Package: ppwdeming (via r-universe)

July 23, 2026

Version 3.0.2

Date 2026-7-23

Title Precision Profile Weighted Deming Regression

Author Douglas M. Hawkins [aut, cph], Jessica J. Kraker [aut, cre]

Maintainer Jessica J. Kraker <krakerjj@uwec.edu>

Depends R (>= 3.6.0)

Suggests testthat (>= 3.0.0), lifecycle (>= 1.0.5)

Imports stats

Description Weighted Deming regression, also known as 'errors-in-variable' regression, is applied with suitable weights. Weights are modeled via a precision profile; thus the methods implemented here are referred to as precision profile weighted Deming (PWD) regression. The package covers two settings – one where the precision profiles are known either from external studies or from adequate replication of the X and Y readings, and one in which there is a plausible functional form for the precision profiles but the exact (unknown) function must be estimated from the (generally singlicate) readings. The function set includes tools for: estimated standard errors (via jackknifing); standardized-residual analysis function with regression diagnostic tools for normality, linearity and constant variance; and an outlier analysis identifying significant outliers for closer investigation. The following reference provides further information on mathematical derivations and applications. Hawkins, D.M., and J.J. Kraker (2026). 'Precision Profile Weighted Deming Regression for Methods Comparison'. The Journal of Applied Laboratory Medicine 11, 379-392 <doi:10.1093/jalm/jfaf183>. Weighted Deming regression is also now implemented for multiple instruments , as set out in Hawkins, D.M., and J.J. Kraker (2026). 'Multiple Instrument Methods Comparison by Precision weighted Deming Regression', on Arxiv <doi:10.48550/arXiv.2607.11776>. The "multi" functions refer to the multiple instrument analysis, and the ``PWD"

functions refer to the two-instrument.

License GPL (>= 3)

Encoding UTF-8

Config/testthat/edition 3

Config/roxygen2/version 8.0.0

NeedsCompilation no

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

Date/Publication 2026-07-23 14:39:40 UTC

RemoteUrl https://github.com/cran/ppwdeming

RemoteRef HEAD

RemoteSha f77aeea934f8413c9e0f9cb4225a1a56bdc1350e

Contents

`multi_PWD` *Multiple Instrument Weighted Deming Regression*

Description

This code estimates the Rocke-Lorenzato precision profile parameters and the regression of each instrument's values on the underlying latent true concentration.

Usage

```
multi_PWD(X, names, refine=1)
```

Arguments

X	the *n* by *I* matrix of values reported by the *I* instruments on the *n* samples.
names	*optional* (default of NA) - labels of the *I* instruments.
refine	*optional* (default of 1) - the number of resubstitutions used to refine lambda estimates.

Details

The function optimizes the likelihood over the parameter rho, the ratio of the Rocke-Lorenzato parameters sigma to kappa: $\rho = \frac{\sigma}{\kappa}$. Calls multi_PWD_inner to provide the estimates of the other parameters.

Value

A list containing the following components:

alpha	the vector of *I* estimated intercepts.
beta	the vector of *I* estimated slopes.
fity	the *n* by *I* matrix of fitted values.
mu	the vector of *n* estimated true concentrations.
resi	the *n* by *I* matrix of residuals.
rho	the estimate of ρ.
sigma	the estimate of σ.
kappa	the estimate of κ.
lambda	the vector of *I* estimated λ_i.
L	the -2 log likelihood
scalr	the *n* by *I* matrix of scaled residuals.
whichmissing	a logical vector identifying locations of row of X with missing values
results	a printer-ready data frame of the fitted regression
names	labels of the *I* instruments.

Author(s)

Douglas M. Hawkins, Jessica J. Kraker krakerjj@uwec.edu

References

Hawkins DM and Kraker JJ. Multiple Instrument Methods Comparison by Precision weighted Deming Regression, on *Arxiv* (2026) doi:10.48550/arXiv.2607.11776

.

Examples

```
# library
library(ppwdeming)

# parameter specifications
n  <- 100
I  <- 4
alpha  <- c(0  , 0  , 2  , -2 )
beta   <- c(0.9, 1.1, 1.2, 0.8)
names   <- c("Inst_1", "Inst_2", "Inst_3", "Inst_4")
lambda <- c(1.8, 0.45, 0.9, 0.9)
rutlam <- sqrt(lambda)
sigma  <- 2
kappa  <- 0.08
true   <- 8*10^((0:(n-1))/(n-1))

# simulated data
set.seed(1039)
X     <- NULL
for (i in 1:I)  {
  truey <- alpha[i] + beta[i]*true
  X <- cbind(X, truey+rutlam[i]*(sigma*rnorm(n) + truey*kappa*rnorm(n)))
}

# fit with RL precision profile to estimate parameters
fit <- multi_PWD(X, names=names)

# results
summary(fit)
```

`multi_PWD_inf` *Jack-knife inference on the parameters of the Multiple Instrument fit*

Description

This code uses jackknifing of the Multiple-Instrument fit to get standard errors of the regression parameters and lambda values.

Usage

```
multi_PWD_inf(X, names=NA, refine=1)
```

Arguments

`X`	the *n* by *I* matrix of values reported by the *I* instruments on the *n* samples.
`names`	*optional* (default of NA) - labels of the *I* instruments.
`refine`	*optional* (default of 1) - the number of re-substitutions used to refine lambda estimates.

Details

Uses jackknifing to get standard errors for the slopes `beta`, intercepts `alpha`, and `lambda` values of the *I* instruments.

Value

A list containing the following components:

`L`	the -2 log likelihood of the full-sample fit.
`rho`	the full-sample estimate of ρ.
`kappa`	the full-sample estimate of κ.
`lambda`	the vector of *I* estimated λ_i.
`lambdase`	the vector of *I* standard errors of the estimated λ_i.
`alpha`	the vector of *I* estimated intercepts.
`alphase`	the vector of *I* standard errors of the estimated intercepts.
`beta`	the vector of *I* estimated slopes.
`betase`	the vector of *I* standard errors of the estimated slopes.
`mu`	the vector of *n* estimated true concentrations.
`fity`	the *n* by *I* matrix of fitted values.
`resi`	the *n* by *I* matrix of residuals.
`whichmissing`	a logical vector identifying locations of row of X with missing values
`scalr`	the *n* by *I* matrix of scaled residuals.
`names`	labels of the *I* instruments.

Author(s)

Douglas M. Hawkins, Jessica J. Kraker krakerjj@uwec.edu

References

Hawkins DM and Kraker JJ. Multiple Instrument Methods Comparison by Precision weighted Deming Regression, on *Arxiv* (2026) doi:10.48550/arXiv.2607.11776

Examples

```
# library
library(ppwdeming)

# parameter specifications
n  <- 100
I  <- 4
alpha  <- c(0  , 0  , 2  , -2 )
beta   <- c(0.9, 1.1, 1.2, 0.8)
names   <- c("Inst_1", "Inst_2", "Inst_3", "Inst_4")
lambda <- c(1.8, 0.45, 0.9, 0.9)
rutlam <- sqrt(lambda)
```

```
sigma  <- 2
kappa  <- 0.08
true   <- 8*10^((0:(n-1))/(n-1))

# simulated data
set.seed(1039)
X     <- NULL
for (i in 1:I)  {
  truey <- alpha[i] + beta[i]*true
  X <- cbind(X, truey+rutlam[i]*(sigma*rnorm(n) + truey*kappa*rnorm(n)))
}

# application
inf <- multi_PWD_inf(X, names=names)
inf$alpha; inf$beta

# summary
summary(inf)
```

`multi_PWD_inner` *Multiple Instrument Parameter Estimation from rho*

Description

This is primarily a workhorse routine that `multi_PWD` calls and, given a value of `rho`, estimates the other model parameters.

Usage

```
multi_PWD_inner(X, rho, refine=1, eps=1e-9)
```

Arguments

`X`	the *n* by *I* matrix of values reported by the *I* instruments on the *n* samples.
`rho`	the input value of $\rho = \frac{\sigma}{\kappa}$.
`refine`	*optional* (default of 1) - the number of re-substitutions used to refine lambda estimates.
`eps`	*optional* (default of 1e-9) - a convergence criterion.

Details

Using the specified ρ, the routine calculates the estimates of the regression parameters and λ.

Value

A list containing the following components:

`alpha`	the vector of *I* estimated intercepts α_i.
`beta`	the vector of *I* estimated slopes β_i.
`fity`	the *n* by *I* matrix of fitted values.
`mu`	the vector of *n* estimated true concentrations.
`resi`	the *n* by *I* matrix of residuals.
`rho`	the input value of ρ.
`sigma`	the estimate of σ.
`kappa`	the estimate of κ.
`lambda`	the vector of *I* estimated λ_i.
`L`	the -2 log likelihood
`scalr`	the *n* by *I* matrix of scaled residuals.

Author(s)

Douglas M. Hawkins, Jessica J. Kraker krakerjj@uwec.edu

References

Hawkins DM and Kraker JJ. Multiple Instrument Methods Comparison by Precision weighted Deming Regression, on *Arxiv* (2026) doi:10.48550/arXiv.2607.11776

.

Examples

```
# library
library(ppwdeming)

# parameter specifications
n  <- 100
I  <- 4
alpha  <- c(0  , 0  , 2  , -2 )
beta   <- c(0.9, 1.1, 1.2, 0.8)
names   <- c("Inst_1", "Inst_2", "Inst_3", "Inst_4")
lambda <- c(1.8, 0.45, 0.9, 0.9)
rutlam <- sqrt(lambda)
sigma  <- 2
kappa  <- 0.08
true   <- 8*10^((0:(n-1))/(n-1))

# simulated data
set.seed(1039)
X     <- NULL
for (i in 1:I)  {
  truey <- alpha[i] + beta[i]*true
```

```
  X <- cbind(X, truey+rutlam[i]*(sigma*rnorm(n) + truey*kappa*rnorm(n)))
}
```

`multi_PWD_out` *Locate and test for outliers of the fit*

Description

The routine uses a conventional multivariate outlier test statistic set, in the framework of Rosner's multiple outlier search methodology, for the Multiple-Instrument fit. The vectors tested as multivariate outliers are the vectors of scaled residuals of each case.

Usage

```
multi_PWD_out(X, K=NA, names=NA, refine=1, Pcut=0.01)
```

Arguments

`X`	the *n* by *I* matrix of values reported by the *I* instruments on the *n* samples.
`K`	*optional* (default of n/20) the number of cases to seek in the forward outlier screen.
`names`	*optional* (default of NA) - labels of the *I* instruments.
`refine`	*optional* (default of 1) - the number of re-substitutions used to refine lambda estimates.
`Pcut`	*optional* (default of 0.01) the Bonferroni significance level for the outlier test.

Details

A forward selection step identifies the *K* sequentially most extreme cases using scaled residuals from a sequence of fits. This is done by finding the Mahalanobis distance of each case's vector of scaled residuals from the mean vector of all cases currently thought to be "clean". The case with the largest such distance is moved from the "clean" list to the "suspect" list and the model refitted using the new "clean" list. This is repeated until `K` cases are on the suspect list.

Backward re-inclusion follows. The model is fitted to the clean cases, and the Hotelling T^2 statistic between each suspect and the set of clean cases is computed. The case with the minimum T^2 is found and its Bonferroni P value computed. If it is not significant, the case is moved from the suspect back to the clean list. The process is repeated until either all suspects have been exonerated or a set of statistically significant outliers remains.

Value

A list containing the following components:

`ndrop`	the number of statistically significant outliers.
`drop`	the identities of the significant outliers.

initlis	the K vector of cases identified as potential outliers in forward selection.
Dilis	the K vector of Mahalanobis distances of each initial suspect.
Tilis	the K vector of transforms of the Dilis to Hotelling T^2.
backlis	the vector of cases re-included in the backward step.
backP	their outlier P values.
bonfP	The Bonferroni P values of the remaining suspect cases.
forward	a print-ready listing of the forward selection steps.
backward	a print-ready listing of the backward re-inclusion steps.
outlis	a print-ready list of the significant outliers' scaled residuals and Bonferroni P values.

Author(s)

Douglas M. Hawkins, Jessica J. Kraker krakerjj@uwec.edu

References

Hawkins DM and Kraker JJ. Multiple Instrument Methods Comparison by Precision weighted Deming Regression, on *Arxiv* (2026) doi:10.48550/arXiv.2607.11776

Examples

```
# library
library(ppwdeming)

# parameter specifications
n  <- 100
I  <- 4
alpha  <- c(0  , 0  , 2  , -2 )
beta   <- c(0.9, 1.1, 1.2, 0.8)
names   <- c("Inst_1", "Inst_2", "Inst_3", "Inst_4")
lambda <- c(1.8, 0.45, 0.9, 0.9)
rutlam <- sqrt(lambda)
sigma  <- 2
kappa  <- 0.08
true   <- 8*10^((0:(n-1))/(n-1))

# simulated data
set.seed(1039)
X     <- NULL
for (i in 1:I)  {
  truey <- alpha[i] + beta[i]*true
  X <- cbind(X, truey+rutlam[i]*(sigma*rnorm(n) + truey*kappa*rnorm(n)))
}

# add outliers
X[1,4] <- X[1,4] + 15
X[n,2] <- X[n,2] / 1.5
```

```
# application
out <- multi_PWD_out(X, K=4, names=names)

# full summary of process
summary(out)
```

PWD_get_gh *Estimate of Variance Profile Functions (proportional)*

Description

This code estimates the variance profiles, assumed proportional, of the Rocke-Lorenzato form; also provides the resulting weighted Deming fit and residuals.

Usage

```
PWD_get_gh(X, Y, lambda = 1,
           rho=lifecycle::deprecated(),
           alpha=lifecycle::deprecated(), beta=lifecycle::deprecated(),
           mu=lifecycle::deprecated(),
           quad=FALSE, epsilon = 1e-8,
           printem=lifecycle::deprecated())
```

Arguments

X	the vector of predicate readings.
Y	the vector of test readings.
lambda	*optional* (default of 1) - the ratio of the X to the Y precision profile.
rho	**[Deprecated]** rho = numvalue initialization is no longer implemented in algorithm.
alpha	**[Deprecated]** alpha = numvalue initialization is no longer implemented in algorithm.
beta	**[Deprecated]** beta = numvalue initialization is no longer implemented in algorithm.
mu	**[Deprecated]** mu = numvector initialization is no longer implemented in algorithm.
quad	*optional* (default of FALSE) - logical, selects fitting a linear or a quadratic regression.
epsilon	*optional* (default of 1.e-8) - convergence tolerance limit.
printem	**[Deprecated]** printem = TRUE is no longer supported; separate summary functions are provided for printing output.

Details

This workhorse routine optimizes the likelihood in the **unknown** g, h setting over its n+4 or n+5 parameters: the two Rocke-Lorenzato precision profile parameters σ and κ, the intercept α and slope β (and coefficien of the squared term γ, if quadratic), and the n latent true concentrations μ_i.

That is, the assumed forms are:

- predicate precision profile model: $g_i = var(X_i) = \lambda \left(\sigma^2 + [\kappa \cdot \mu_i]^2\right)$ and
- test precision profile model: $h_i = var(Y_i) = \sigma^2 + [\kappa \cdot (\alpha + \beta\mu_i)]^2$.

The search algorithm implements an efficient approach via reparameterization to the ratio $\rho = \frac{\sigma}{\kappa}$.

Value

A list containing the following components:

`alpha`	the fitted intercept
`beta`	the fitted slope
`gamma`	the fitted coefficient of the square: if quad is FALSE, the value is zero
`quad`	logical, FALSE for linear or TRUE for quadratic regression
`fity`	the vector of predicted Y
`mu`	the vector of estimated latent true values
`resi`	the vector of residuals
`sigma`	the estimate of the Rocke-Lorenzato σ
`kappa`	the estimate of the Rocke-Lorenzato κ
`L`	the -2 log likelihood L

Author(s)

Douglas M. Hawkins, Jessica J. Kraker krakerjj@uwec.edu

References

Hawkins DM and Kraker JJ (2026). Precision Profile Weighted Deming Regression for Methods Comparison. *The Journal of Applied Laboratory Medicine*, **11**(2), 379-392. doi:10.1093/jalm/jfaf183

Rocke DM, Lorenzato S (2012). A Two-Component Model for Measurement Error in Analytical Chemistry. *Technometrics*, **37:2**:176-184.

Examples

```
# library
library(ppwdeming)

# parameter specifications
n <- 100
```

```
quad  <- FALSE   # for a linear fit
quad  <- TRUE    # for a quadratic fit
sigma <- 1
kappa <- 0.08
alpha <- 1
beta  <- 1.1
gamma <- 0
if (quad) gamma <- -0.005
true  <- 8*10^((0:(n-1))/(n-1))
truey <- alpha+beta*true+gamma*true^2
# simulate single sample - set seed for reproducibility
set.seed(1039)
# specifications for predicate method
X     <- sigma*rnorm(100)+true *(1+kappa*rnorm(n))
# specifications for test method
Y     <- sigma*rnorm(100)+truey*(1+kappa*rnorm(n))

# fit with RL precision profile to estimate parameters
RL_gh_fit  <- PWD_get_gh(X,Y, quad=quad)

# results
summary(RL_gh_fit)
```

`PWD_inference` *Precision-Profile Weighted Deming Regression – Inference*

Description

This routine fits the regression and uses the jackknife to get its precision. Implements Rocke-Lorenzato as the variance profile model.

Usage

```
PWD_inference(X, Y, lambda=1,
              rho=lifecycle::deprecated(),
              alpha=lifecycle::deprecated(), beta=lifecycle::deprecated(),
              mu=lifecycle::deprecated(),
              MDL=NA,
              quad=FALSE, epsilon=1e-8,
              printem=lifecycle::deprecated())
```

Arguments

`X`	the vector of predicate readings.
`Y`	the vector of test readings.
`lambda`	*optional* (default of 1) - the ratio of the `X` to the `Y` precision profile.

rho	**[Deprecated]** rho = numvalue initialization is no longer implemented in algorithm.
alpha	**[Deprecated]** alpha = numvalue initialization is no longer implemented in algorithm.
beta	**[Deprecated]** beta = numvalue initialization is no longer implemented in algorithm.
mu	**[Deprecated]** mu = numvector initialization is no longer implemented in algorithm.
MDL	*optional* (default to missing) - medical decision level(s).
quad	*optional* (default of FALSE) - logical, selects fitting a linear or a quadratic regression.
epsilon	*optional* (default of 1.e-8) - convergence tolerance limit.
printem	**[Deprecated]** printem = TRUE is no longer supported; separate summary functions are provided for printing output.

Details

For the linear model relating the predicate and test readings, the standard errors of the estimators $\hat{\alpha}$, $\hat{\beta}$, (and $\hat{\gamma}$, if quadratic) and their covariance are estimated by the jackknife. The point estimates of the intercept and slope are output, along with their standard errors and covariance.

These estimates are further used to estimate the predictions at the input MDL, if appropriate.

Value

A list containing the following components:

alpha	the fitted intercept
beta	the fitted slope
gamma	the fitted coefficient of the square: if quad is FALSE, the value is zero
quad	logical, FALSE for linear or TRUE for quadratic regression
cor	the Pearson correlation between X and Y
fity	the vector of predicted Y
mu	the vector of estimated latent true values
resi	the vector of residuals
preresi	the vector of leave-one-out predicted residuals
sigma	the estimate of the Rocke-Lorenzato σ
kappa	the estimate of the Rocke-Lorenzato κ
L	the -2 log likelihood L
sealpha	the jackknife standard error of $alpha$
sebeta	the jackknife standard error of $beta$
segamma	the jackknife standard error of $gamma$
covar	the jackknife covariance between alpha and beta (and gamma, if quadratic)

CIalpha	the level 95% confidence-interval estimate of α
CIbeta	the level 95% confidence-interval estimate of β
CIgamma	the level 95% confidence-interval estimate of γ
preMDL	the predictions at the MDL(s)
sepreMDL	the jackknife standard error of predictions at the MDL
preMDLl	the lower confidence limit(s) of preMDL
preMDLu	the upper confidence limit(s) of preMDL
whichmissing	a logical vector identifying locations of missing X or Y values
MDL	vector of medical decision level(s)

Author(s)

Douglas M. Hawkins, Jessica J. Kraker krakerjj@uwec.edu

References

Hawkins DM and Kraker JJ (2026). Precision Profile Weighted Deming Regression for Methods Comparison. *The Journal of Applied Laboratory Medicine*, **11**(2), 379-392. doi:10.1093/jalm/jfaf183

Efron, B (1982). The jackknife, the bootstrap and other resampling plans. Society for Industrial and Applied Mathematics.

Examples

```
# library
library(ppwdeming)

# parameter specifications
n <- 100

sigma <- 1
kappa <- 0.08
alpha <- 1
beta  <- 1.1
quad  <- FALSE # or
#quad  <- TRUE
gamma <- 0
if (quad) gamma <- -0.005
true  <- 8*10^((0:(n-1))/(n-1))
truey <- alpha+beta*true+gamma*true^2
# simulate single sample - set seed for reproducibility
set.seed(1039)
# specifications for predicate method
X     <- sigma*rnorm(100)+true *(1+kappa*rnorm(n))
# specifications for test method
Y     <- sigma*rnorm(100)+truey*(1+kappa*rnorm(n))

# fit with RL precision profile to estimate parameters and variability
```

```
RL_inf <- PWD_inference(X,Y,MDL=12, quad=quad)

# summary of results
summary(RL_inf)
```

PWD_known *Weighted Deming Regression – general weights*

Description

This code is used for the setting of known precision profiles implemented in user-provided R functions called gfun and hfun.

Usage

```
PWD_known(X, Y, gfun, hfun, gparms, hparms, epsilon=1e-8,
          MDL=NA, getCI=TRUE,
          printem=lifecycle::deprecated())
```

Arguments

X	the vector of predicate readings.
Y	the vector of test readings.
gfun	a function with two arguments, a vector of size *n* and a vector of parameters.
hfun	a function with two arguments, a vector of size *n* and a vector of parameters.
gparms	a numeric vector containing any parameters referenced by gfun.
hparms	a numeric vector containing any parameters referenced by hfun.
epsilon	*optional* (default of 1.e-8) - convergence tolerance limit.
MDL	*optional* (default of NA) - medical decision level(s).
getCI	*optional* (default of TRUE) - allows for jackknifed standard errors on the regression and MDL.
printem	**[Deprecated]** printem = TRUE is no longer supported; separate summary functions are provided for printing output.

Details

The functions gfun and hfun are allowed as inputs, to support flexibility in specification of the forms of these variance functions. The known precision profiles specified by the functions gfun and hfun, when provided with estimated vectors of μ and $\alpha + \beta\mu$ respectively and with any required parameters, will produce the vectors g and h. These vectors are then integrated into the iterative estimation of the slope and intercept of the linear relationship between predicate and test readings.

Value

A list containing the following components:

alpha	the fitted intercept
beta	the fitted slope
cor	the Pearson correlation between X and Y
fity	the vector of predicted Y
mu	the vector of estimated latent true values
resi	the vector of residuals
scalr	the vector of scaled residuals using the specified g and h
L	the -2 log likelihood L
sealpha	the jackknife standard error of alpha
sebeta	the jackknife standard error of beta
covar	the jackknife covariance between alpha and beta
preMDL	the predictions at the MDL(s)
sepreMDL	the jackknife standard error of predictions at the MDL
preMDLl	the lower confidence limit(s) of preMDL
preMDLu	the upper confidence limit(s) of preMDL
whichmissing	a logical vector identifying locations of missing X or Y values
MDL	vector of medical decision level(s)

Author(s)

Douglas M. Hawkins, Jessica J. Kraker krakerjj@uwec.edu

Examples

```
# library
library(ppwdeming)

# parameter specifications
n <- 100

alpha <- 1
beta  <- 1.1
true  <- 8*10^((0:(n-1))/(n-1))
truey <- alpha+beta*true
# forms of precision profiles
gfun    <- function(true, gparms) {
  gvals = gparms[1]+gparms[2]*true^gparms[3]
  gvals
}
hfun    <- function(true, hparms) {
  hvals = hparms[1]+hparms[2]*true^hparms[3]
  hvals
```

```
}

# Loosely motivated by Vitamin D data set
g     <- 4e-16+0.07*true^1.27
h     <- 6e-2+7e-5*truey^2.2
# simulate single sample - set seed for reproducibility
set.seed(1039)
# specifications for predicate method
X     <- true +sqrt(g)*rnorm(n)
# specifications for test method
Y     <- truey+sqrt(h)*rnorm(n)

# fit with to estimate linear parameters
pwd_known_fit <- PWD_known(X, Y, gfun, hfun,
                           gparms=c(4e-16, 0.07, 1.27),
                           hparms=c(6e-2, 7e-5, 2.2), MDL=12)
# display results
summary(pwd_known_fit)
```

PWD_outlier *Weighted Deming Regression – Outlier scanning*

Description

This function tests for outliers from the fitted regression, and refits on a sanitized data set (with outliers removed).

Usage

```
PWD_outlier(X, Y, K, lambda=1, Pcut=0.01,
            rho=lifecycle::deprecated(),
            alpha=lifecycle::deprecated(), beta=lifecycle::deprecated(),
            mu=lifecycle::deprecated(),
            quad=FALSE)
```

Arguments

X	the vector of predicate readings.
Y	the vector of test readings.
K	the maximum number of outliers to seek.
lambda	*optional* (default of 1) - the ratio of the X to the Y precision profile.
Pcut	*optional*, default 0.01 (1%), cutoff for statistical significance of Bonferroni P.
rho	**[Deprecated]** rho = numvalue initialization is no longer implemented in algorithm.
alpha	**[Deprecated]** alpha = numvalue initialization is no longer implemented in algorithm.

beta	**[Deprecated]** beta = numvalue initialization is no longer implemented in algorithm.
mu	**[Deprecated]** mu = numvector initialization is no longer implemented in algorithm.
quad	*optional* (default of FALSE) - logical, selects fitting a linear or a quadratic regression.

Details

The method is modeled on the Rosner sequential ESD outlier procedure and assumes the sample is large enough to ignore the effect of random variability in the parameter estimates on the distribution of the residuals.

Value

A list containing the following components:

ndrop	the number of significant outliers
drop	a vector of the indices of the outliers
cor	the Pearson correlation between X and Y
cleancor	the Pearson correlation between cleaned X and Y (after outliers removed)
scalr	the scaled residuals of all cases from the sanitized fit and whose normal tail areas provide the basis for the outlier P values
basepar	the sigma, kappa, alpha, beta, gamma of the full data set
lastpar	the sigma, kappa, alpha, beta, gamma of the sanitized data set
forward	dataframe summarizing the forward identification of possible outliers
backward	dataframe summarizing the backward reinclusion of cases
tee	the t statistics of the final identified outliers
BonP	the Bonferroni P-value of the final identified outliers
outlis	dataframe containing the outlier cases, test statistics, and P-values

Author(s)

Douglas M. Hawkins, Jessica J. Kraker krakerjj@uwec.edu

References

Hawkins DM and Kraker JJ (2026). Precision Profile Weighted Deming Regression for Methods Comparison. *The Journal of Applied Laboratory Medicine*, **11**(2), 379-392. doi:10.1093/jalm/jfaf183

Hawkins DM (2008). *Outliers* in Wiley Encyclopedia of Clinical Trials, eds R. D'Agostino, L. Sullivan, and J. Massaro. Wiley, New York.

Examples

```
# library
library(ppwdeming)

# parameter specifications
n <- 100

sigma <- 1
kappa <- 0.08
alpha <- 1
beta  <- 1.1
true  <- 8*10^((0:(n-1))/(n-1))
truey <- alpha+beta*true
# simulate single sample - set seed for reproducibility
set.seed(1069)
# specifications for predicate method
X     <- sigma*rnorm(100)+true *(1+kappa*rnorm(n))
# specifications for test method
Y     <- sigma*rnorm(100)+truey*(1+kappa*rnorm(n))
# add some outliers
Y[c(1,2,100)] <- Y[c(1,2,100)] + c(-10,9,-50)

# check for outliers and store output
outliers_assess <- PWD_outlier(X, Y, K=5)

# summary of process
summary(outliers_assess)
```

`PWD_resi` *Fit Rocke-Lorenzato profile model to residuals*

Description

This routine fits the Rocke-Lorenzato precision profile model to the **residuals** from the fit.

Usage

```
PWD_resi(true, resi, epsilon=1e-8,
         printem=lifecycle::deprecated())
```

Arguments

`true`	the vector of values used to predict the precision – commonly X.
`resi`	the vector of residuals whose variance is thought to be a function of "true".
`epsilon`	*optional* (default of 1e-8) - convergence tolerance limit.
`printem`	**[Deprecated]** `printem = TRUE` is no longer supported; separate summary functions are provided for printing output.

Details

The Rocke-Lorenzato precision profile model is

$$SD^2 = \sigma_r^2 + (\kappa_r \cdot true)^2$$

for the *residuals* from a precision-profile model fit.

Under this model, the approach for reviewing residuals is to fit a variance profile model to the residuals r_i themselves. The output of this function includes a maximum-likelihood estimate of the remaining parameter in the special cases of:

- constant variance (κ_r = 0); and
- constant coefficient of variation (σ_r = 0).

Value

A list containing the following components:

`sigmar`	the estimate of σ_r
`kappar`	the estimate of κ_r
`L`	the -2 log likelihood
`scalr`	the scaled residuals
`poolsig`	the maximum likelihood estimate of σ_r if κ_r = 0
`poolkap`	the maximum likelihood estimate of κ_r if σ_r = 0

Author(s)

Douglas M. Hawkins, Jessica J. Kraker krakerjj@uwec.edu

References

Hawkins DM and Kraker JJ (2026). Precision Profile Weighted Deming Regression for Methods Comparison. *The Journal of Applied Laboratory Medicine*, **11**(2), 379-392. doi:10.1093/jalm/jfaf183

Hawkins DM (2014). A Model for Assay Precision. *Statistics in Biopharmaceutical Research*, **6**, 263-269. http://dx.doi.org/10.1080/19466315.2014.899511

Examples

```
# library
library(ppwdeming)

# parameter specifications
n <- 100

sigma <- 1
kappa <- 0.08
alpha <- 1
beta  <- 1.1
```

```
true  <- 8*10^((0:(n-1))/(n-1))
truey <- alpha+beta*true
# simulate single sample - set seed for reproducibility
set.seed(1039)
# specifications for predicate method
X     <- sigma*rnorm(100)+true *(1+kappa*rnorm(n))
# specifications for test method
Y     <- sigma*rnorm(100)+truey*(1+kappa*rnorm(n))

# fit the model and store output
RL_gh_fit  <- PWD_get_gh(X,Y)
# run the residual analysis from the model output
post  <- PWD_resi(X, RL_gh_fit$resi)
summary(post)
```

PWD_RL *Weighted Deming – Rocke-Lorenzato - known sigma, kappa*

Description

This code fits the weighted Deming regression on predicate readings (X) and test readings (Y), with user-supplied Rocke-Lorenzato ("RL") parameters sigma (σ) and kappa (κ).

Usage

```
PWD_RL(X, Y, sigma, kappa, lambda=1, alpha=NA, beta=NA, mu=NA, epsilon=1e-8)
```

Arguments

X	the vector of predicate readings.
Y	the vector of test readings.
sigma	the RL σ parameter.
kappa	the RL κ parameter.
lambda	*optional* (default of 1) - the ratio of the X to the Y precision profile.
alpha	*optional* (default of NA) - numeric, single value, initial estimate of α.
beta	*optional* (default of NA) - numeric, single value, initial estimate of β.
mu	*optional* (default of NA) - numeric, vector of length of X, initial estimate of μ.
epsilon	*optional* (default of 1e-8) - convergence tolerance limit.

Details

The Rocke-Lorenzato precision profile model assumes the following forms for the variances, with proportionality constant λ:

- predicate precision profile model: $g_i = var(X_i) = \lambda \left(\sigma^2 + [\kappa \cdot \mu_i]^2\right)$ and

- test precision profile model: $h_i = var(Y_i) = \sigma^2 + [\kappa \cdot (\alpha + \beta\mu_i)]^2$.

The algorithm uses maximum likelihood estimation. Proportionality constant λ is assumed to be known or estimated externally.

Value

A list containing the following components:

`alpha`	the fitted intercept
`beta`	the fitted slope
`fity`	the vector of predicted Y
`mu`	the vector of estimated latent true values
`resi`	the vector of residuals
`L`	the -2 log likelihood L
`innr`	the number of inner refinement loops executed
`error`	an error code if the iteration fails

Author(s)

Douglas M. Hawkins, Jessica J. Kraker krakerjj@uwec.edu

References

Hawkins DM and Kraker JJ (2026). Precision Profile Weighted Deming Regression for Methods Comparison. *The Journal of Applied Laboratory Medicine*, **11**(2), 379-392. doi:10.1093/jalm/jfaf183

Hawkins DM (2014). A Model for Assay Precision. *Statistics in Biopharmaceutical Research*, **6**, 263-269. doi:10.1080/19466315.2014.899511

Examples

```
# library
library(ppwdeming)

# parameter specifications
n <- 100

sigma <- 1
kappa <- 0.08
alpha <- 1
beta  <- 1.1
true  <- 8*10^((0:(n-1))/(n-1))
truey <- alpha+beta*true
# simulate single sample - set seed for reproducibility
set.seed(1039)
# specifications for predicate method
X     <- sigma*rnorm(100)+true *(1+kappa*rnorm(n))
# specifications for test method
```

```
Y     <- sigma*rnorm(100)+truey*(1+kappa*rnorm(n))

# fit RL with given sigma and kappa
RL_results <- PWD_RL(X,Y,sigma,kappa)
cat("\nWith given sigma and kappa, the estimated intercept is",
    signif(RL_results$alpha,4), "and the estimated slope is",
    signif(RL_results$beta,4), "\n")
```

---

summary.pwdgetgh *Summary information of output from* PWD_get_gh *or* multi_PWD

Description

The function, invisible to the user, is called by summary() to display the formatted results of a model fitted by PWD_inference or by multi_PWD.

Usage

```
## S3 method for class 'pwdgetgh'
summary(object, digits = 6, ...)
```

Arguments

object	an object produced by a call to function PWD_inference or multi_PWD
digits	*optional* (default of 6) - number of decimal places to use for rounding displayed results.
...	Further arguments passed to or from other methods (required for compatibility with the generic summary function).

Value

object is returned invisibly per generic summary method conventions.

Examples

```
# library
library(ppwdeming)

# parameter specifications
quad  <- FALSE   # for a linear fit
quad  <- TRUE    # for a quadratic fit
sigma <- 1
kappa <- 0.08
alpha <- 1
beta  <- 1.1
gamma <- 0
if (quad) gamma <- -0.005
true  <- 8*10^((0:99)/99)
```

```
truey <- alpha+beta*true+gamma*true^2
# simulate single sample - set seed for reproducibility
set.seed(1039)
# specifications for predicate method
X     <- sigma*rnorm(100)+true *(1+kappa*rnorm(100))
# specifications for test method
Y     <- sigma*rnorm(100)+truey*(1+kappa*rnorm(100))

# fit with RL precision profile to estimate parameters
RL_gh_fit  <- PWD_get_gh(X,Y, quad=quad)

# results
summary(RL_gh_fit)
```

`summary.pwdinf` *Summary information of output from* `PWD_inference` *or* `multiPWD_inf`

Description

The function, invisible to the user, is called by `summary()` to display the formatted results of a model fitted by `PWD_inference` or by `multiPWD_inf`.

Usage

```
## S3 method for class 'pwdinf'
summary(object, conflevel = 0.95, digits = 4, ...)
```

Arguments

`object`	an object produced by a call to function `PWD_inference` or `multiPWD_inf`
`conflevel`	*optional* (default of 0.95) - confidence level, between 0 to 1; value of `NA` will remove the confidence bounds.
`digits`	*optional* (default of 4) - number of decimal places to use for rounding displayed results.
`...`	Further arguments passed to or from other methods (required for compatibility with the generic `summary` function).

Value

`object` is returned invisibly per generic `summary` method conventions.

Examples

```
# library
library(ppwdeming)

# parameter specifications
sigma <- 1
kappa <- 0.08
alpha <- 1
beta  <- 1.1
true  <- 8*10^((0:99)/99)
truey <- alpha+beta*true
# simulate single sample - set seed for reproducibility
set.seed(1039)
# specifications for predicate method
X     <- sigma*rnorm(100)+true *(1+kappa*rnorm(100))
# specifications for test method
Y     <- sigma*rnorm(100)+truey*(1+kappa*rnorm(100))

# fit with RL precision profile to estimate parameters and variability
RL_inf <- PWD_inference(X,Y,MDL=12)
summary(RL_inf)
```

---

summary.pwdknown *Summary information of output from* PWD_known

---

Description

The function, invisible to the user, is called by summary() to display the formatted results of a model fitted by PWD_known.

Usage

```
## S3 method for class 'pwdknown'
summary(object, digits = 4, ...)
```

Arguments

object	an object produced by a call to function PWD_known
digits	*optional* (default of 4) - number of decimal places to use for rounding displayed results.
...	Further arguments passed to or from other methods (required for compatibility with the generic summary function).

Value

object is returned invisibly per generic summary method conventions.

Examples

```
# library
library(ppwdeming)

# parameter specifications
alpha <- 1
beta  <- 1.1
true  <- 8*10^((0:99)/99)
truey <- alpha+beta*true
# forms of precision profiles
gfun    <- function(true, gparms) {
  gvals = gparms[1]+gparms[2]*true^gparms[3]
  gvals
}
hfun    <- function(true, hparms) {
  hvals = hparms[1]+hparms[2]*true^hparms[3]
  hvals
}

# Loosely motivated by Vitamin D data set
g     <- 4e-16+0.07*true^1.27
h     <- 6e-2+7e-5*truey^2.2
# simulate single sample - set seed for reproducibility
set.seed(1039)
# specifications for predicate method
X     <- true +sqrt(g)*rnorm(100)
# specifications for test method
Y     <- truey+sqrt(h)*rnorm(100)

# fit with to estimate linear parameters
pwd_known_fit <- PWD_known(X, Y, gfun, hfun,
                           gparms=c(4e-16, 0.07, 1.27),
                           hparms=c(6e-2, 7e-5, 2.2), MDL=12)
summary(pwd_known_fit)
```

---

summary.pwdout *Summary information of output from* PWD_outlier *or* multi_PWD_out

---

Description

The function, invisible to the user, is called by summary() to display the formatted results of a model fitted by PWD_outlier or by multi_PWD_out.

Usage

```
## S3 method for class 'pwdout'
summary(object, ...)
```

Arguments

object	an object produced by a call to function PWD_outlier or multi_PWD_out
...	Further arguments passed to or from other methods (required for compatibility with the generic summary function).

Value

object is returned invisibly per generic summary method conventions.

Examples

```
# library
library(ppwdeming)

# parameter specifications
sigma <- 1
kappa <- 0.08
alpha <- 1
beta  <- 1.1
true  <- 8*10^((0:99)/99)
truey <- alpha+beta*true
# simulate single sample - set seed for reproducibility
set.seed(1069)
# specifications for predicate method
X     <- sigma*rnorm(100)+true *(1+kappa*rnorm(100))
# specifications for test method
Y     <- sigma*rnorm(100)+truey*(1+kappa*rnorm(100))
# add some outliers
Y[c(1,2,100)] <- Y[c(1,2,100)] + c(-10,9,-50)

# check for outliers, re-fit, and store output
outliers_assess <- PWD_outlier(X, Y, K=5)
summary(outliers_assess)
```

---

summary.pwdresi *Summary information of output from* PWD_resi

---

Description

The function, invisible to the user, is called by summary() to display the formatted results of output from PWD_resi.

Usage

```
## S3 method for class 'pwdresi'
summary(object, digits = 4, ...)
```

Arguments

object	an object produced by a call to function PWD_resi
digits	*optional* (default of 4) - number of decimal places to use for rounding displayed results.
...	Further arguments passed to or from other methods (required for compatibility with the generic summary function).

Value

object is returned invisibly per generic summary method conventions.

Examples

```
# library
library(ppwdeming)

# parameter specifications
sigma <- 1
kappa <- 0.08
alpha <- 1
beta  <- 1.1
true  <- 8*10^((0:99)/99)
truey <- alpha+beta*true
# simulate single sample - set seed for reproducibility
set.seed(1039)
# specifications for predicate method
X     <- sigma*rnorm(100)+true *(1+kappa*rnorm(100))
# specifications for test method
Y     <- sigma*rnorm(100)+truey*(1+kappa*rnorm(100))

# fit the model and store output
RL_gh_fit  <- PWD_get_gh(X,Y)
# run the residual analysis from the model output
post  <- PWD_resi(X, RL_gh_fit$resi)
summary(post)
```

WD_General *Weighted Deming Regression*

Description

This code fits the weighted Deming regression on predicate readings (X) and test readings (Y).

Usage

```
WD_General(X, Y, g, h, epsilon=1e-8)
```

Arguments

X	the vector of predicate readings.
Y	the vector of test readings.
g	the vector of variances of the X.
h	the vector of variances of the Y.
epsilon	*optional* (default of 1e-8) - convergence tolerance limit.

Details

This function is used when the variances of X and Y are already known, as can be the case from an adequate number of replicate X and Y readings of each sample. For input vectors g and h containing the variances of predicate readings X and test readings Y, respectively, iteratively fits weighted Deming regression.

Value

A list containing the following components:

alpha	the fitted intercept
beta	the fitted slope
cor	the Pearson correlation between X and Y
fity	the vector of predicted Y
mu	the vector of estimated latent true values
resi	the vector of residuals
L	the -2 log likelihood L
innr	the number of inner refinement loops executed

Author(s)

Douglas M. Hawkins, Jessica J. Kraker krakerjj@uwec.edu

References

Ripley BD and Thompson M (1987). Regression techniques for the detection of analytical bias. *Analyst*, **112**, 377-383.

Examples

```
# library
library(ppwdeming)

# parameter specifications
n <- 100

alpha <- 1
beta  <- 1.1
true  <- 8*10^((0:(n-1))/(n-1))
```

```
truey <- alpha+beta*true
# Loosely motivated by Vitamin D data set
g     <- 4e-16+0.07*true^1.27
h     <- 6e-2+7e-5*truey^2.2

# simulate single sample - set seed for reproducibility
set.seed(1039)
# specifications for predicate method
X     <- true +sqrt(g)*rnorm(n)
# specifications for test method
Y     <- truey+sqrt(h)*rnorm(n)

# fit with to estimate linear parameters
wd_fit <- WD_General(X,Y,g,h)
cat("\nWith given g and h, the estimated intercept is",
    signif(wd_fit$alpha,4), "and the estimated slope is",
    signif(wd_fit$beta,4), "\n")
```

`WD_Linnet` *Linnet proportional CV weighted Deming*

Description

This routine, provided for convenience, makes Linnet's constant CV fit.

Usage

```
WD_Linnet(X, Y, lambda=1, MDL=NA, getCI=TRUE, epsilon=1e-10, printem=FALSE)
```

Arguments

`X`	the vector of predicate readings.
`Y`	the vector of test readings.
`lambda`	ratio of g function to h function.
`MDL`	*optional* (default of NA) - medical decision limit(s).
`getCI`	*optional* (default of TRUE) - if TRUE, generates jackknife standard errors.
`epsilon`	*optional* (default of 1e-10) - tolerance limit.
`printem`	*optional* (default of FALSE) - if TRUE, prints out results as a `message`.

Details

Note that in cases where sigma happens to come out zero, Linnet's constant CV fit differs from the precision-profile fit since the underlying precision profile models are not the same.

Value

A list containing the following components:

alpha	the fitted intercept
beta	the fitted slope
cor	the Pearson correlation between X and Y
sealpha	the jackknife standard error of alpha
sebeta	the jackknife standard error of beta
covar	the jackknife covariance between alpha and beta
preMDL	the predictions at the MDL(s)
preMDLl	the lower confidence limit(s) of preMDL
preMDLu	the upper confidence limit(s) of preMDL

Author(s)

Douglas M. Hawkins, Jessica J. Kraker krakerjj@uwec.edu

References

Linnet K (1993). Evaluation of regression procedures for methods comparison studies. *Clinical Chemistry*, **39**: 424-432.

Examples

```
# library
library(ppwdeming)

# parameter specifications
n <- 100
alpha <- 1
beta  <- 1.1
true  <- 8*10^((0:(n-1))/(n-1))
truey <- alpha+beta*true
kappa <- 0.1

# simulate single sample - set seed for reproducibility
set.seed(1039)
# specifications for predicate method
X     <- true *(1+kappa*rnorm(n))
# specifications for test method
Y     <- truey *(1+kappa*rnorm(n))

# fit with to estimate linear parameters
wd_fit <- WD_Linnet(X, Y, MDL=12, printem=TRUE)
cat("\nThe Linnet constant-CV estimated intercept is",
    signif(wd_fit$alpha,4), "and the estimated slope is",
    signif(wd_fit$beta,4), "\n")
```

Index