Package ‘mmc’

October 13, 2022

Type Package

Title Multivariate Measurement Error Correction

Version 0.0.3

Date 2015-08-1

Author Jaejoon Song <jjsong2@mdanderson.org>

Maintainer Jaejoon Song <jjsong2@mdanderson.org>

Description Provides routines for multivariate measurement error correction. Includes procedures for linear, logistic and Cox regression models. Bootstrapped standard errors and confidence intervals can be obtained for corrected estimates.

Depends R (>= 2.10), survival (>= 2.38-1), MASS (>= 7.3), stats (>= 2.5.0)

Repository CRAN

NeedsCompilation no

Date/Publication 2015-08-12 00:41:06

R topics documented:

`mmc` *Multivariate measurement error correction*

Description

Multivariate measurement error correction for linear, logistic and Cox models.

Usage

```
mmc(model,type,mdata,rdata,rep,evar,rvar,bootstrap,boot)
```

Arguments

- `model` — Model is specified. For example, a Cox model can be specified as model = 'Surv(time,death) ~ x1'; a logistic regression model as model = 'glm(y ~ x1, family = 'binomial)'; a linear regression model as model = 'glm(y ~ x1, family = 'gaussian')'.
- `type` — Type of model is specified. Options are 'linear', 'logistic', or 'Cox'.
- `mdata` — Main study data. This dataset includes outcome variables and covariates measured with and without error, to be included in the model statement. Data for individuals with any missing data are deleted.
- `rdata` — Reliability data. This dataset includes repeated measurements for predictor variables measured with error. This data is used to estimate within-person covariance matrix. Reliability data can consist of repeated observations for a subset of individuals in the data. For instance, the main study data can have observations for 1000 individuals and the reliability data can consist of repeated measurements for 100 individuals in the main study data. Reliability data should contain at least two repeated measurements for each variable measured with error. It is assumed that variables in the reliability data have same number of repeated measurements. Data for individuals with any missing data are deleted.
- `rep` — Number of repeated measurements in the reliability data.
- `evar` — Variables measured with error in the main study data and in the model statement.
- `rvar` — Variable names in the reliability study data. It is assumed that the order of variables is same as it appears in the model statement. For example, if two predictor variables x1 and x2 are measured with error, rvar = c('x11','x12','x13','x21','x22','x23') would represent variable names for the three repeated measurements for x1 and x2, respectively.
- `bootstrap` — Specifies whether standard errors and 95 percent confidence intervals should be obtained for the corrected estimates using bootstrap resamples. By default, the bootstrap procedure is not implemented (i.e. only the corrected estimates are returned).
- `boot` — Number of bootstrap resamples to be used to obtain standard errors and confidence intervals. By default, the bootstrap procedure is not implemented (i.e. only the corrected estimates are returned).

Value

A list of returned, consisting of

- `uncorrected` — Uncorrected estimates.
- `total` — Total covariance matrix for variables measured with error. This is estimated from the main study data.
- `within` — Within person covariance matrix for variables measured with error. This is estimated from the reliability data.
- `between` — Between person covariance matrix for variables measured with error. This is estimated by subtracting the estimated within person covariance from the total covariance matrix.
- `corrected` — Corrected estimates. Standard errors and 95 percent confidence intervals are returned if bootstrap procedure is requested.

Note

For logistic regression and Cox models, the method of correction performed in this function is only recommended when: 1. The outcome is rare (disease probability less than 5 percent) 2. All predictors measured with error are continuous 3. The degree of measurement error is not severe (e.g. reliability coefficient > .5)

Author(s)

Jaejoon Song <jjsong2@mdanderson.org>

References

Rosner, B., Spiegelan D., and Willett W. C. (1992). Correction of logistic regression risk estimates and confidence intervals for random within-person measurement error. American Journal of Epidemiology, 136, 1400-13.

Rosner, B., and Gore R. (2001). Measurement error correction in nutritional epidemiology based on individual foods, with application to the relation of diet to breast cancer. American Journal of Epidemiology, 154, 827-35.

Spiegelman, D., Schneeweis, S., and McDermott, A. (1997). Measurement error correction for logistic regression models with an "Alloyed Gold Standard". American Journal of Epidemiology, 145, 184-86.

The SAS RELIBPLS Macro. (2015, July 1). Retrieved from
http://www.hsph.harvard.edu/donna-spiegelman/software/relibpls8

Examples

```
##########################################################################
## Example 1
## Generate data for linear regression
## assuming that one predictor variable is measured with error
##########################################################################

# Setting seed for replicability
set.seed(12345)
library(MASS)
library(mmc)

# Generating main dataset with 1000 observations
n <- 1000
# We generate data assuming that one predictor is measured with error
# Setting parameters for between person covariance matrix
var_between <- 1
# Setting parameters for within person covariance matrix
var_within <- .5
var_total <- var_within + var_between

# Generating data with a random variable
data_truth <- as.matrix(rnorm(n, -2, var_between),ncol=1)
# Generating measurement error for the random variable
measurement_error <-  as.matrix(rnorm(n, 0, var_within),ncol=1)
```

```
# The 'observed' data is constructed by adding the measurement error to the generated data
data_observed <- data_truth + measurement_error

# Setting true parameter values for linear regression
beta0 <- 1
beta1 <- 1

# Generate a continuous outcome variable
y <- beta0 + beta1*data_truth[1:n,1] + rnorm(n, 0,1)

# Set up a main dataset with the outcome variable and a predictor,
# measured with error.
datalin <- data.frame(y = y, x1 = data_observed[,1])

# Generating a reliability data.
# We assume that three repeated measurements are available for the predictor.
numrep <- 3
times <- 1:numrep
psi <- .8  # We assume that the repeated measurements have a AR(1) corr. structure
H <- abs(outer(times, times, "-"))

# Setting up a covariance structure for the reliability data.
Rcov <- var_between * psi^H

# We assume that the reliability dataset has
# data for only a subset of individuals in the main study data (n = 100)
# We assume that there are three repeated measurements for the predictor variable.
nr <- 100
mat <- mvrnorm(nr, Rcov, mu = c(rep(-2,numrep)), empirical = TRUE)

relibdata <- data.frame(x11 = mat[,1], x12 = mat[,2], x13 = mat[,3])

# Fitting a linear regression
glmfit.lin1 <- glm(y~x1, family="gaussian", data=datalin)
summary(glmfit.lin1)

# Using function mmc to get measurement error corrected estimates
rcfit.lin1 <- mmc(model = 'y~x1',
             type = 'linear',
             mdata = datalin,
             rdata <- relibdata,
             rep = 3,
             evar = c('x1'),
             rvar = c('x11','x12','x13'),
             bootstrap = 'FALSE')
rcfit.lin1


# Using function mmc to get measurement error corrected estimates
# with bootstrapped SE and confidence intervals
## Not run:
rcfit.lin1 <- mmc(model = 'y~x1',
             type = 'linear',
```

```
             mdata = datalin,
             rdata <- relibdata,
             rep = 3,
             evar = c('x1'),
             rvar = c('x11','x12','x13'),
             bootstrap = 'TRUE',
             boot <- 20)
rcfit.lin1

## End(Not run)





####################################################################
## Example 2
## Generate data for linear regression
## assuming that two predictor variables are measured with error
####################################################################

# Setting seed for replicability
set.seed(12345)
library(MASS)
library(mmc)

# Generating main dataset with 1000 observations
n <- 1000
# We generate data assuming that two predictors are measured with errors
# Setting parameters for between person covariance matrix
sigma1sq <- 1
sigma2sq <- 1
rho <- .3      # We assume that the two predictor variables are correlated
# Setting parameters for within person covariance matrix
theta1sq <- .3
theta2sq <- .3
# We assume that measurement errors for two predictor variables are independent
phi <- 0

cov_between <- matrix(
  c(sigma1sq,rho*sqrt(sigma1sq)*sqrt(sigma2sq),
  rho*sqrt(sigma1sq)*sqrt(sigma2sq), sigma2sq),2,2)

cov_within <- matrix(
  c(theta1sq,phi*sqrt(theta1sq)*sqrt(theta2sq),
  phi*sqrt(theta1sq)*sqrt(theta2sq), theta2sq),2,2)

cov_total <- cov_within + cov_between

# Generating data with two random variables
data_truth <- mvrnorm(n, c(-2,0.9), cov_between, empirical = TRUE)
# Generating measurement error for the two random variables
measurement_error <-  mvrnorm(n, c(0,0), cov_within, empirical = TRUE)
```

```
# The 'observed' data is constructed by adding the measurement error to the generated data
data_observed <- data_truth + measurement_error

# Setting true parameter values for linear regression
beta0 <- 1
beta1 <- 1
beta2 <- 2

# Generate a continuous outcome variable
y <- beta0 + beta1*data_truth[1:n,1] + beta2*data_truth[1:n,2] + rnorm(n, 0,1)

# Set up a main dataset with the outcome variable and the two predictors,
# measured with error.
datalin <- data.frame(y = y, x1 = data_observed[,1], x2 = data_observed[,2])

# Generating a reliability data.
# We assume that three repeated measurements are available, or each predictor.
numrep <- 3
times <- 1:numrep
psi <- .8  # We assume that the repeated measurements have a AR(1) corr. structure

H <- abs(outer(times, times, "-"))
R1 <- sigma1sq * psi^H
R2 <- sigma2sq * psi^H
RRcov <- sqrt(theta1sq)*sqrt(theta2sq)* rho * psi^H

# Setting up a covariance structure for the reliability data.
Rcov <- rbind(cbind(R1,RRcov),cbind(RRcov,R2))

# The reliability dataset has data for 100 individuals,
# three repeated measurements for each predictor variable.
nr <- 100
mat <- mvrnorm(nr, Rcov, mu = c(rep(-2,numrep),rep(0.9,numrep)), empirical = TRUE)

relibdata <- data.frame(x11 = mat[,1], x12 = mat[,2], x13 = mat[,3],
                x21 = mat[,4], x22 = mat[,5], x23 = mat[,6])

# Fitting a linear regression
glmfit.lin2 <- glm(y~x1+x2, family="gaussian", data=datalin)
summary(glmfit.lin2)

# Using function mmc to get measurement error corrected estimates
rcfit.lin2 <- mmc(model = 'y~x1+x2',
             type = 'linear',
             mdata = datalin,
             rdata <- relibdata,
             rep = 3,
             evar = c('x1','x2'),
             rvar = c('x11','x12','x13','x21','x22','x23'),
             bootstrap = 'FALSE')
rcfit.lin2

# Using function mmc to get measurement error corrected estimates
```

```
# with bootstrapped SE and confidence intervals
## Not run:
rcfit.lin2 <- mmc(model = 'y~x1+x2',
             type = 'linear',
             mdata = datalin,
             rdata <- relibdata,
             rep = 3,
             evar = c('x1','x2'),
             rvar = c('x11','x12','x13','x21','x22','x23'),
             bootstrap = 'FALSE')
rcfit.lin2

## End(Not run)



####################################################################
## Example 3
## Generate data for logistic regression
## assuming that one predictor variable is measured with error
####################################################################

# Setting seed for replicability
set.seed(12345)
library(MASS)
library(mmc)

# Generating main dataset with 10,000 observations
n <- 10000
# We generate data assuming that a predictor was measured with error
# Setting parameter for between person variance
var_between <- 1
# Setting parameters for within person variance
var_within <- .3
var_total <- var_within + var_between

# Generating data with a random variable
data_truth <- as.matrix(rnorm(n, -2, var_between),ncol=1)
# Generating measurement error for the random variable
measurement_error <-  as.matrix(mvrnorm(n, 0, var_within),ncol=1)
# The 'observed' data is constructed by adding the measurement error to the generated data
data_observed <- data_truth + measurement_error

# Setting true parameter values for logistic regression
beta0 <- -2
beta1 <- 1

linpred <- beta0 + beta1*data_truth[1:n,1]
prob = exp(linpred)/(1 + exp(linpred))
runis = runif(n,0,1)
y = ifelse(runis < prob,1,0)

data <- data.frame(y = y, x1 = data_observed[,1])
```

```
# Generating a reliability data.
# We assume that three repeated measurements are available for the predictor.
numrep <- 3
times <- 1:numrep
psi <- .8

H <- abs(outer(times, times, "-"))
Rcov<- var_between * psi^H

# The reliability dataset has data for 100 individuals,
# three repeated measurements for the predictor variable.
nr <- 100
mat <- mvrnorm(nr, Rcov, mu = rep(-2,numrep), empirical = TRUE)
d <- data.frame(x11 = mat[,1], x12 = mat[,2], x13 = mat[,3])

# Fitting a logistic model
glmfit <- glm( y~x1, family="binomial",data=data)
summary(glmfit)

# Using function mmc to get measurement error corrected estimates
rcfit <- mmc(
model = 'y ~ x1',
type = 'logistic',
mdata = data,
rdata <- d,
rep = 3,
evar = c('x1'),
rvar = c('x11','x12','x13'),
bootstrap = 'FALSE')

rcfit

# Using function mmc to get measurement error corrected estimates
# with bootstrapped SE and confidence intervals
## Not run:
rcfit <- mmc(
model = 'y ~ x1',
type = 'logistic',
mdata = data,
rdata <- d,
rep = 3,
evar = c('x1'),
rvar = c('x11','x12','x13'),
bootstrap = 'TRUE',
boot <- 20)

rcfit

## End(Not run)
```

```
################################################################
## Example 4
## Generate data for logistic regression
## assuming that two predictor variables are measured with error
################################################################

# Setting seed for replicability
set.seed(12345)
library(MASS)
library(mmc)

# Generating main dataset with 10,000 observations
n <- 10000
# We generate data assuming that two predictors are measured with errors
# Setting parameters for between person covariance matrix
sigma1sq <- 1
sigma2sq <- 1
rho <- .3      # We assume that two predictor variables are correlated
# Setting parameters for within person covariance matrix
theta1sq <- .3
theta2sq <- .3
# We assume that measurement errors for two predictor variables are independent
phi <- 0

cov_between <- matrix(
c(sigma1sq,rho*sqrt(sigma1sq)*sqrt(sigma2sq),
  rho*sqrt(sigma1sq)*sqrt(sigma2sq),sigma2sq),2,2)

cov_within <- matrix(
c(theta1sq,phi*sqrt(theta1sq)*sqrt(theta2sq),
  phi*sqrt(theta1sq)*sqrt(theta2sq),theta2sq),2,2)

cov_total <- cov_within + cov_between

# Generating data with two random variables
data_truth <- mvrnorm(n, c(-2,0.9), cov_between, empirical = TRUE)
# Generating measurement error for the two random variables
measurement_error <-  mvrnorm(n, c(0,0), cov_within, empirical = TRUE)
# The 'observed' data is constructed by adding the measurement error to the generated data
data_observed <- data_truth + measurement_error

# Setting true parameter values for logistic regression
beta0 <- -6
beta1 <- 1
beta2 <- 2

linpred <- beta0 + beta1*data_truth[1:n,1] + beta2*data_truth[1:n,2]
prob = exp(linpred)/(1 + exp(linpred))
runis = runif(n,0,1)
y = ifelse(runis < prob,1,0)

data <- data.frame(y = y, x1 = data_observed[,1], x2 = data_observed[,2])
```

```
# Generating a reliability data.
# We assume that three repeated measurements are available, or each predictor.
numrep <- 3
times <- 1:numrep
psi <- .8

H <- abs(outer(times, times, "-"))
R1 <- sigma1sq * psi^H
R2 <- sigma2sq * psi^H
RRcov <- sqrt(theta1sq)*sqrt(theta2sq)* rho * psi^H

# Setting up a covariance structure for the reliability data.
Rcov <- rbind(cbind(R1,RRcov),cbind(RRcov,R2))

# The reliability dataset has data for 100 individuals,
# three repeated measurements for each predictor variable.
nr <- 100
mat <- mvrnorm(nr, Rcov, mu = c(rep(-2,numrep),rep(0.9,numrep)), empirical = TRUE)
d <- data.frame(x11 = mat[,1], x12 = mat[,2], x13 = mat[,3],
                x21 = mat[,4], x22 = mat[,5], x23 = mat[,6])

# Fitting a logistic model
glmfit <- glm( y~x1+x2, family="binomial",data=data)
summary(glmfit)

# Using function mmc to get measurement error corrected estimates
rcfit <- mmc(
model = 'y ~ x1 + x2',
type = 'logistic',
mdata = data,
rdata <- d,
rep = 3,
evar = c('x1','x2'),
rvar = c('x11','x12','x13','x21','x22','x23'),
bootstrap = 'FALSE')

rcfit

# Using function mmc to get measurement error corrected estimates
# with bootstrapped SE and confidence intervals
## Not run:
rcfit <- mmc(
model = 'y ~ x1 + x2',
type = 'logistic',
mdata = data,
rdata <- d,
rep = 3,
evar = c('x1','x2'),
rvar = c('x11','x12','x13','x21','x22','x23'),
bootstrap = 'TRUE',
boot <- 20)
```

```
rcfit

## End(Not run)



#################################################################
## Example 5
## Generate data for Cox regression
## assuming that one predictor variable is measured with error
#################################################################

# Setting seed for replicability
set.seed(1234)
library(MASS)
library(survival)
library(mmc)

# Generating main dataset with 10,000 observations
n <- 10000
# We generate data assuming that a predictor is measured with error
# Setting parameters for between person variance
var_between <- 1
# Setting parameters for within person variance
var_within <- .5
var_total <- var_within + var_between

# Generating data with a random variable
data_truth <- as.matrix(rnorm(n, -2, var_between),ncol=1)
# Generating measurement error for the random variable
measurement_error <-  as.matrix(rnorm(n, 0, var_within),ncol=1)
# The 'observed' data is constructed by adding the measurement error to the generated data
data_observed <- data_truth + measurement_error

# Setting true parameter values for linear regression
beta0 <- 2
beta1 <- 1

# Now set up some parameters for the Cox model
lambdaT = 0.1 # baseline hazard
lambdaC = 0.002 # hazard of censoring

# Setting up a Cox model
hazard <- exp(beta0 + beta1*data_truth[1:n,1])
# Generate event time from a Weibull distribution
Y <- rweibull(n, shape=1, scale=lambdaT/hazard)
# Generate random censoring time from a Weibull distribution
ctime <- rweibull(n, shape=1, scale = lambdaC)

# Now set up censoring variable
status <- numeric(n)+1
status[Y>ctime] <- 0 # 0=censored (event not observed), 1=not censored (event observed)
y <- pmin(Y, ctime)
```

```
# Set up a main dataset with the outcome variable and the predictor,
# measured with error.
datacox <- data.frame(y = y, status = status, x1 = data_observed[,1])

# Generating a reliability data.
# We assume that three repeated measurements are available for the predictor.
numrep <- 3
times <- 1:numrep
psi <- .8

H <- abs(outer(times, times, "-"))
Rcov <- var_between * psi^H

# The reliability dataset has data for 100 individuals,
# three repeated measurements for the predictor variable.
nr <- 100
mat <- mvrnorm(nr, Rcov, mu = rep(-2,numrep), empirical = TRUE)
d <- data.frame(x11 = mat[,1], x12 = mat[,2], x13 = mat[,3])

# Fitting a Cox model
coxfit <- coxph( Surv(y,status)~x1, method="breslow",data=datacox)
summary(coxfit)

# Using function mmc to get measurement error corrected estimates
rcfit <- mmc(
model = 'Surv(y,status) ~ x1',
type = 'cox',
mdata = datacox,
rdata <- d,
rep = 3,
evar = c('x1'),
rvar = c('x11','x12','x13'),
bootstrap = 'FALSE')

rcfit


# Using function mmc to get measurement error corrected estimates
# with bootstrapped SE and confidence intervals
## Not run:
rcfit <- mmc(
model = 'Surv(y,status) ~ x1',
type = 'cox',
mdata = datacox,
rdata <- d,
rep = 3,
evar = c('x1'),
rvar = c('x11','x12','x13'),
bootstrap = 'TRUE',
boot <- 20)

rcfit
```

```
## End(Not run)



######################################################################
## Example 6
## Generate data for Cox regression
## assuming that two predictor variables are measured with error
######################################################################

# Setting seed for replicability
set.seed(12345)
library(MASS)
library(survival)
library(mmc)

# Generating main dataset with 10,000 observations
n <- 10000
# We generate data assuming that two predictors are measured with errors
# Setting parameters for between person covariance matrix
sigma1sq <- 1
sigma2sq <- 1
rho <- .3      # We assume that two predictor variables are correlated
# Setting parameters for within person covariance matrix
theta1sq <- .3
theta2sq <- .3
# We assume that measurement errors for two predictor variables are independent
phi <- 0


cov_between <- matrix(
c(sigma1sq,rho*sqrt(sigma1sq)*sqrt(sigma2sq),
  rho*sqrt(sigma1sq)*sqrt(sigma2sq),sigma2sq),2,2)

cov_within <- matrix(
c(theta1sq,phi*sqrt(theta1sq)*sqrt(theta2sq),
  phi*sqrt(theta1sq)*sqrt(theta2sq),theta2sq),2,2)

cov_total <- cov_within + cov_between

# Generating data with two random variables
data_truth <- mvrnorm(n, c(-2,0.9), cov_between, empirical = TRUE)
# Generating measurement error for the two random variables
measurement_error <-  mvrnorm(n, c(0,0), cov_within, empirical = TRUE)
# The 'observed' data is constructed by adding the measurement error to the generated data
data_observed <- data_truth + measurement_error

# Setting true parameter values for Cox regression
beta0 <- 0.05
beta1 <- 1
beta2 <- 2
```

```
# Now set up some parameters for the Cox model
lambdaT = 0.1 # baseline hazard
lambdaC = 0.0001 # hazard of censoring

# Setting up a Cox model
hazard <- exp(beta0 + beta1*data_truth[1:n,1] + beta2*data_truth[1:n,2])
# Generate event time from a Weibull distribution
Y <- rweibull(n, shape=1, scale=lambdaT/hazard)
# Generate random censoring time from a Weibull distribution
ctime <- rweibull(n, shape=1, scale = lambdaC)

# Now set up censoring variable
status <- numeric(n)+1
status[Y>ctime] <- 0 # 0=censored (event not observed), 1=not censored (event observed)
y <- pmin(Y, ctime)

# Set up a main dataset with the outcome variable and the two predictors,
# measured with error.
datacox <- data.frame(y = y, status = status, x1 = data_observed[,1], x2 = data_observed[,2])

# Generating a reliability data.
# We assume that three repeated measurements are available, or each predictor.
numrep <- 3
times <- 1:numrep
psi <- .8

H <- abs(outer(times, times, "-"))
R1 <- sigma1sq * psi^H
R2 <- sigma2sq * psi^H
RRcov <- sqrt(theta1sq)*sqrt(theta2sq)* rho * psi^H

# Setting up a covariance structure for the reliability data.
Rcov <- rbind(cbind(R1,RRcov),cbind(RRcov,R2))

# The reliability dataset has data for 100 individuals,
# three repeated measurements for each predictor variable.
nr <- 100
mat <- mvrnorm(nr, Rcov, mu = c(rep(-2,numrep),rep(0.9,numrep)), empirical = TRUE)
d <- data.frame(x11 = mat[,1], x12 = mat[,2], x13 = mat[,3],
                x21 = mat[,4], x22 = mat[,5], x23 = mat[,6])

# Fitting a Cox model
coxfit <- coxph( Surv(y,status)~x1+x2, method="breslow",data=datacox)
summary(coxfit)

# Using function mmc to get measurement error corrected estimates
rcfit <- mmc(
model = 'Surv(y,status) ~ x1 + x2',
type = 'cox',
mdata = datacox,
rdata <- d,
rep = 3,
```

```
evar = c('x1','x2'),
rvar = c('x11','x12','x13','x21','x22','x23'),
bootstrap = 'FALSE')

rcfit



# Using function mmc to get measurement error corrected estimates
# with bootstrapped SE and confidence intervals
## Not run:
rcfit <- mmc(
model = 'Surv(y,status) ~ x1 + x2',
type = 'cox',
mdata = datacox,
rdata <- d,
rep = 3,
evar = c('x1','x2'),
rvar = c('x11','x12','x13','x21','x22','x23'),
bootstrap = 'TRUE',
boot <- 20)

rcfit

## End(Not run)
```

Index