Package ‘icemelt’

July 15, 2019

Type Package

Title Parameter Estimation in Linear Transformation Model with Interval-Censored Data and Covariate Measurement Error

Version 1.0

Date 2019-07-01

Author Soutrik Mandal, Suojin Wang and Samiran Sinha

Maintainer Soutrik Mandal <mandals3@mail.nih.gov>

Description Estimates the parameters of the semiparametric linear transformation model using imputation method, naive method and regression calibration method when time-to-event is interval-censored and a covariate is measured with error. A right censoring indicator must be available. The methods implemented in this package can be found in Mandal, S., Wang, S. and Sinha, S. (2019+). Analysis of Linear Transformation Models with Covariate Measurement Error and Interval Censoring. (accepted, Statistics In Medicine).

License GPL (>= 2)

Imports survival(>= 2.39-5)

NeedsCompilation yes

Repository CRAN

Date/Publication 2019-07-15 11:20:03 UTC

R topics documented:

`icemelt-package` *Parameter Estimation in Linear Transformation Model with Interval-Censored Data and Covariate Measurement Error*

Description

Estimates the parameters of the semiparametric linear transformation model using imputation method, naive method and regression calibration method when time-to-event is interval-censored and a covariate is measured with error. A right censoring indicator must be available. The methods implemented in this package can be found in Mandal, S., Wang, S. and Sinha, S. (2019+). Analysis of Linear Transformation Models with Covariate Measurement Error and Interval Censoring. (accepted, Statistics In Medicine).

Details

The DESCRIPTION file:

Package:	icemelt
Type:	Package
Title:	Parameter Estimation in Linear Transformation Model with Interval-Censored Data and Covariate Measuremen
Version:	1.0
Date:	2019-07-01
Author:	Soutrik Mandal, Suojin Wang and Samiran Sinha
Maintainer:	Soutrik Mandal <mandals3@mail.nih.gov>
Description:	Estimates the parameters of the semiparametric linear transformation model using imputation method, naive me
License:	GPL(>=2)
Imports:	survival(>= 2.39-5)

Index of help topics:

```
icemelt-package         Parameter Estimation in Linear Transformation
                        Model with Interval-Censored Data and Covariate
                        Measurement Error
im                      Parameter Estimation in Linear Transformation
                        Model with Interval-censored Data and Covariate
                        Measurement Error
imw1                    Parameter Estimation in Linear Transformation
                        Model with Interval-censored Data and Covariate
                        Measurement Error
nv                      Parameter Estimation in Linear Transformation
                        Model with Interval-censored Data and Covariate
                        Measurement Error
rc                      Parameter Estimation in LTM using Regression
                        Calibration with Interval-censored Data and
                        Measurement Error
rcw1                    Parameter Estimation in LTM using Regression
```

`Calibration with Interval-censored Data and Measurement Error`

Author(s)

Soutrik Mandal, Suojin Wang and Samiran Sinha

Maintainer: Soutrik Mandal <mandals3@mail.nih.gov>

References

Mandal, S., Wang, S. and Sinha, S. (2019+). Analysis of Linear Transformation Models with Covariate Measurement Error and Interval Censoring. (accepted, Statistics In Medicine) L'ecuyer, P., (1988). Efficient and portable combined random number generators. Communications of the ACM, 31(6), pp.742-751.

`im` *Parameter Estimation in Linear Transformation Model with Interval-censored Data and Covariate Measurement Error*

Description

Estimates the parameters of the semiparametric linear transformation model using imputation method when time-to-event is interval-censored and a covariate is measured with error. Estimated standard errors of the model parameters are also provided. A right censoring indicator must be available along with multiple replications of the surrogate variable that was measured with error. If only one such replication is available, please see imw1() function.

Usage

```
im(datamat, wmat, rfix, gridlen, ntimp, nximp)
```

Arguments

`datamat`	A data matrix with as many rows as there are subjects (n) in the study. The first column contains left points of the intervals, second column contains right points of the intervals, third column is the right censoring indicator and the fourth (and final) column is the error-free covariate.
`wmat`	A matrix of surrogate measurements for the covariate with measurement error.
`rfix`	A value that characterizes the error density in the linear transformation model. The value 0 corresponds to the Cox PH model and 1 corresponds to Proportional Odds model.
`gridlen`	A non-negative value representing the desired grid length used to divide the observed failure-time intervals.
`ntimp`	Number of failure time imputations required.
`nximp`	Number of imputations desired for the error-prone covariate.

Value

Parameter estimates of the semiparamteric linear transformation model and their corresponding standard errors.

Note

Larger sample sizes or imputation numbers will result in longer run times.

Author(s)

Soutrik Mandal, Suojin Wang and Samiran Sinha

References

Mandal, S., Wang, S. and Sinha, S. (2019+). Analysis of Linear Transformation Models with Covariate Measurement Error and Interval Censoring. (accepted, Statistics In Medicine)

Examples

```
## this function is used in generating epsilon from its CDF
  rsep= function(u,r)
        {
          if(r==0)
            return(  log(-log(1-u))  )
          else
            return(  log((exp(-r*log(1-u))-1)/r)  )
        }


  n= 30 #200     # sample size; small number used for quick demonstration only
  rfix= 1.0

 m= 3 #10      # imputed datasets for failure time; small number used for quick demonstration only

  nrep= 2        # number of repeated measurement of error prone covariate

  gridlen= 0.1

  result= NULL
  ah=1

  set.seed(ah)

#  z1= rnorm(n, mean= 0, sd=1)
  z1= (rgamma(n,shape=2,scale=2)-4)/sqrt(8)
  z2= rbinom(n,1,0.5)

  ugen= runif(n)
  ep= rsep(ugen,rfix)

  truebeta= c(-1,1)
```

```
  logt= -truebeta[1]*z1 -truebeta[2]*z2 + ep + 3
  ttime= exp(logt)

  cen= runif(n,0,0.0001)

## creating tau matrix to locate the actual times in each rows and form the corresponding intervals
  len= 0.15
  taumat= matrix(0,n,10)
  taumat[,10]= 9000000000             # if you're changing this, change rcpos below
  taumat[,2]= cen
  for(i1 in 3:9)
    taumat[,i1]= taumat[,2]+(i1-1)*len

## now forming the intervals
  right1= rep(0,n)
  left= rep(0,n)
  for(i2 in 1:n)
  {
    lenleft= length(which(taumat[i2,2:9]<ttime[i2]))
    leftpot= rep(0,8)
    leftpot[1:lenleft]= 1
    missvec1= c(rbinom(4,1,0.7),rbinom(4,1,0.5))
    left[i2]= max(leftpot*missvec1*taumat[i2,2:9])

    if(left[i2]==0)
      left[i2]= taumat[i2,2]

    lenright= length(which(taumat[i2,2:10]>ttime[i2]))
    rightpot= rep(0,9)
    rightpot[(9-lenright+1):9]= 1
    missvec2= c(rbinom(4,1,0.7),rbinom(4,1,0.5),1)
    right1temp= rightpot*missvec2*taumat[i2,2:10]
    right1[i2]= min(right1temp[right1temp!=0])
  }

  rcpos= which(right1==9000000000)
  lrcpos= length(rcpos)
  notrcpos= (1:n)[-rcpos]

  delta_temp= rep(1,n) # del=1 are uncensored observations
  delta_temp[rcpos]= 0
 k= sum(delta_temp)   # this is just the number of data points that are not right censored

## measurement error generation
  umat= matrix((rgamma(n*nrep,shape=2,scale=2)-4)*0.5/sqrt(8),n,nrep)

  wmat= z1+umat


  datamat= cbind(left,right1,delta_temp,z2)
  ntimp= m
 nximp= 5 #20   # number of x imputations; small number used for quick demonstration only
```

```
library(icemelt)
out_im= im(datamat, wmat, rfix, gridlen, ntimp, nximp)
out_im
```

---

imw1 *Parameter Estimation in Linear Transformation Model with Interval-censored Data and Covariate Measurement Error*

Description

This function should be used when only a single replication of the error-prone covariate is available. If multiple replications are available please see im() function. This function estimates the parameters of the semiparametric linear transformation model using imputation method when time-to-event is interval-censored and a covariate is measured with error. Estimated standard errors of the model parameters are also provided. A right censoring indicator must be available along with the measurement error variance.

Usage

```
imw1(datamat, wmat, rfix, gridlen, ntimp, nximp, sigma2u)
```

Arguments

`datamat`	A data matrix with as many rows as there are subjects (n) in the study. The first column contains left points of the intervals, second column contains right points of the intervals, third column is the right censoring indicator and the fourth (and final) column is the error-free covariate.
`wmat`	A matrix with only one column of surrogate measurements for the covariate with measurement error.
`rfix`	A value that characterizes the error density in the linear transformation model. The value 0 corresponds to the Cox PH model and 1 corresponds to Proportional Odds model.
`gridlen`	A non-negative value representing the desired grid length used to divide the observed failure-time intervals.
`ntimp`	Number of failure time imputations required.
`nximp`	Number of imputations desired for the error-prone covariate.
`sigma2u`	A known value for the measurement error variance.

Value

Parameter estimates of the semiparamteric linear transformation model and their corresponding standard errors.

Note

Larger sample sizes or imputation numbers will result in longer run times.

Author(s)

Soutrik Mandal, Suojin Wang and Samiran Sinha

References

Mandal, S., Wang, S. and Sinha, S. (2019+). Analysis of Linear Transformation Models with Covariate Measurement Error and Interval Censoring. (accepted, Statistics In Medicine)

Examples

```
## this function is used in generating epsilon from its CDF
  rsep= function(u,r)
        {
          if(r==0)
            return(  log(-log(1-u))  )
          else
            return(  log((exp(-r*log(1-u))-1)/r)  )
        }


  n= 30 #200     # sample size; small number used for quick demonstration only
  rfix= 1.0


  m= 3 #10      # imputed datasets for failure time; small number used for quick demonstration only

  nrep= 1        # number of repeated measurement of error prone covariate

  gridlen= 0.1
  sigma2u= 0.5

  result= NULL
  ah=1

  set.seed(ah)

#  z1= rnorm(n, mean= 0, sd=1)
  z1= (rgamma(n,shape=2,scale=2)-4)/sqrt(8)
  z2= rbinom(n,1,0.5)

  ugen= runif(n)
  ep= rsep(ugen,rfix)

  truebeta= c(-1,1)

  logt= -truebeta[1]*z1 -truebeta[2]*z2 + ep + 3
  ttime= exp(logt)

  cen= runif(n,0,0.0001)

## creating tau matrix to locate the actual times in each rows and form the corresponding intervals
  len= 0.15
```

```
  taumat= matrix(0,n,10)
  taumat[,10]= 9000000000             # if you're changing this, change rcpos below
  taumat[,2]= cen
  for(i1 in 3:9)
    taumat[,i1]= taumat[,2]+(i1-1)*len

## now forming the intervals
  right1= rep(0,n)
  left= rep(0,n)
  for(i2 in 1:n)
  {
    lenleft= length(which(taumat[i2,2:9]<ttime[i2]))
    leftpot= rep(0,8)
    leftpot[1:lenleft]= 1
    missvec1= c(rbinom(4,1,0.7),rbinom(4,1,0.5))
    left[i2]= max(leftpot*missvec1*taumat[i2,2:9])

    if(left[i2]==0)
      left[i2]= taumat[i2,2]

    lenright= length(which(taumat[i2,2:10]>ttime[i2]))
    rightpot= rep(0,9)
    rightpot[(9-lenright+1):9]= 1
    missvec2= c(rbinom(4,1,0.7),rbinom(4,1,0.5),1)
    right1temp= rightpot*missvec2*taumat[i2,2:10]
    right1[i2]= min(right1temp[right1temp!=0])
  }

  rcpos= which(right1==9000000000)
  lrcpos= length(rcpos)
  notrcpos= (1:n)[-rcpos]

  delta_temp= rep(1,n) # del=1 are uncensored observations
  delta_temp[rcpos]= 0
  k= sum(delta_temp)   # number of data points that are not right censored

## measurement error generation
  umat= matrix((rgamma(n*nrep,shape=2,scale=2)-4)*0.5/sqrt(8),n,nrep)

  wmat= z1+umat


  datamat= cbind(left,right1,delta_temp,z2)
  ntimp= m
  nximp= 5 #20   # number of x imputations; small number used for quick demonstration only

  library(icemelt)
  out_imw1= imw1(datamat, wmat, rfix, gridlen, ntimp, nximp, sigma2u)
  out_imw1
```

nv — *Parameter Estimation in Linear Transformation Model with Interval-censored Data and Covariate Measurement Error*

Description

Estimates the parameters of the semiparametric linear transformation model using naive method when time-to-event is interval-censored and a covariate is measured with error. Estimated standard errors of the model parameters are also provided. A right censoring indicator must be available.

Usage

```
nv(datamat, wmat, rfix, gridlen, ntimp)
```

Arguments

`datamat`	A data matrix with as many rows as there are subjects (n) in the study. The first column contains left points of the intervals, second column contains right points of the intervals, third column is the right censoring indicator and the fourth (and final) column is the error-free covariate.
`wmat`	A matrix of surrogate measurements for the covariate with measurement error.
`rfix`	A value that characterizes the error density in the linear transformation model. The value 0 corresponds to the Cox PH model and 1 corresponds to Proportional Odds model.
`gridlen`	A non-negative value representing the desired grid length used to divide the observed failure-time intervals.
`ntimp`	Number of failure time imputations required.

Value

Parameter estimates of the semiparamteric linear transformation model and their corresponding standard errors.

Note

Larger sample sizes or imputation numbers will result in longer run times.

Author(s)

Soutrik Mandal, Suojin Wang and Samiran Sinha

References

Mandal, S., Wang, S. and Sinha, S. (2019+). Analysis of Linear Transformation Models with Covariate Measurement Error and Interval Censoring. (accepted, Statistics In Medicine)

Examples

```
## this function is used in generating epsilon from its CDF
  rsep= function(u,r)
        {
          if(r==0)
            return(  log(-log(1-u))  )
          else
            return(  log((exp(-r*log(1-u))-1)/r)  )
        }


  n= 30 #200     # sample size; small number used for quick demonstration only
  rfix= 1.0


  m= 3 #10      # imputed datasets for failure time; small number used for quick demonstration only

  nrep= 2        # number of repeated measurement of error prone covariate

  gridlen= 0.1

  result= NULL
  ah=1

  set.seed(ah)

#  z1= rnorm(n, mean= 0, sd=1)
  z1= (rgamma(n,shape=2,scale=2)-4)/sqrt(8)
  z2= rbinom(n,1,0.5)

  ugen= runif(n)
  ep= rsep(ugen,rfix)

  truebeta= c(-1,1)

  logt= -truebeta[1]*z1 -truebeta[2]*z2 + ep + 3
  ttime= exp(logt)

  cen= runif(n,0,0.0001)

## creating tau matrix to locate the actual times in each rows and form the corresponding intervals
  len= 0.15
  taumat= matrix(0,n,10)
  taumat[,10]= 9000000000            # if you're changing this, change rcpos below
  taumat[,2]= cen
  for(i1 in 3:9)
    taumat[,i1]= taumat[,2]+(i1-1)*len

## now forming the intervals
  right1= rep(0,n)
  left= rep(0,n)
  for(i2 in 1:n)
```

```
    {
      lenleft= length(which(taumat[i2,2:9]<ttime[i2]))
      leftpot= rep(0,8)
      leftpot[1:lenleft]= 1
      missvec1= c(rbinom(4,1,0.7),rbinom(4,1,0.5))
      left[i2]= max(leftpot*missvec1*taumat[i2,2:9])

      if(left[i2]==0)
        left[i2]= taumat[i2,2]

      lenright= length(which(taumat[i2,2:10]>ttime[i2]))
      rightpot= rep(0,9)
      rightpot[(9-lenright+1):9]= 1
      missvec2= c(rbinom(4,1,0.7),rbinom(4,1,0.5),1)
      right1temp= rightpot*missvec2*taumat[i2,2:10]
      right1[i2]= min(right1temp[right1temp!=0])
    }

    rcpos= which(right1==9000000000)
    lrcpos= length(rcpos)
    notrcpos= (1:n)[-rcpos]

    delta_temp= rep(1,n) # del=1 are uncensored observations
    delta_temp[rcpos]= 0
   k= sum(delta_temp)   # this is just the number of data points that are not right censored

  ## measurement error generation
    umat= matrix((rgamma(n*nrep,shape=2,scale=2)-4)*0.5/sqrt(8),n,nrep)

    wmat= z1+umat


    datamat= cbind(left,right1,delta_temp,z2)
    ntimp= m

    library(icemelt)
    out_nv= nv(datamat, wmat, rfix, gridlen, ntimp)
    out_nv
```

`rc` *Parameter Estimation in LTM using Regression Calibration with Interval-censored Data and Measurement Error*

Description

Estimates the parameters of the semiparametric linear transformation model using regression calibration method when time-to-event is interval-censored and a covariate is measured with error. Estimated standard errors of the model parameters are also provided. A right censoring indicator must be available along with multiple replications of the surrogate variable that was measured with error. If only one such replication is available, please see rcw1() function.

Usage

```
rc(datamat, wmat, rfix, gridlen, ntimp)
```

Arguments

- `datamat` A data matrix with as many rows as there are subjects (n) in the study. The first column contains left points of the intervals, second column contains right points of the intervals, third column is the right censoring indicator and the fourth (and final) column is the error-free covariate.
- `wmat` A matrix of surrogate measurements for the covariate with measurement error.
- `rfix` A value that characterizes the error density in the linear transformation model. The value 0 corresponds to the Cox PH model and 1 corresponds to Proportional Odds model.
- `gridlen` A non-negative value representing the desired grid length used to divide the observed failure-time intervals.
- `ntimp` Number of failure time imputations required.

Value

Parameter estimates of the semiparamteric linear transformation model and their corresponding standard errors.

Note

Larger sample sizes or imputation numbers will result in longer run times.

Author(s)

Soutrik Mandal, Suojin Wang and Samiran Sinha

References

Mandal, S., Wang, S. and Sinha, S. (2019+). Analysis of Linear Transformation Models with Covariate Measurement Error and Interval Censoring. (accepted, Statistics In Medicine)

Examples

```
## this function is used in generating epsilon from its CDF
  rsep= function(u,r)
        {
          if(r==0)
            return(  log(-log(1-u))  )
          else
            return(  log((exp(-r*log(1-u))-1)/r)  )
        }


  n= 30 #200     # sample size; small number used for quick demonstration only
  rfix= 1.0
```

```
 m= 3 #10     # imputed datasets for failure time; small number used for quick demonstration only

  nrep= 2       # number of repeated measurement of error prone covariate

  gridlen= 0.1

  result= NULL
  ah=1

  set.seed(ah)

#  z1= rnorm(n, mean= 0, sd=1)
  z1= (rgamma(n,shape=2,scale=2)-4)/sqrt(8)
  z2= rbinom(n,1,0.5)

  ugen= runif(n)
  ep= rsep(ugen,rfix)

  truebeta= c(-1,1)

  logt= -truebeta[1]*z1 -truebeta[2]*z2 + ep + 3
  ttime= exp(logt)

  cen= runif(n,0,0.0001)

## creating tau matrix to locate the actual times in each rows and form the corresponding intervals
  len= 0.15
  taumat= matrix(0,n,10)
  taumat[,10]= 9000000000            # if you're changing this, change rcpos below
  taumat[,2]= cen
  for(i1 in 3:9)
    taumat[,i1]= taumat[,2]+(i1-1)*len

## now forming the intervals
  right1= rep(0,n)
  left= rep(0,n)
  for(i2 in 1:n)
  {
    lenleft= length(which(taumat[i2,2:9]<ttime[i2]))
    leftpot= rep(0,8)
    leftpot[1:lenleft]= 1
    missvec1= c(rbinom(4,1,0.7),rbinom(4,1,0.5))
    left[i2]= max(leftpot*missvec1*taumat[i2,2:9])

    if(left[i2]==0)
      left[i2]= taumat[i2,2]

    lenright= length(which(taumat[i2,2:10]>ttime[i2]))
    rightpot= rep(0,9)
    rightpot[(9-lenright+1):9]= 1
    missvec2= c(rbinom(4,1,0.7),rbinom(4,1,0.5),1)
```

```
    right1temp= rightpot*missvec2*taumat[i2,2:10]
    right1[i2]= min(right1temp[right1temp!=0])
  }

  rcpos= which(right1==9000000000)
  lrcpos= length(rcpos)
  notrcpos= (1:n)[-rcpos]

  delta_temp= rep(1,n) # del=1 are uncensored observations
  delta_temp[rcpos]= 0
 k= sum(delta_temp)   # this is just the number of data points that are not right censored

## measurement error generation
  umat= matrix((rgamma(n*nrep,shape=2,scale=2)-4)*0.5/sqrt(8),n,nrep)

  wmat= z1+umat

  datamat= cbind(left,right1,delta_temp,z2)
  ntimp= m

  library(icemelt)
  out_rc= rc(datamat, wmat, rfix, gridlen, ntimp)
  out_rc
```

`rcw1` *Parameter Estimation in LTM using Regression Calibration with Interval-censored Data and Measurement Error*

Description

This function should be used when only a single replication of the error-prone covariate is available. If multiple replications are available please see rc() function. This function estimates the parameters of the semiparametric linear transformation model using regression calibration method when time-to-event is interval-censored and a covariate is measured with error. Estimated standard errors of the model parameters are also provided. A right censoring indicator must be available.

Usage

```
rcw1(datamat, wmat, rfix, gridlen, ntimp, sigma2u)
```

Arguments

`datamat`	A data matrix with as many rows as there are subjects (n) in the study. The first column contains left points of the intervals, second column contains right points of the intervals, third column is the right censoring indicator and the fourth (and final) column is the error-free covariate.
`wmat`	A matrix of surrogate measurements for the covariate with measurement error.

rfix	A value that characterizes the error density in the linear transformation model. The value 0 corresponds to the Cox PH model and 1 corresponds to Proportional Odds model.
gridlen	A non-negative value representing the desired grid length used to divide the observed failure-time intervals.
ntimp	Number of failure time imputations required.
sigma2u	A known value for the measurement error variance.

Value

Parameter estimates of the semiparamteric linear transformation model and their corresponding standard errors.

Note

Larger sample sizes or imputation numbers will result in longer run times.

Author(s)

Soutrik Mandal, Suojin Wang and Samiran Sinha

References

Mandal, S., Wang, S. and Sinha, S. (2019+). Analysis of Linear Transformation Models with Covariate Measurement Error and Interval Censoring. (accepted, Statistics In Medicine)

Examples

```
## this function is used in generating epsilon from its CDF
  rsep= function(u,r)
        {
          if(r==0)
            return(  log(-log(1-u))  )
          else
            return(  log((exp(-r*log(1-u))-1)/r)  )
        }


  n= 30 #200     # sample size; small number used for quick demonstration only
  rfix= 1.0
  sigma2u= 0.5

 m= 3 #10      # imputed datasets for failure time; small number used for quick demonstration only

  nrep= 1        # number of repeated measurement of error prone covariate

  gridlen= 0.1

  result= NULL
  ah=1
```

```
  set.seed(ah)

#  z1= rnorm(n, mean= 0, sd=1)
  z1= (rgamma(n,shape=2,scale=2)-4)/sqrt(8)
  z2= rbinom(n,1,0.5)

  ugen= runif(n)
  ep= rsep(ugen,rfix)

  truebeta= c(-1,1)

  logt= -truebeta[1]*z1 -truebeta[2]*z2 + ep + 3
  ttime= exp(logt)

  cen= runif(n,0,0.0001)

## creating tau matrix to locate the actual times in each rows and form the corresponding intervals
  len= 0.15
  taumat= matrix(0,n,10)
  taumat[,10]= 9000000000             # if you're changing this, change rcpos below
  taumat[,2]= cen
  for(i1 in 3:9)
    taumat[,i1]= taumat[,2]+(i1-1)*len

## now forming the intervals
  right1= rep(0,n)
  left= rep(0,n)
  for(i2 in 1:n)
  {
    lenleft= length(which(taumat[i2,2:9]<ttime[i2]))
    leftpot= rep(0,8)
    leftpot[1:lenleft]= 1
    missvec1= c(rbinom(4,1,0.7),rbinom(4,1,0.5))
    left[i2]= max(leftpot*missvec1*taumat[i2,2:9])

    if(left[i2]==0)
      left[i2]= taumat[i2,2]

    lenright= length(which(taumat[i2,2:10]>ttime[i2]))
    rightpot= rep(0,9)
    rightpot[(9-lenright+1):9]= 1
    missvec2= c(rbinom(4,1,0.7),rbinom(4,1,0.5),1)
    right1temp= rightpot*missvec2*taumat[i2,2:10]
    right1[i2]= min(right1temp[right1temp!=0])
  }

  rcpos= which(right1==9000000000)
  lrcpos= length(rcpos)
  notrcpos= (1:n)[-rcpos]

  delta_temp= rep(1,n) # del=1 are uncensored observations
  delta_temp[rcpos]= 0
 k= sum(delta_temp)   # this is just the number of data points that are not right censored
```

```
## measurement error generation
  umat= matrix((rgamma(n*nrep,shape=2,scale=2)-4)*0.5/sqrt(8),n,nrep)

  wmat= z1+umat


  datamat= cbind(left,right1,delta_temp,z2)
  ntimp= m

  library(icemelt)
  out_rcw1= rcw1(datamat, wmat, rfix, gridlen, ntimp, sigma2u)
  out_rcw1
```

Index