

Package ‘RCreliability’

October 12, 2022

Type Package

Title Correct Bias in Estimated Regression Coefficients

Version 0.1.0

Description This function corrects the bias in estimated regression coefficients due to classical additive measurement error (i.e., within-person variation) in logistic regressions under the main study/external reliability study design and the main study/internal reliability study design. The output includes the naive and corrected estimators for the regression coefficients; for the variance estimates of the corrected estimators, the extra variation due to estimating the parameters in the measurement error model is ignored or taken into account. Reference: Carroll RJ, Ruppert D, Stefanski L, Crainiceanu CM (2006) <[doi:10.1201/9781420010138](https://doi.org/10.1201/9781420010138)>.

License GPL-3

Encoding UTF-8

RoxygenNote 7.1.2

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

Imports sandwich, mgcv

Config/testthat/edition 3

VignetteBuilder knitr

Author Yu Lu [aut, cre, cph],
Molin Wang [aut]

Maintainer Yu Lu <yu.lu@hsph.harvard.edu>

Repository CRAN

NeedsCompilation no

Date/Publication 2021-12-10 09:00:30 UTC

R topics documented:

RCreliability.ex	2
RCreliability.in	5

Index	9
--------------	----------

RCreliability.ex *RCreliability.ex*

Description

This function corrects the bias in estimated regression coefficients due to classical additive measurement error (i.e., within-person variation) in logistic regressions under the main study/external reliability study design. The output includes naive and corrected estimators for the regression coefficients; for the variance estimates of the corrected estimators, the extra variation due to estimating the parameters in the measurement error model is ignored or taken into account.

Usage

```
RCreliability.ex(z.main, r, z.rep, W=NULL, Y)
```

Arguments

<code>z.main</code>	covariates measured with error in the main study, a $n \times p$ matrix, where n is the number of subjects in the main study and p is the number of covariates.
<code>r</code>	number of replicates in the reliability study, vector of length m , where m is the number of subjects in the reliability study. Note: For each subject, the covariates with error in the reliability study should have the same number of replicates.
<code>z.rep</code>	covariates measured with error in the reliability study, a list with p elements, each element in a form of a $m \times \max(r)$ matrix; subjects with less observations than $\max(r)$ should also have $\max(r)$ columns with the unobserved elements filled with NA.
<code>W</code>	covariates without measurement errors, a $n \times q$ matrix, where q stands for the number of covariates without measurement errors. Default is NULL.
<code>Y</code>	response variable in the main study, vector of length n . Values should be 0 or 1 in this logistic regression setting.

Value

A list with 3 table of regression statistics.

Naive estimates

Estimates of regression coefficients ignoring the measurement errors.

Corrected estimates

Regression calibration estimates without taking into account the extra variation due to estimating the parameters in the measurement error model.

Corrected estimates, taking into account the extra variation due to estimating the parameters in the measurement error model.

Regression calibration estimates taking into account the extra variation due to estimating the parameters in the measurement error model.

Author(s)

Yu Lu, Molin Wang

References

Carroll RJ, Ruppert D, Stefanski L, Crainiceanu CM. Measurement Error in Nonlinear Models: A Modern Perspective. 2nd ed. New York: Chapman & Hall/CRC; 2006

See Also

RCreliability.in function

Examples

```
library(RCreliability)
library(mgcv)

# Regression on only one covariates measured with error
x<-rnorm(3000,0,1)
#ICC=0.7 generate z
z.main <- matrix(x[1:1500]+rnorm(1500,0,sqrt(0.4)))
r<-c(rep(3,700),rep(4,800))
z.rep<-list(rbind(cbind(x[1501:2200]+rnorm(700,0,sqrt(0.4)),
x[1501:2200]+rnorm(700,0,sqrt(0.4)),
x[1501:2200]+rnorm(700,0,sqrt(0.4)),NA),
cbind(x[2201:3000]+rnorm(800,0,sqrt(0.4)),
x[2201:3000]+rnorm(800,0,sqrt(0.4)),
x[2201:3000]+rnorm(800,0,sqrt(0.4)),
x[2201:3000]+rnorm(800,0,sqrt(0.4)))))
#prevalence about 0.105
p<-exp(-2.2+log(1.5)*x[1:1500])/
(1+exp(-2.2+log(1.5)*x[1:1500]))
Y<-sapply(p,function(x) rbinom(1,1,x))
fit1 <- RCreliability.ex(z.main,r,z.rep,W=NULL,Y)
fit1

# Regression on one covariates measured with error and one confounder
x<-rnorm(3000,0,1)
#ICC=0.7 generate z
z.main <- matrix(x[1:1500]+rnorm(1500,0,sqrt(0.4)))
r<-c(rep(3,700),rep(4,800))
z.rep<-list(rbind(cbind(x[1501:2200]+rnorm(700,0,sqrt(0.4)),
x[1501:2200]+rnorm(700,0,sqrt(0.4)),
x[1501:2200]+rnorm(700,0,sqrt(0.4)),NA),
cbind(x[2201:3000]+rnorm(800,0,sqrt(0.4)),
x[2201:3000]+rnorm(800,0,sqrt(0.4)),
x[2201:3000]+rnorm(800,0,sqrt(0.4)),
x[2201:3000]+rnorm(800,0,sqrt(0.4)))))
W<-matrix(sapply(x[1:1500], function(t){if(t>median(x)) {return(rbinom(1,1,0.5))}
if(t<=median(x)){return(rbinom(1,1,0.3))}}))
#prevalence about 0.103
p<-exp(-2.4+log(1.5)*x[1:1500]+log(1.5)*W)/
(1+exp(-2.4+log(1.5)*x[1:1500]+log(1.5)*W))
Y<-sapply(p,function(x) rbinom(1,1,x))
fit2<-RCreliability.ex(z.main,r,z.rep,W=W,Y)
```

```

fit2

# Regression on two covariates measured with error and no confounder
x<-rmvn(3000,c(0,0),matrix(c(1,0.3,0.3,1),nrow=2))
#ICC=0.7 generate z
z.main = x[1:1500,1:2]+rnorm(1500,0,sqrt(0.4))
r<-c(rep(2,500),rep(3,400),rep(4,600))
z.rep<-list(rbind(cbind(x[1501:2000,1]+rnorm(500,0,sqrt(0.4))),
x[1501:2000,1]+rnorm(500,0,sqrt(0.4)),NA,NA),
           cbind(x[2001:2400,1]+rnorm(400,0,sqrt(0.4)),
x[2001:2400,1]+rnorm(400,0,sqrt(0.4)),
x[2001:2400,1]+rnorm(400,0,sqrt(0.4)),NA),
           cbind(x[2401:3000,1]+rnorm(600,0,sqrt(0.4)),
x[2401:3000,1]+rnorm(600,0,sqrt(0.4)),
x[2401:3000,1]+rnorm(600,0,sqrt(0.4)),
x[2401:3000,1]+rnorm(600,0,sqrt(0.4)))))
rbind(cbind(x[1501:2000,2]+rnorm(500,0,sqrt(0.4)),
x[1501:2000,2]+rnorm(500,0,sqrt(0.4)),NA,NA),
      cbind(x[2001:2400,2]+rnorm(400,0,sqrt(0.4)),
x[2001:2400,2]+rnorm(400,0,sqrt(0.4)),
x[2001:2400,2]+rnorm(400,0,sqrt(0.4)),NA),
      cbind(x[2401:3000,2]+rnorm(600,0,sqrt(0.4)),
x[2401:3000,2]+rnorm(600,0,sqrt(0.4)),
x[2401:3000,2]+rnorm(600,0,sqrt(0.4)),
x[2401:3000,2]+rnorm(600,0,sqrt(0.4)))))
#prevalence about 0.105
p<-exp(-2.3+log(1.5)*rowSums(x[1:1500,]))/
(1+exp(-2.3+log(1.5)*rowSums(x[1:1500,])))
Y<-sapply(p,function(x) rbinom(1,1,x))
fit3<-RCreliability.ex(z.main,r,z.rep,W=NULL,Y)
fit3

# Regression on two covariates measured with error and two confounders
x<-rmvn(3000,c(0,0,0),matrix(c(1,0.3,0.2,0.3,1,0.5,0.2,0.5,1),nrow=3))
w2<-sapply(x[,1], function(t){if(t>median(x[,1])) {return(rbinom(1,1,0.5))}
if(t<=median(x[,1])){return(rbinom(1,1,0.3))}})
#ICC=0.7 generate z
r<-c(rep(2,500),rep(3,400),rep(4,600))
W<-cbind(x[1:1500,3],w2[1:1500])

z.main = x[1:1500,1:2]+rnorm(1500,0,sqrt(0.4))

z.rep<-list(rbind(cbind(x[1501:2000,1]+rnorm(500,0,sqrt(0.4))),
x[1501:2000,1]+rnorm(500,0,sqrt(0.4)),NA,NA),
           cbind(x[2001:2400,1]+rnorm(400,0,sqrt(0.4)),
x[2001:2400,1]+rnorm(400,0,sqrt(0.4)),
x[2001:2400,1]+rnorm(400,0,sqrt(0.4)),NA),
           cbind(x[2401:3000,1]+rnorm(600,0,sqrt(0.4)),
x[2401:3000,1]+rnorm(600,0,sqrt(0.4)),
x[2401:3000,1]+rnorm(600,0,sqrt(0.4)),
x[2401:3000,1]+rnorm(600,0,sqrt(0.4)))))
rbind(cbind(x[1501:2000,2]+rnorm(500,0,sqrt(0.4)),
x[1501:2000,2]+rnorm(500,0,sqrt(0.4)),NA,NA),
      cbind(x[2001:2400,2]+rnorm(400,0,sqrt(0.4)),
x[2001:2400,2]+rnorm(400,0,sqrt(0.4)),
x[2001:2400,2]+rnorm(400,0,sqrt(0.4)),NA),
      cbind(x[2401:3000,2]+rnorm(600,0,sqrt(0.4)),
x[2401:3000,2]+rnorm(600,0,sqrt(0.4)),
x[2401:3000,2]+rnorm(600,0,sqrt(0.4)),
x[2401:3000,2]+rnorm(600,0,sqrt(0.4)))))

```

```

      cbind(x[2001:2400,2]+rnorm(400,0,sqrt(0.4)),
      x[2001:2400,2]+rnorm(400,0,sqrt(0.4)),
      x[2001:2400,2]+rnorm(400,0,sqrt(0.4)),NA),
      cbind(x[2401:3000,2]+rnorm(600,0,sqrt(0.4)),
      x[2401:3000,2]+rnorm(600,0,sqrt(0.4)),
      x[2401:3000,2]+rnorm(600,0,sqrt(0.4)),
      x[2401:3000,2]+rnorm(600,0,sqrt(0.4))))
#prevalence about 0.105
p<-exp(-2.7+log(1.5)*rowSums(x[1:1500,1:3])+log(1.5)*w2[1:1500])/
(1+exp(-2.7+log(1.5)*rowSums(x[1:1500,1:3])+log(1.5)*w2[1:1500]))
Y<-sapply(p,function(x) rbinom(1,1,x))[1:1500]
fit4<-RCreliability.ex(z.main,r,z.rep,W=W,Y)
fit4

```

RCreliability.in

RCreliability.in

Description

This function corrects the bias in estimated regression coefficients due to classical additive measurement error (i.e., within-person variation) in logistic regressions under the partially or fully replicated design. The output includes naive and corrected estimators for the regression coefficients; for the variance estimates of the corrected estimators, the extra variation due to estimating the parameters in the measurement error model is ignored or taken into account.

Usage

```
RCreliability.in(r, z, W=NULL, Y)
```

Arguments

- | | |
|----------|---|
| <i>r</i> | number of replicates in the reliability study, vector of length <i>n</i> , where <i>n</i> is the number of subjects in the reliability study. Note: For each subject, the covariates with error in the reliability study should have the same number of replicates. |
| <i>z</i> | covariates measured with error in the reliability study, a list with <i>p</i> elements, each element in a form of a $n \times \max(r)$ matrix; subjects with less observations than $\max(r)$ should also have $\max(r)$ columns with the unobserved elements filled with NA. |
| <i>W</i> | covariates without measurement errors, a $n \times q$ matrix, where <i>q</i> stands for the number of covariates without measurement errors. Default is NULL. |
| <i>Y</i> | response variable in the main study, vector of length <i>n</i> . Values should be 0 or 1 in this logistic regression setting. |

Value

A list with 3 table of regression statistics.

Naive estimates

Estimates of regression coefficients ignoring the measurement errors.

Corrected estimates

Regression calibration estimates without taking into account the extra variation due to estimating the parameters in the measurement error model.

Corrected estimates, taking into account the extra variation due to estimating the parameters in the measurement error model.

Regression calibration estimates taking into account the extra variation due to estimating the parameters in the measurement error model.

Author(s)

Yu Lu, Molin Wang

References

Carroll RJ, Ruppert D, Stefanski L, Crainiceanu CM. Measurement Error in Nonlinear Models: A Modern Perspective. 2nd ed. New York: Chapman & Hall/CRC; 2006

See Also

RCreliability.ex function

Examples

```
library(RCreliability)
library(mgcv)

# Regression on only one covariates measured with error
x<-rnorm(3000,0,1)
#ICC=0.7 generate z
r<-c(rep(1,1500),rep(3,700),rep(4,800))
z<-list(rbind(cbind(x[1:1500]+rnorm(1500,0,sqrt(0.4)),NA,NA,NA),
              cbind(x[1501:2200]+rnorm(700,0,sqrt(0.4)),
                    x[1501:2200]+rnorm(700,0,sqrt(0.4)),
                    x[1501:2200]+rnorm(700,0,sqrt(0.4)),NA),
              cbind(x[2201:3000]+rnorm(800,0,sqrt(0.4)),
                    x[2201:3000]+rnorm(800,0,sqrt(0.4)),
                    x[2201:3000]+rnorm(800,0,sqrt(0.4))))))

#prevalence=0.105
p<-exp(-2.2+log(1.5)*x)/(1+exp(-2.2+log(1.5)*x))
Y<-sapply(p,function(x) rbinom(1,1,x))
fit1 <- RCreliability.in(r,z,W=NULL,Y)
fit1

# Regression on one covariates measured with error and one confounder
```

```

x<-rnorm(3000,0,1)
#ICC=0.7 generate z
r<-c(rep(1,1500),rep(3,700),rep(4,800))
z<-list(rbind(cbind(x[1:1500]+rnorm(1500,0,sqrt(0.4)),NA,NA,NA),
              cbind(x[1501:2200]+rnorm(700,0,sqrt(0.4)),
                    x[1501:2200]+rnorm(700,0,sqrt(0.4)),
                    x[1501:2200]+rnorm(700,0,sqrt(0.4)),NA),
              cbind(x[2201:3000]+rnorm(800,0,sqrt(0.4)),
                    x[2201:3000]+rnorm(800,0,sqrt(0.4)),
                    x[2201:3000]+rnorm(800,0,sqrt(0.4))))))
W<-sapply(x, function(t){if(t>median(x)) {return(rbinom(1,1,0.5))}
              if(t<=median(x)){return(rbinom(1,1,0.3))}})
#prevalence about 0.104
p<-exp(-2.4+log(1.5)*x+log(1.5)*W)/(1+exp(-2.4+log(1.5)*x+log(1.5)*W))
Y<-sapply(p,function(x) rbinom(1,1,x))
fit2<-RCreliability.in(r,z,W=W,Y)
fit2

# Regression on two covariates measured with error and no confounder
x<-rmvn(3000,c(0,0),matrix(c(1,0.3,0.3,1),nrow=2))
#ICC=0.7 generate z
r<-c(rep(1,1500),rep(2,500),rep(3,400),rep(4,600))
z<-list(rbind(cbind(x[1:1500,1]+rnorm(1500,0,sqrt(0.4)),NA,NA,NA),
              cbind(x[1501:1500,1]+rnorm(500,0,sqrt(0.4)),
                    x[1501:1500,1]+rnorm(500,0,sqrt(0.4)),NA,NA),
              cbind(x[2001:2400,1]+rnorm(400,0,sqrt(0.4)),
                    x[2001:2400,1]+rnorm(400,0,sqrt(0.4)),
                    x[2001:2400,1]+rnorm(400,0,sqrt(0.4)),NA),
              cbind(x[2401:3000,1]+rnorm(600,0,sqrt(0.4)),
                    x[2401:3000,1]+rnorm(600,0,sqrt(0.4)),
                    x[2401:3000,1]+rnorm(600,0,sqrt(0.4))))),
        rbind(cbind(x[1:1500,2]+rnorm(1500,0,sqrt(0.4)),NA,NA,NA),
              cbind(x[1501:1500,2]+rnorm(500,0,sqrt(0.4)),
                    x[1501:1500,2]+rnorm(500,0,sqrt(0.4)),NA,NA),
              cbind(x[2001:2400,2]+rnorm(400,0,sqrt(0.4)),
                    x[2001:2400,2]+rnorm(400,0,sqrt(0.4)),
                    x[2001:2400,2]+rnorm(400,0,sqrt(0.4)),NA),
              cbind(x[2401:3000,2]+rnorm(600,0,sqrt(0.4)),
                    x[2401:3000,2]+rnorm(600,0,sqrt(0.4)),
                    x[2401:3000,2]+rnorm(600,0,sqrt(0.4))))))
#prevalence about 0.105
p<-exp(-2.3+log(1.5)*rowSums(x))/(1+exp(-2.3+log(1.5)*rowSums(x)))
Y<-sapply(p,function(x) rbinom(1,1,x))
fit3<-RCreliability.in(r,z, W=NULL,Y)
fit3

# Regression on two covariates measured with error and two confounders
x<-rmvn(3000,c(0,0,0),matrix(c(1,0.3,0.2,0.3,1,0.5,0.2,0.5,1),nrow=3))
w2<-sapply(x[,1], function(t){if(t>median(x[,1])) {return(rbinom(1,1,0.5))}

```

```

    if(t<=median(x[,1])){return(rbinom(1,1,0.3))}})
#ICC=0.7 generate z
r<-c(rep(1,1500),rep(2,500),rep(3,400),rep(4,600))
W<-cbind(x[,3],w2)
z<-list(rbind(cbind(x[1:1500,1]+rnorm(1500,0,sqrt(0.4)),NA,NA,NA),
              cbind(x[1501:1500,1]+rnorm(500,0,sqrt(0.4)),
                    x[1501:1500,1]+rnorm(500,0,sqrt(0.4)),NA,NA),
              cbind(x[2001:2400,1]+rnorm(400,0,sqrt(0.4)),
                    x[2001:2400,1]+rnorm(400,0,sqrt(0.4)),NA),
              cbind(x[2401:3000,1]+rnorm(600,0,sqrt(0.4)),
                    x[2401:3000,1]+rnorm(600,0,sqrt(0.4)),
                    x[2401:3000,1]+rnorm(600,0,sqrt(0.4))),
        rbind(cbind(x[1:1500,2]+rnorm(1500,0,sqrt(0.4)),NA,NA,NA),
              cbind(x[1501:1500,2]+rnorm(500,0,sqrt(0.4)),
                    x[1501:1500,2]+rnorm(500,0,sqrt(0.4)),NA,NA),
              cbind(x[2001:2400,2]+rnorm(400,0,sqrt(0.4)),
                    x[2001:2400,2]+rnorm(400,0,sqrt(0.4)),NA),
              cbind(x[2401:3000,2]+rnorm(600,0,sqrt(0.4)),
                    x[2401:3000,2]+rnorm(600,0,sqrt(0.4)),
                    x[2401:3000,2]+rnorm(600,0,sqrt(0.4))))))
#prevalence about 0.104
p<-exp(-2.65+log(1.5)*rowSums(x[,1:3])+log(1.5)*w2)/
(1+exp(-2.65+log(1.5)*rowSums(x[,1:3])+log(1.5)*w2))
Y<-sapply(p,function(x) rbinom(1,1,x))
fit4<-RCreliability.in(r,z,W=Y)
fit4

```


Index

RCreliability.ex, [2](#)
RCreliability.in, [5](#)