the likelihood 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.
...	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 datasets - pdat2 has poolwise (Y, Xtilde) values and pdat2_c has
# individual-level C values. Xtilde values are affected by processing error.
data(pdat2)
data(pdat2_c)

# Estimate log-OR for X and Y adjusted for C, ignoring processing error
fit1 <- p_dfa_xerrors2(g = pdat2$g, y = pdat2$y, xtilde = pdat2$xtilde,
                       c = pdat2_c, errors = "neither")
fit1$estimates

# Repeat, but accounting for processing error. Takes about 5 minutes to run
# due to numerical integration. Gives log-OR closer to true value of 0.5.
# fit2 <- p_dfa_xerrors2(g = pdat2$g, y = pdat2$y, xtilde = pdat2$xtilde,
#                        c = pdat2_c, errors = "processing",
#                        control = list(trace = 1))
# fit2$estimates
```

---

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, ...)
```

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.
...	Additional arguments to pass to nlminb.

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 = "both",
  nondiff_pe = TRUE, nondiff_me = TRUE, constant_pe = TRUE, prev = NULL,
  samp_y1y0 = NULL, approx_integral = TRUE, integrate_tol = 1e-08,
  integrate_tol_start = integrate_tol,
  integrate_tol_hessian = integrate_tol, estimate_var = TRUE, ...)
```

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.

`integrate_tol` Numeric value specifying the `tol` input to `adaptIntegrate`. Only used if `approx_integral = FALSE`.

`integrate_tol_start` Same as `integrate_tol`, but applies only to the very first iteration of ML maximization. The first iteration tends to take much longer than subsequent ones, so less precise integration at the start can speed things up.

`integrate_tol_hessian` Same as `integrate_tol`, but for use when estimating the Hessian matrix only. Sometimes more precise integration (i.e. smaller tolerance) than used for maximizing the likelihood 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.

`...` Additional arguments to pass to `nlminb`.

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 = "both",
  nondiff_pe = TRUE, nondiff_me = TRUE, constant_pe = TRUE, prev = NULL,
  samp_y1y0 = NULL, integrate_tol = 1e-08,
  integrate_tol_start = integrate_tol,
  integrate_tol_hessian = integrate_tol, estimate_var = TRUE, ...)
```

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.

`integrate_tol` Numeric value specifying the `tol` input to `adaptIntegrate`.

`integrate_tol_start` Same as `integrate_tol`, but applies only to the very first iteration of ML maximization. The first iteration tends to take much longer than subsequent ones, so less precise integration at the start can speed things up.

`integrate_tol_hessian` Same as `integrate_tol`, but for use when estimating the Hessian matrix only. Sometimes more precise integration (i.e. smaller tolerance) than used for maximizing the likelihood 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.

`...` Additional arguments to pass to `nlminb`.

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 datasets - pdat2 has poolwise (Y, Xtilde) values and pdat2_c has
# individual-level C values. Xtilde values are affected by processing error.
data(pdat2)
data(pdat2_c)

# Estimate log-OR for X and Y adjusted for C, ignoring processing error
fit1 <- p_logreg_xerrors2(g = pdat2$g, y = pdat2$y, xtilde = pdat2$xtilde,
                          c = pdat2_c, errors = "neither")
fit1$theta.hat

# Repeat, but accounting for processing error. Takes about 1 minute to run
# due to numerical integration. Gives log-OR closer to true value of 0.5.
# fit2 <- p_logreg_xerrors2(g = pdat2$g, y = pdat2$y, xtilde = pdat2$xtilde,
#                           c = pdat2_c, errors = "processing",
#                           control = list(trace = 1))
# fit2$theta.hat
```

`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