Package 'meerva'

October 13, 2022

Title Analysis of Data with Measurement Error Using a Validation Subsample

Version 0.2-2

Date 2021-10-26

Depends R (>= 3.4.0)

Imports survival , dplyr , tidyr , ggplot2 , mvtnorm , matrixcalc

ByteCompile Yes

Description

Sometimes data for analysis are obtained using more convenient or less expensive means yielding ``surrogate" variables for what could be obtained more accurately, albeit with less convenience; or less conveniently or at more expense yielding ``reference" variables, thought of as being measured without error. Analysis of the surrogate variables measured with error generally yields biased estimates when the objective is to make inference about the reference variables. Often it is thought that ignoring the measurement error in surrogate variables only biases effects toward the null hypothesis, but this need not be the case. Measurement errors may bias parameter estimates either toward or away from the null hypothesis. If one has a data set with surrogate variable data from the full sample, and also reference variable data from a randomly selected subsample, then one can assess the bias introduced by measurement error in parameter estimation, and use this information to derive improved estimates based upon all available data. Formulaically these estimates based upon the reference variables from the validation subsample combined with the surrogate variables from the whole sample can be interpreted as starting with the estimate from reference variables in the validation subsample, and ``augmenting" this with additional information from the surrogate variables. This suggests the term ``augmented" estimate. The meerva package calculates these augmented estimates in the regression setting when there is a randomly selected subsample with both surrogate and reference variables. Measurement errors may be differential or non-differential, in any or all predictors (simultaneously) as well as outcome. The augmented estimates derive, in part, from the multivariate correlation between regression model parameter estimates from the reference variables and the surrogate variables, both from the validation subset. Because the validation subsample is chosen at random any biases imposed by measurement error, whether non-differential or differential, are reflected in this correlation and these correlations can be used to derive estimates for the reference variables using data from the whole sample. The main functions in the package are meerva.fit which calculates estimates for a dataset, and meerva.sim.block which simulates multiple datasets as described by the user, and analyzes these datasets, storing the regres-

sion coefficient estimates for inspection. The augmented estimates, as well as how measurement error may arise in practice, is described in more detail by Kremers WK (2021) <arXiv:2106.14063> and is an extension of the works by Chen Y-H, Chen H. (2000) <doi:10.1111/1467-9868.00243>, Chen Y-H. (2002) <doi:10.1111/1467-9868.00324>, Wang X, Wang Q (2015) <doi:10.1016/j.jmva.2015.05.017> and Tong J, Huang J, Chubak J, et al. (2020) <do

License GPL-3

NeedsCompilation no

Copyright Mayo Foundation for Medical Education and Research

RoxygenNote 7.1.2

Suggests R.rsp

VignetteBuilder R.rsp

Author Walter K Kremers [aut, cre] (<https://orcid.org/0000-0001-5714-3473>)

Maintainer Walter K Kremers <kremers.walter@mayo.edu>

Repository CRAN

Date/Publication 2021-10-27 10:30:06 UTC

R topics documented:

compmse . . . 3
coverage . . . 3
coverage2 . . . 4
dfbetac . . . 4
meerva . . . 5
meerva.fit . . . 5
meerva.sim.block . . . 11
meerva.sim.brn . . . 15
meerva.sim.cox . . . 18
meerva.sim.nrm . . . 20
myttest . . . 22
plot.meerva.sim . . . 23
print.meerva . . . 23
print.meerva.sim . . . 24
reldif . . . 24
subvec . . . 25
summary.meerva . . . 25
summary.meerva.sim . . . 26
ztest . . . 26

Index **28**

compmse *Comapre bias, var and MSE*

Description

Comapre bias, var and MSE

Usage

```
compmse(ibias, ivals)
```

Arguments

ibias	Matrix of bias's
ivals	Matrix of var's

Value

A matrix

coverage *Calculate coverage probabilties*

Description

Calculate coverage probabilties

Usage

```
coverage(estimates, vars, beta, round = 3)
```

Arguments

estimates	A matrix of estimates
vars	A matrix of varainces
beta	The beta under H0
round	The decimal places for rounding

Value

95

`coverage2` *Compare coverage probabilities between two estimators on matched simulations*

Description

Compare coverage probabilities between two estimators on matched simulations

Usage

```
coverage2(estimates1, vars1, estimates2, vars2, beta, round = 3)
```

Arguments

`estimates1`	Matrix of estimates for 1
`vars1`	Matrix of vars for 1
`estimates2`	Matrix of estimates for 2
`vars2`	Matrix of vars for 2
`beta`	The beta under H0
`round`	The decimal places for rounding

Value

Comparison of coverage probabilities between two estimators.

`dfbetac` *Sum dfbeta s According to id_vector Clusters*

Description

Sum dfbeta s According to id_vector Clusters

Usage

```
dfbetac(id_vector, dfbeta)
```

Arguments

`id_vector`	Input id vector
`dfbeta`	dfbeta s for sandwich

Value

dfbeta by id_vector clusters

`meerva` *Analysis of Data with Measurement Error Using a Validation Subsample*

Description

The meerva package performs regression analyses on data with measurement error when there is a validation subsample. The functional .fit program is meerva.fit. The meerva function is intended for future development and use as a wrapper for meerva.fit. Try help(meerva.fit).

Usage

```
meerva()
```

Value

Describes future development of the meerva package.

Author(s)

Walter Kremers (kremers.walter@mayo.edu)

See Also

`meerva.fit` , `meerva.sim.block` , `meerva.sim.brn` , `meerva.sim.cox` , `meerva.sim.nrm`

`meerva.fit` *Analysis of Data with Measurement Error Using a Validation Subsample*

Description

The meerva package is designed to analyze data with measurement error when there is a validation subsample randomly selected from the full sample. The method assumes surrogate variables measured with error are available for the full sample, and reference variables measured with little or no error are available for this randomly chosen subsample of the full sample. Measurement errors may be differential or non differential, in any or all predictors (simultaneously) as well as outcome. The "augmented" estimates derived by meerva are based upon the multivariate correlation between regression models based upon the reference variables and the surrogate variables in the validation subset. Because the validation subsample is chosen at random whatever biases are imposed by measurement error, non-differential or differential, are reflected in this correlation and can be used to derives estimates for the reference variables using data from the whole sample.

Intuitively one expects there to be at least one surrogate for each reference variable but the method is based upon multivariate correlations and therefore also works if there are more or fewer surrogate than reference variables. The package fits linear, logistic or Cox regression models individually to the reference variales and to the surrogate varibles, then combines the results to descibe a model in terms of the reference variables based upon the entire dataset.

Usage

```
meerva.fit(
  x_val,
  y_val,
  xs_val,
  ys_val,
  xs_non,
  ys_non,
  e_val = NULL,
  es_val = NULL,
  es_non = NULL,
  id_val = NULL,
  id_non = NULL,
  weights_val = NULL,
  weights_non = NULL,
  familyr = NULL,
  familys = NULL,
  vmethod = NULL,
  jksize = 0,
  compare = 1
)
```

Arguments

x_val	A matrix object including reference predictor variables (and predictors "without" error) in validation subsample. This and other x_ matrices must not include any missing values (NA). All data vectors and matrices must be numerical. For categorical variables one should first construct corresponding numerical variables to represent these categories.
y_val	A vector object for the reference outcome variable in validation subsample. This and other y_ vectors must not include any missing values (NA).
xs_val	A matrix object including surrogate predictors (and predictors "without" error) in validation subsample
ys_val	A vector object for the surrogate outcome variable in validation sample.
xs_non	A matrix object including surrogate predictors (and predictors "without" error) in NON validation data
ys_non	A vector object for the surrogate outcome variable in the NON validation sample.
e_val	A vector object for the survival data reference event outcome variable in validation subsample. This and the other e_ vectors are optional. The e_ vectors are required when analyzing survival data based upon an underlying Cox regression model (survival package). This and other e_ vectors must not include any missing values (NA).
es_val	A vector object for the survival data surrogate event outcome variable in validation subsample.

es_non — A vector object for the survival data surrogate event outcome variable in NON validation data.

id_val — A vector object identifying clusters in case of multiple records per subject in the validation subsample. This and id_non are optional. They must not include any missing values (NA). No subjects should be included in both the validation subsample and the NON validation data.

id_non — A vector object identifying clusters in case of multiple records per subject in the NON validation data.

weights_val — A vector object with weights used in model fit of the validation subsample. This can be used, for example, to describe inverse sampling probability weights. Note, when fitting the "binomial" or logistic model, weights for weights_val and weights_non must be integer. This is a restriction of the glm.fit routine called from meerva. The user may rescale or round the weights to achieve integers. By using robust variance estimates meerva provides correct variance estimates.

weights_non — A vector object with weights used in model fit of the NON validation subsample. This and weights_val, can be used, for example, to down weight records from patients with multiple records.

familyr — The family for the underlying regression model amongst "binomial", "gaussian" and "Cox". Default is NULL and the program chooses amongst these three based upon a simple data inspection. The regression model for the reference variables may be of a different type than for the surrogate variables. For example the reference outcome could be yes/no in nature while the surrogate outcome could be a numeric, and the method continues to work.

familys — The family for the underlying surrogate regression model if different from the reference model. See familyr. Default is NULL and familys takes the same form as for familyr, if specified. If both familyr and familys are NULL then the program chooses from "binomial", "gaussian" and "Cox" based upon a simple data inspection.

vmethod — Method for robust estimation of variance covariance matrices needed for calculation of the augmented estimates (beta aug). 0, 1 or 2 determines JK (slow), IJK using dfbeta of glm or coxph, or IJK using an alternate formula for dfbeta. Recommendations: For "gaussian" use 1, for "Cox" use 1 for speed and 0 for accuracy, and for "binomial" use 2 for speed, and 0 for accuracy.

jksize — Number of elements to leave out number in each cycle of the grouped jackknife for non validation data. The default is 0 where the program chooses jksize so that the number of leave out groups is about validation subsample size. For the grouped jackknife the program randomly sorts the non validation subsample. To get the exact same results twice one can set the seed for the random generator with the statement set.seed(seed) for some value of seed, and to get a "random" seed one can first run the statement seed = round(runif(1)*1000000000) .

compare — 1 to compare gamma_val with gamma_ful (default) or 0 with gamma_non. See below under "coef_gamma" for clarificaiton of gamma_ful and gamma_non. Comparisons of gamma_val with gamma_ful is consistent with the principle of the validation set being a subsample of the entire dataset. This assures the correlations between gamma_val and beta_val are representative of what should

be the case based upon the whole dataset. If there were an external validation sample where one could be reasonably certain that the correlation between gamma_val and beta_val would be representative then one could also use this method.

Details

As currently implemented the package requires the data to be input as vectors and matrices with no missing values (NA). All data vectors and matrices must be numerical. For categorical variables one should first construct corresponding numerical variables to represent these categories. Note, variables thought of as measured without error should be included in both the reference variable set and the surrogate variable set. Such variables may be thought of as perfect surrogates. This applies for both outcome variables and predictor variables. For the Cox model both the time to event and the event indicator may be measured with error.

The length of the vectors for the validation subsample must all be the same, and be the same as the number of rows in the predictor matrices for the validation subsample. Data for sample elements not included in the validation subsample are referred to as NON validation data and are to be included in separate vectors and matrix. Here, too, the length of all vectors must be the same as number of rows in the predictor matrix. The columns in the data matrix for the validation subsample surrogates must be logically the same as the columns in the data matrix for the NON validation surrogates.

The data for analysis may include weights, for example to account for non identical sampling probabilities when selecting the subsample, by taking weights as the inverse of these probabilities. The data may include cluster identifiers in case of multiple observations on study participants. Weights may also be used to lessen the influence of individuals with multiple observations.

Internally the analysis uses robust variance estimation which accounts for deviations from the usual regression model assumptions for the surrogate variables, and accounts for multiple observations per patient.

This package came out of our work analyzing electronic health records data, where different sources, e.g diagnosis codes and natural language processing, may provide different surrogate variables. Reference variables were obtained by manual chart review. For our datasets to date with tens of thousands of patients the analyses take a few seconds when run on a PC.

In the examples we generate simulated data of the form expected for input, call the main program, and summarize the output.

Value

meerva.fit returns an object of class meerva which contains the augmented estimates based upon the full data set accounting for measurement error, estimates based upon reference variables from the validation subsample, estimates based upon the surrogate variables from the whole sample, along with robust variance-covariances matrix estimates for these estimates. This meerva class list contains the following objects.

Call — The call used to invoke meerva.fit.

FitInput — A list with

— familyr — The type of regression model fit to the data.

— compare — The input parameter compare, 1 to compare the validation data with the whole dataset, or 0 to compare with the NON validation data.

— comparec — A short text interpretation of compare.

— vmethod — The method used to estimate the variance-covariance matrices needed for calculation of the estimates.

— vmethodc — A short text description of vmethod.

— n_val — The number of observations in the validation subsample.

— n_ful — The number of observations in the whole dataset.

— n_val_id — The number of clusters identified by id_val in the validation subsample.

— n_ful_id — The number of clusters identified by id_val and id_non in the whole dataset.

— dim_beta — The number of parameters in the regression model for reference variables including a possible intercept.

— dim_gamma — The number of parameters in the regression model for surrogate variables including a possible intercept.

names_x — The reference variable predictors used in analysis.

names_xs — The surrogate variable predictors used in analysis.

names_y — The reference outcome variable used in analysis.

names_ys — The surrogate outcome variable used in analysis.

coef_beta — The regression parameter estimates for the reference variables including both beta_val based upon the reference variables alone (available only in the validation subsample) and beta_aug, the augmented estimates based upon the reference variables in the validation subsample augmented by the surrogate variables in the whole dataset.

coef_gamma — The regression parameter estimates for the surrogate variables for both gamma_val derived using dataset elements included in the validation subsample, and either gamma_ful or gamma_non, derived using either the whole sample or the NON validation data.

var_beta — Robust variance estimates for coef_beta, which are also included in vcov_beta and vcov_beta_val.

var_gamma — Robust variance estimates for coef_gamma, which are also included in vcov_gamma.

vcov_beta_aug — Robust variance-covariance estimates for beta_aug of coef_beta.

vcov_beta_val — Robust variance-covariance estimates for beta_val of coef_beta.

vcov_beta_val_naive — Naive variance-covariance estimates for beta_val of coef_beta obtained without any consideration of clustering optionally described by input parameters id_val and id_non.

vcov_gamma_ful — Robust variance-covariance estimates for gamma_ful of coef_gamma.

or vcov_gamma_non — Robust variance-covariance estimates for gamma_non of coef_gamma.

vcov_gamma_ful_naive — Naive variance-covariance estimates for gamma_ful of coef_gamma obtained without any consideration of clustering optionally described by input parameters id_val and id_non.

or vcov_gamma_non_naive — Like vcov_gamma_ful_naive but for gamma_non.

omega — The robust covariance estimate between beta_val and either gamma_ful or gamma_non, which is integral for derivation of beta_aug.

omega_cor — The robust correlation estimate between beta_val and either gamma_ful or gamma_non, which reflects the relative amount of information on reference variable estimates contained in the surrogate variables.

kappa — The robust variance covariance estimate of either (gamma_val - gamma_ful) or (gamma_val - gamma_non), which is integral for derivation of beta_aug.

Author(s)

Walter Kremers (kremers.walter@mayo.edu)

References

Chen Y-H, Chen H. A Unified Approach to Regression Analysis under Double-Sampling Designs. Journal of the Royal Statistical Society. Series B (Statistical Methodology) , 2000 (62) 449-460.

Chen Y-H. Cox regression in cohort studies with validation sampling. Journal of the Royal Statistical Society. Series B (Statistical Methodology), 2002 64, 51-62.

Wang X, Wang QH. Semiparametric linear transformation model with differential measurement error and validation sampling. J Multivariate Anal. 2015;141:67-80.

Tong JY, Huang J, Chubak J, et al. An augmented estimation procedure for EHR-based association studies accounting for differential misclassification. J Am Med Inform Assn. 2020;27(2):244-253.

See Also

meerva.sim.block , meerva.sim.brn , meerva.sim.cox , meerva.sim.nrm

Examples

```
#=====================================================

# Simulate logistic regression data with measurement error
simd = meerva.sim.brn(n=4000, m=400,
     beta = c(-0.5, 0.5, 0.2, 1, 0.5) ,
     alpha1 = c(0.95, 0.90, 0.90, 0.95) ,
     alpha2 = c(0.98,0.94,0.95,0.95) ,
     bx3s1 = c(0.05, 0, 0, NA, NA) ,
     bx3s2 = c(NA,NA,NA) )

# Read the simulated data to input data format
x_val  = simd$x_val
y_val  = simd$y_val
xs_val = simd$xs_val
ys_val = simd$ys_val
xs_non = simd$xs_non
ys_non = simd$ys_non

# Analyze the data
brn.me = meerva.fit(x_val, y_val, xs_val, ys_val, xs_non, ys_non)
summary(brn.me)

#=====================================================

# Simulate linear regression data with measurement error
simd = meerva.sim.nrm(n=4000, m=400,
     beta=c(-0.5,0.5,0.2,1,0.5),
```

```
     alpha1=c(-0.05,0.1,0.05,0.1),
     alpha2=c(0.95,0.91,0.9,0.9),
     bx3s1= c(0.05, 0, 0, NA, NA),
     bx3s2=c(1.1,0.9,0.05),
     sd=5)

# Read the simulated data to input data format
x_val  = simd$x_val
y_val  = simd$y_val
xs_val = simd$xs_val
ys_val = simd$ys_val
xs_non = simd$xs_non
ys_non = simd$ys_non

# Analyze the data
nrm.me = meerva.fit(x_val, y_val, xs_val, ys_val, xs_non, ys_non)
summary(nrm.me)

#=====================================================
# Simulate Cox regression data with measurement error
simd = meerva.sim.cox(n=4000, m=400,
     beta   = c(-0.5, 0.5, 0.2, 1, 0.5) ,
     alpha1 = c(0.95,0.90,0.90,0.95)  ,
     alpha2 = c(0.98,0.94,0.94,0.98) ,
     bx3s1  = c(0.05,0,0,NA,NA) ,
     bx3s2  = c(1.1, NA, NA) ,
     sd=0.1)

# Read the simulated data to input data format
x_val  = simd$x_val
y_val  = simd$y_val
xs_val = simd$xs_val
ys_val = simd$ys_val
xs_non = simd$xs_non
ys_non = simd$ys_non
e_val  = simd$e_val
es_val = simd$es_val
es_non = simd$es_non

# Analyze the data
cox.me = meerva.fit(x_val, y_val, xs_val, ys_val, xs_non, ys_non,
                    e_val, es_val, es_non)
summary(cox.me)

#=====================================================
```

---

meerva.sim.block *Simulation of meerva used to Analyze Data with Measurement Error*

Description

The meerva package is designed to analyze data with measurement error when there is a validation subsample randomly selected from the full sample. The method assumes surrogate variables measured with error are available for the full sample, and reference variables measured with little or no error are available for this randomly chosen subsample of the full sample. Measurement errors may be differential or non differential, in any or all predictors (simultaneously) as well as outcome.

The meerva.sim.block lets the user specify a model with measurement error, and then simulate and analyze many datasets to examine the model fits and judge how the method works. Data sets are generated according to 3 functions for simulating Cox PH, linear and logistic regression models. These functions generate data sets with 4 reference predictor variables and from 3 to 5 surrogate predictor variables. The user can consider, program and simulate data sets of greater complexity but these examples provided with the package should serve as a reasonable introduction to the robustness of the method.

Usage

```
meerva.sim.block(
  simfam = "gaussian",
  nsims = 100,
  seed = 0,
  n = 4000,
  m = 400,
  beta = c(-0.5, 0.5, 0.2, 1, 0.5),
  alpha1 = c(-0.05, 0.1, 0.05, 0.1),
  alpha2 = c(0.98, 0.98, 0.95, 0.95),
  bx3s1 = c(0.05, 0, 0, NA, NA),
  bx3s2 = c(0.95, NA, NA),
  bx12 = c(0.25, 0.15),
  sd = 1,
  fewer = 0,
  mncor = 0,
  sigma = NULL,
  vmethod = NA,
  jksize = 0,
  compare = 1,
  diffam = NA,
  simtime = 1
)
```

Arguments

simfam	The family for the underlying regression model to be simulated, amongst "binomial", "gaussian" and "Cox".
nsims	Number of datasets to be simulated
seed	A seed for the R random number generator. The default is 0 in which case the program random selects and records the seed so one ca replicate simulation studies.

n	The full dataset size.
m	The validation subsample size (m < n).
beta	A vector of length 5 for the true regression parameter for the linear regression model with 5 predictors including the intercept. For the Cox model beta[0] is not estimated but determines a basal event rate.
alpha1	A vector of length four determining the measurement error or misclassification probabilities for the outcome surrogate ys. Usage is slightly different for the different simfam values "gaussian", "binomial" and "Cox". See the help pages for meerva.sim.brn, meerva.sim.cox and meerva.sim.nrm for clarification.
alpha2	A vector describing the correct classification probabilities for x1s, the surrogate for x1. Usage is slighlty different for the different simfam values "gaussian", "binomial" and "Cox". See the help pages for meerva.sim.brn, meerva.sim.cox and meerva.sim.nrm for clarification.
bx3s1	A vector of length 5 determining the relation between the reference variable x3 and the mean and SD of the surrogate x3s1. Roughly, bx3s1[1] determines a minimal measurement error SD, conditional on x3 bx3s1[2] determines a rate of increase in SD for values of x3 greater than bx3s1[3], bx3s1[4] is a value above which the relation between x3 and the mean of x3s is determined by the power bx3s1[5]. The mean values for x3s1 are rescaled to have mean 0 and variance 1.
bx3s2	A vector of length 3 determining scale in x3s and potentially x3s2, a second surrogate for xs. Roughly, bx3s2[1] takes the previously determined mean for x3s1 using bx3s1 and multiples by bx3s2[1]. Conditional on x3, x3s2 has mean bx3s2[2] * x3 and variance bx3s2[3].
bx12	Bernoulli probabilities for reference variables x1 and x2. A vector of length 2, default is c(0.25, 0.15). If mncor (see below) is positive the correlations between these Bernoulli and continuous predictors remains positive.
sd	In case of simfam == "gaussain" for linear regression, the sd of outcome y. In case of simfam == "Cox" for Cox PH regression, the multiplicative error term for ys, the surrogate for the time to event y (ys = log(sd * a (random variable) * y).
fewer	When set to 1 x3s1 and x4 will be collapsed to one variable in the surrogate set. This demonstrates how the method works when there are fewer surrogate variables than reference variables. If bx3s2 is specified such that there are duplicate surrogate variables for the reference variable x3 then the number of surrogate predictors will not be reduced.
mncor	Correlation of the columns in the x matrix before x1 and x2 are dichotomized to Bernoulli random variables. Default is 0.
sigma	A 4x4 varaince-covarniance matrix for the multivarite normal dsitribution used to derive the 4 reference predictor variables.
vmethod	Method for robust estimation of variance covariance matrices needed for calculation of the augmented estimates (beta aug). 0 for JK or jackknife (slowest but more accurate), 1 for IJK or the infinitesimal JK using the R default dfbeta's 2 for IJK using an alternate formula for the dfbeta, and 3 for all three of these methods to be used NA to let the program choose a stronger, faster method.

jksize	leave out number for grouped jackknife used for non validation data The default is 0 where the program chooses jksize so that the number of leave out groups is about validation subsample size.
compare	1 to compare gamma_val with gamma_ful (default) or 0 with gamma_non.
diffam	inidcates a cutoff if for a "guassian" family in surrogate a "binomial" famliy is to be similated for the refernce model. For example, the surrogate outcome could be an estimated probit (or logit) based upon a convolutional neural network. Normal data are simulated and y_val is repalced by 1*(y_val >= diffam). Default is NA and the surrogate and reference have the same model form. Only for use with vmethod of 0 or 1.
simtime	1 (default) to print out time duirng simulalation to inform user how long the simulation may run, 0 to not print out this information.

Value

meerva.sim.block returns a list object of class meerva.sim. The list will contain summary information used to simulate the data, and for each data set simulated with measurement error, the augmented estimates based upon the full data set accounting for measurement error, estimates based upon reference variables from the validation subsample, estimates based upon the surrogate variables from the whole sample, along with estimated variances for these estimates. These can be inspected by the user directly or by as shown in the example.

Author(s)

Walter Kremers (kremers.walter@mayo.edu)

See Also

meerva.fit , meerva.sim.brn , meerva.sim.cox , meerva.sim.nrm

Examples

```
# Simulation study for logistic reg data with
# differential misclassification in outcome
# and a predictor and measurement error in
# another predictor.  nsims=10 is as an
# example only.  Try running nsims=100 or
# 1000, but be prepared to wait a little while.
sim.binomial = meerva.sim.block(simfam="binomial",
    nsims=10, seed=0, n=4000, m=400,
    beta = c(-0.5, 0.5, 0.2, 1, 0.5) ,
    alpha1 = c(0.95, 0.90, 0.90, 0.95),
    alpha2 = c(0.98,0.98,0.95,0.95),
    bx3s1=c(0.05, 0, 0, NA, NA) ,
    bx3s2 = c(NA,NA,NA) ,
    vmethod=2, jksize=0, compare=1)

plot(sim.binomial)
summary(sim.binomial, 1)
```

```
# Simulation study for linear reg data.
# For this example there are more surrogate
# predictors than reference predictors.
# nsims=10 is as an example only.  Try
# running nsims=100 or 1000, but be
# prepared to wait a little while.
sim.gaussianm = meerva.sim.block(simfam="gaussian",
    nsims=10, seed=0, n=4000, m=400,
    beta = c(-0.5, 0.5, 0.2, 1, 0.5) ,
    alpha1 = c(-0.05, 0.1, 0.05, 0.1) ,
    alpha2 = c(0.98,0.94,0.95,0.95) ,
    bx3s1=c(0.05, 0, 0, NA, NA) ,
    bx3s2  = c(1.1,0.9,0.05) ,
    sd=1, fewer=0,
    vmethod=1, jksize=0, compare=1)

plot(sim.gaussianm)
summary(sim.gaussianm)

# Simulation study for Cox PH data.
# For this example there are fewer surrogates
# than reference variables yet they provide
# information to decrease the variance in the
# augmented estimate.  nsims=10 is as an
# example only.  Try running nsims=100 or
# 1000, but be prepared to wait a little
# while.
sim.coxphf = meerva.sim.block(simfam="Cox",
    nsims=10, seed=0, n=4000, m=400,
    beta   = c(-0.5, 0.5, 0.2, 1, 0.5) ,
    alpha1 = c(0.95,0.90,0.90,0.95)  ,
    alpha2 = c(0.98,0.94,0.94,0.98) ,
    bx3s1  = c(0.05,0,0,NA,NA) ,
    bx3s2  = c(1.1, NA, NA) ,
    sd=0.1, fewer=1,
    vmethod=1, jksize=0, compare=1 )

plot(sim.coxphf)
summary(sim.coxphf)
```

`meerva.sim.brn` *Simulate logistic Regression Data with Measurement Errors in Outcome and Predictors*

Description

The meerva package is designed to analyze data with measurement error when there is a validation subsample. The merva.sim.brn function generates a simulated data set for the logistic regression setting demonstrating the data form expected for input to the meervad.fit function. This simulation function first generates 4 reference predictors based upon a multivariate normal distribution,

with variance-covariance specified by the user. The first two predictors are dichotomized to have probabilites specified by the user. This results in two class and two quantitative reference predictor variables. The response variable may have a surrogate with differential misclassification error. There is one yes/no surrogate predictor variable involving error in place of one of the yes/no reference predictors, and one quantitative surrogate predictor variable involving error in place of one of the quantitative reference predictors. The simulated data are not necessarily realistic, but their analysis shows how even with rather strong measurement error the method yields reasonable solutions. The method is able to handle different types of measurement error without the user having to specify any relationship between the reference variables measured without error and the surrogate variables measured with error.

Usage

```
meerva.sim.brn(
  n = 4000,
  m = 400,
  beta = c(-0.5, 0.5, 0.2, 1, 0.5),
  alpha1 = c(1, 1, 1, 1),
  alpha2 = c(1, 1, 1, 1),
  bx3s1 = c(NA, NA, NA, NA, NA),
  bx3s2 = c(NA, NA, NA),
  fewer = 0,
  bx12 = c(0.25, 0.15),
  mncor = 0,
  sigma = NULL
)
```

Arguments

n	The full dataset size.
m	The validation subsample size (m < n).
beta	A vector of length 5 for the true regression parameter for the logistic regression model with 5 predictors including the intercept.
alpha1	A vector of length four determining the misclassification probabilities by the surrogate outcome, ys. if x1==1 then the probability of correct classification of true yes's is alpha1[1] and true no's is alpha1[2]. if x1==0 then the probability of correct classification of true yes's is alpha1[3] and true no's is alpha1[4].
alpha2	A vector describing the correct classification probabilities for x1s, the surrogate for x1. if y==1 then the probability of correct classification by the surrogate x1s is is alpha1[1] when x1==1, and alpha1[2] when x1==0. if y==0 then the probability of correct classification by the surrogate x1s is is alpha1[3] when x1==1, and alpha1[4] when x1==0.
bx3s1	A vector of length 5 determining the relation between the reference variable x3 and the mean and SD of the surrogate x3s1. Roughly, bx3s1[1] determines a minimal measurement error SD, conditional on x3 bx3s1[2] determines a rate of increase in SD for values of x3 greater than bx3s1[3], bx3s1[4] is a value above which the relation between x3 and the mean of x3s is determined by the power bx3s1[5]. The mean values for x3s1 are rescaled to have mean 0 and variance 1.

bx3s2	A vector of length 3 determining scale in x3s and potentially x3s2, a second surrogate for xs. Roughly, bx3s2[1] takes the previously determined mean for x3s1 using bx3s1 and multiples by bx3s2[1]. Conditional on x3, x3s2 has mean bx3s2[2] * x3 and variance bx3s2[3].
fewer	When set to 1 x3s1 and x4 will be collapsed to one variable in the surrogate set. This demonstrates how the method works when there are fewer surrogate variables than reference variables. If bx3s2 is specified such that there are duplicate surrogate variables for the reference variable x3 then the number of surrogate predictors will not be reduced.
bx12	Bernoulli probabilities for reference variables x1 and x2. A vector of length 2, default is c(0.25, 0.15). If mncor (see below) is positive the correlations between these Bernoulli and continuous predictors remains positive.
mncor	Correlation of the columns in the x matrix before x1 and x2 are dichotomized to Bernoulli random variables. Default is 0.
sigma	A 4x4 varaince-covarniance matrix for the multivarite normal dsitribution used to derive the 4 reference predictor variables.

Value

meerva.sim.brn returns a list containing vectors and matrices which can be used as example input to the meerva.fit function.

See Also

meerva.sim.block , meerva.sim.cox , meerva.sim.nrm , meerva.fit

Examples

```
# Logistic model with differential misclassification of outcome and a
# predictor and non constant measuremnt error in another predictor
simd = meerva.sim.brn(beta=c(-0.5, 0.5, 0.2, 1, 0.5),
  alpha1=c(0.90,0.95,0.95,0.90), alpha2=c(0.95,0.91,0.9,0.9),
  bx3s1=c(0.15,0.15,-1,-5,1), bx3s2=c(1,NA,NA)) ;

# Copy the data vectors and matrices to input to meerva.fit
x_val  = simd$x_val
y_val  = simd$y_val
xs_val = simd$xs_val
ys_val = simd$ys_val
xs_non = simd$xs_non
ys_non = simd$ys_non

# Analyze the data and display results
brnout = meerva.fit(x_val, y_val, xs_val, ys_val, xs_non, ys_non)
summary(brnout)
```

`meerva.sim.cox` *Simulate Cox Regression Data with Measurement Errors in Outcome and Predictors*

Description

The meerva package is designed to analyze data with measurement error when there is a validation subsample. The merva.sim.cox function generates a simulated data set for the Cox proportional hazards regression setting demonstrating the data form expected for input to the meervad.fit function. This simulation function first generates 4 reference predictors based upon a multivariate normal distribution, with variance-covariance specified by the user. The first two predictors are dichotomized to have probabilites specified by the user. This results in two class and two quantitative reference predictor variables. The yes/no event response variable may have a surrogate with differential misclassification. The time to event may have a surrogate measured with a multiplicative error. There is one yes/no surrogate predictor variable involving error in place of one of the yes/no reference predictors, and one quantitative surrogate predictor variable involving error in place of one of the quantitative reference predictors. The simulated data are not necessarily realistic, but their analysis shows how even with rather strong measurement error the method yields reasonable solutions. The method is able to handle different types of measurement error without the user having to specify any relationship between the reference variables measured without error and the surrogate variables measured with error.

Usage

```
meerva.sim.cox(
  n = 4000,
  m = 400,
  beta = c(-0.5, 0.5, 0.2, 1, 0.5),
  alpha1 = c(1, 1, 1, 1),
  alpha2 = c(1, 1, 1, 1),
  bx3s1 = c(NA, NA, NA, NA, NA),
  bx3s2 = c(NA, NA, NA),
  sd = 0,
  fewer = 0,
  bx12 = c(0.25, 0.15),
  mncor = 0,
  sigma = NULL
)
```

Arguments

`n`	The full dataset size.
`m`	The validation subsample size (m < n).
`beta`	A vector of length 5 determining the baseline hazard and proportional hazards of the simulated survival time data. For the Cox model beta[1] is not estimated but determines a baseline event rate.

`alpha1` A vector of length four determining the misclassification probabilities by the surrogate outcome, ys. if x1==1 then the probability of correct classification of true yes's is alpha1[1] and true no's is alpha1[2]. if x1==0 then the probability of correct classification of true yes's is alpha1[3] and true no's is alpha1[4].

`alpha2` A vector describing the correct classification probabilities for x1s, the surrogate for x1. if y==1 then the probability of correct classification by the surrogate x1s is is alpha1[1] when x1==1, and alpha1[2] when x1==0. if y==0 then the probability of correct classification by the surrogate x1s is is alpha1[3] when x1==1, and alpha1[4] when x1==0.

`bx3s1` A vector of length 5 determining the relation between the reference variable x3 and the mean and SD of the surrogate x3s1. Roughly, bx3s1[1] determines a minimal measurement error SD, conditional on x3 bx3s1[2] determines a rate of increase in SD for values of x3 greater than bx3s1[3], bx3s1[4] is a value above which the relation between x3 and the mean of x3s is determined by the power bx3s1[5]. The mean values for x3s1 are rescaled to have mean 0 and variance 1.

`bx3s2` A vector of length 3 determining scale in x3s and potentially x3s2, a second surrogate for xs. Roughly, bx3s2[1] takes the previously determined mean for x3s1 using bx3s1 and multiples by bx3s2[1]. Conditional on x3, x3s2 has mean bx3s2[2] * x3 and variance bx3s2[3].

`sd` The multiplicative error term for ys, the surrogate for the time to event y (ys = log(sd * a (random variable) * y).

`fewer` When set to 1 x3s1 and x4 will be collapsed to one variable in the surrogate set. This demonstrates how the method works when there are fewer surrogate variables than reference variables. If bx3s2 is specified such that there are duplicate surrogate variables for the reference variable x3 then the number of surrogate predictors will not be reduced.

`bx12` Bernoulli probabilities for reference variables x1 and x2. A vector of length 2, default is c(0.25, 0.15). If mncor (see below) is positive the correlations between these Bernoulli and continuous predictors remains positive.

`mncor` Correlation of the columns in the x matrix before x1 and x2 are dichotomized to Bernoulli random variables. Default is 0.

`sigma` A 4x4 varaince-covarniance matrix for the multivarite normal dsitribution used to derive the 4 reference predictor variables.

Value

meerva.sim.cox returns a list containing vectors and matrices which can be used as example input to the meerva.fit function.

See Also

`meerva.sim.block` , `meerva.sim.brn` , `meerva.sim.nrm` , `meerva.fit`

Examples

```
# Simulate Cox PH regression data with mesurement errors
simd = meerva.sim.cox(n=4000,m=400, beta = c(-0.5, 0.5, 0.2, 1, 0.5),
```

```
     alpha1=c(0.98,0.94, 0.94, 0.98), alpha2=c(0.95, 0.91, 0.9, 0.9),
     bx3s1=c(0.05, 0, 0, NA, NA), bx3s2 = c(1.1, 0.9, 0.05), sd=0.02 )

# Copy the data vectors and matrices to input to meerva.fit
x_val  = simd$x_val
y_val  = simd$y_val
xs_val = simd$xs_val
ys_val = simd$ys_val
xs_non = simd$xs_non
ys_non = simd$ys_non
e_val  = simd$e_val
es_val = simd$es_val
es_non = simd$es_non

# Analyze the data and display results
coxout = meerva.fit(x_val, y_val, xs_val, ys_val, xs_non, ys_non,
                  e_val, es_val, es_non)
summary(coxout)
```

`meerva.sim.nrm` *Simulate Linear Regression Data with Measurement Errors in Outcome and Predictors*

Description

The meerva package is designed to analyze data with measurement error when there is a validation subsample. The merva.sim.nrm function generates a simulated data set for the linear regression setting demonstrating the data form expected for input to the meervad.fit function. This simulation function first generates 4 reference predictors based upon a multivariate normal distribution, with variance-covariance specified by the user. The first two predictors are dichotomized to have probabilites specified by the user. This results in two class and two quantitative reference predictor variables. The response variable may have a surrogate with differential measurement error. There is one yes/no surrogate predictor variable involving error in place of one of the yes/no reference predictors, and one quantitative surrogate predictor variable involving error in place of one of the quantitative reference predictors. The simulated data are not necessarily realistic, but their analysis shows how even with rather strong measurement error the method yields reasonable solutions. The method is able to handle different types of measurement error without the user having to specify any relationship between the reference variables measured without error and the surrogate variables measured with error.

Usage

```
meerva.sim.nrm(
  n = 4000,
  m = 400,
  beta = c(-0.5, 0.5, 0.2, 1, 0.5),
  alpha1 = c(0, 0, 0, 0),
  alpha2 = c(1, 1, 1, 1),
```

```
  bx3s1 = c(NA, NA, NA, NA, NA),
  bx3s2 = c(NA, NA, NA),
  sd = 1,
  fewer = 0,
  bx12 = c(0.25, 0.15),
  mncor = 0,
  sigma = NULL
)
```

Arguments

n	The full dataset size.
m	The validation subsample size (m < n).
beta	A vector of length 5 for the true regression parameter for the linear regression model with 5 predictors including the intercept.
alpha1	a vector of length four determining the measurement error for the outcome. if x1==1 then the error has mean alpha1[1] and variance alpha1[2]. if x1==0 then the error has mean alpha1[3] and variance alpha1[4].
alpha2	A vector describing the correct classification probabilities for the surrogate for x1. If the outcome variable has positive error, then alpha2[1] and alpha2[2] are the probabilities of correct classification when x1 is 1 or 0. If the outcome variable has negative error, then alpha2[3] and alpha2[4] are the probabilities of correct classification when x1 is 1 or 0.
bx3s1	A vector of length 5 determining the relation between the reference variable x3 and the mean and SD of the surrogate x3s1. Roughly, bx3s1[1] determines a minimal measurement error SD, conditional on x3 bx3s1[2] determines a rate of increase in SD for values of x3 greater than bx3s1[3], bx3s1[4] is a value above which the relation between x3 and the mean of x3s is determined by the power bx3s1[5]. The mean values for x3s1 are rescaled to have mean 0 and variance 1.
bx3s2	A vector of length 3 determining scale in x3s and potentially x3s2, a second surrogate for xs. Roughly, bx3s2[1] takes the previously determined mean for x3s1 using bx3s1 and multiples by bx3s2[1]. Conditional on x3, x3s2 has mean bx3s2[2] * x3 and variance bx3s2[3].
sd	The sd of outcome y
fewer	When set to 1 x3s1 and x4 will be collapsed to one variable in the surrogate set. This demonstrates how the method works when there are fewer surrogate variables than reference variables. If bx3s2 is specified such that there are duplicate surrogate variables for the reference variable x3 then the number of surrogate predictors will not be reduced.
bx12	Bernoulli probabilities for reference variables x1 and x2. A vector of length 2, default is c(0.25, 0.15). If mncor (see below) is positive the correlations between these Bernoulli and continuous predictors remains positive.
mncor	Correlation of the columns in the x matrix before x1 and x2 are dichotomized to Bernoulli random variables. Default is 0.
sigma	A 4x4 varaince-covarniance matrix for the multivarite normal dsitribution used to derive the 4 reference predictor variables.

Value

meerva.sim.nrm returns a list containing vectors and matrices which can be used as example input to the meerva.fit function.

See Also

meerva.sim.block , meerva.sim.brn , meerva.sim.cox , meerva.fit

Examples

```
# Simulate linear regression data with mesurement errors
simd = meerva.sim.nrm(beta=c(-0.5, 0.5, 0.2, 1, 0.5),
    alpha1=c(-0.05, 0.1, 0.05, 0.1), alpha2=c(0.95, 0.91, 0.9, 0.9),
    bx3s1=c(0.05, 0, 0, NA, NA), bx3s2 = c(1.1, 0.9, 0.05) )

simd = meerva.sim.nrm(beta=c(-0.5, 0.5, 0.2, 1, 0.5),
    alpha1=c(-0.05, 0.1, 0.05, 0.1), alpha2=c(0.95, 0.91, 0.9, 0.9),
    bx3s1=c(0.05, 0, 0, NA, NA), bx3s2 = c(1.1, NA, NA), fewer=1 )
# Copy the data vectors and matrices to input to meerva.fit
x_val  = simd$x_val
y_val  = simd$y_val
xs_val = simd$xs_val
ys_val = simd$ys_val
xs_non = simd$xs_non
ys_non = simd$ys_non

# Analyze the data and display results
nrmout = meerva.fit(x_val, y_val, xs_val, ys_val, xs_non, ys_non )
summary(nrmout)
```

---

myttest *A simple summary description*

Description

A simple summary description

Usage

```
myttest(x, beta0 = NULL)
```

Arguments

x	A data matrix
beta0	A vector for H0

Value

A matrix

plot.meerva.sim *Plot results for meerva.sim output object generated by meerva.sim.block function*

Description

Plot results for meerva.sim output object generated by meerva.sim.block function

Usage

```
## S3 method for class 'meerva.sim'
plot(x, violin = 0, ...)
```

Arguments

x	A meerva.sim class object
violin	1 to produce a violin plot instead of a boxplot
...	further arguments

Value

This displays a plot

print.meerva *Print Minimal Summary Information for a meerva Output Object*

Description

Print Minimal Summary Information for a meerva Output Object

Usage

```
## S3 method for class 'meerva'
print(x, alpha = 0.05, round = NA, ...)
```

Arguments

x	A meerva class object for printing
alpha	level for (1-alpha) confidence intervals
round	number of decimal places to print for some outputs
...	further arguments

Value

Print output

print.meerva.sim *Print Information for a meerva.sim Simulation Study Output Object (alias for summary.meerva.fit())*

Description

Print Information for a meerva.sim Simulation Study Output Object (alias for summary.meerva.fit())

Usage

```
## S3 method for class 'meerva.sim'
print(x, short = 0, round = NA, ...)
```

Arguments

x	Output object from the simulations study program meerva.sim.block
short	0 to produce extensive output summary, 1 to produce only a table of biases and MSEs
round	number of decimal places to round to in some tables, NA for R default
...	further arguments

Value

A summary print

See Also

meerva.sim.block , summary.meerva.sim

reldif *Calculate relative differences*

Description

Calculate relative differences

Usage

```
reldif(a, b)
```

Arguments

a	Object 1 for comparison
b	Object 2 for comparison

Value

A object with relative differences

`subvec` *Subtract a vector from each row of a matrix*

Description

Subtract a vector from each row of a matrix

Usage

```
subvec(x, v)
```

Arguments

`x`	A matrix
`v`	A vector of length dim[x](2)

Value

A matrix

`summary.meerva` *Summarize Information for a meerva Output Object*

Description

Summarize Information for a meerva Output Object

Usage

```
## S3 method for class 'meerva'
summary(object, alpha = 0.05, round = NA, ...)
```

Arguments

`object`	A meerva class object for summary.
`alpha`	level for (1-alpha) confidence intervals
`round`	number of decimal places to print for some outputs
`...`	further arguments

Value

Summarize output

summary.meerva.sim *Summarize Information for a meerva.sim Simulation Study Output Object*

Description

Summarize Information for a meerva.sim Simulation Study Output Object

Usage

```
## S3 method for class 'meerva.sim'
summary(object, short = 0, round = NA, ...)
```

Arguments

object	Output object from the simulations study program meerva.sim.block
short	0 to produce extensive output summary, 1 to produce only a table of biases and MSEs
round	number of decimal places to round to in some tables, NA for R default
...	further arguments

Value

A summary print

See Also

meerva.sim.block , print.meerva.sim

ztest *Ztest for beta coefficients*

Description

Ztest for beta coefficients

Usage

```
ztest(estimate, var, names, alpha = 0.05, round = NA)
```

Arguments

estimate	beta estimates
var	variance of estimates
names	names of varaibles
alpha	level for 91-alpha) confidence intervals
round	number of decimal palces to rount some values

Value

table of summary statistics

Index

compmse, 3
coverage, 3
coverage2, 4

dfbetac, 4

meerva, 5
meerva.fit, *5*, 5, *14*, *17*, *19*, *22*
meerva.sim.block, *5*, *10*, 11, *17*, *19*, *22*, *24*, *26*
meerva.sim.brn, *5*, *10*, *14*, 15, *19*, *22*
meerva.sim.cox, *5*, *10*, *14*, *17*, 18, *22*
meerva.sim.nrm, *5*, *10*, *14*, *17*, *19*, 20
myttest, 22

plot.meerva.sim, 23
print.meerva, 23
print.meerva.sim, 24, *26*

reldif, 24

subvec, 25
summary.meerva, 25
summary.meerva.sim, *24*, 26

ztest, 26