

# Package: pseudo (via r-universe)

October 22, 2024

**Title** Computes Pseudo-Observations for Modeling

**Version** 1.4.3

**Description** Various functions for computing pseudo-observations for censored data regression. Computes pseudo-observations for modeling: competing risks based on the cumulative incidence function, survival function based on the restricted mean, survival function based on the Kaplan-Meier estimator see Klein et al. (2008) <[doi:10.1016/j.cmpb.2007.11.017](https://doi.org/10.1016/j.cmpb.2007.11.017)>.

**Depends** R (>= 3.3.3), KMsurv, geepack

**License** GPL-2

**Encoding** UTF-8

**Author** Maja Pohar Perme [aut], Mette Gerster [aut], Kevin Rodrigues [cre]

**Date/Publication** 2017-07-30 23:00:00 UTC

**NeedsCompilation** no

**Maintainer** Kevin Rodrigues <[kevin.asr@outlook.com](mailto:kevin.asr@outlook.com)>

**Repository** <https://kevin444.r-universe.dev>

**RemoteUrl** <https://github.com/cran/pseudo>

**RemoteRef** HEAD

**RemoteSha** 0f0d7575246694fd5bae39b6e26ac60c43d37cfa

## Contents

pseudoci . . . . .	2
pseudomean . . . . .	3
pseudosurv . . . . .	5
pseudoyl . . . . .	6

<b>Index</b>	<b>8</b>
--------------	----------

---

pseudoci

*Pseudo-observations for the cumulative incidence function*

---

### Description

Computes pseudo-observations for modeling competing risks based on the cumulative incidence function.

### Usage

```
pseudoci(time, event, tmax)
```

### Arguments

time	the follow up time.
event	the cause indicator, use 0 as censoring code and integers to name the other causes.
tmax	a vector of time points at which the pseudo-observations are to be computed. If missing, the pseudo-observations are reported at each event time.

### Details

The function calculates the pseudo-observations for the cumulative incidence function for each individual and each risk at all the required time points. The pseudo-observations can be used for fitting a regression model with a generalized estimating equation. No missing values in either time or event vector are allowed.

Please note that the output of the function has changed and the usage is thus no longer the same as in the reference paper - the new usage is described in the example below. Similar (faster) version of the function is available in the R-package `prodlim` (`jackknife`).

### Value

A list containing the following objects:

time	The ordered time points at which the pseudo-observations are evaluated.
cause	The ordered codes for different causes.
pseudo	A list of matrices - a matrix for each of the causes, ordered by codes. Each row of a matrix belongs to one individual (ordered as in the original data set), each column presents a time point (ordered in time).

### References

Klein J.P., Gerster M., Andersen P.K., Tarima S., POHAR PERME, M.: "SAS and R Functions to Compute Pseudo-values for Censored Data Regression." *Comput. methods programs biomed.*, 2008, 89 (3): 289-300

**See Also**

[pseudoyl](#), [pseudomean](#), [pseudosurv](#)

**Examples**

```
library(KMsurv)
data(bmt)

#calculate the pseudo-observations
cutoffs <- c(50,105,170,280,530)
bmt$icr <- bmt$d1 + bmt$d3
pseudo <- pseudoci(time=bmt$t2,event=bmt$icr,tmax=cutoffs)

#rearrange the data into a long data set, use only pseudo-observations for relapse (icr=2)
b <- NULL
for(it in 1:length(pseudo$time)){
  b <- rbind(b,cbind(bmt,pseudo = pseudo$pseudo[[2]][,it],
    tpseudo = pseudo$time[it],id=1:nrow(bmt)))
}
b <- b[order(b$id),]

# fit the model
library(geepack)
fit <- geese(pseudo ~ as.factor(tpseudo) + as.factor(group) + as.factor(z8) +
  z1 - 1, data = b, id=id, jack = TRUE, scale.fix=TRUE, family=gaussian,
  mean.link = "cloglog", corstr="independence")

#The results using the AJ variance estimate
cbind(mean = round(fit$beta,4), SD = round(sqrt(diag(fit$vbeta.ajs)),4),
  Z = round(fit$beta/sqrt(diag(fit$vbeta.ajs)),4),
  PVal = round(2-2*pnorm(abs(fit$beta/sqrt(diag(fit$vbeta.ajs))))),4))
```

---

pseudomean

*Pseudo-observations for the restricted mean*

---

**Description**

Computes pseudo-observations for modeling survival function based on the restricted mean.

**Usage**

```
pseudomean(time,event, tmax)
```

**Arguments**

time            the follow up time.  
 event          the status indicator: 0=alive, 1=dead.

`tmax` the maximum cut-off point for the restricted mean. If missing or larger than the maximum follow up time, it is replaced by the maximum follow up time.

### Details

The function calculates the pseudo-observations for the restricted mean survival for each individual at prespecified time-points. The pseudo-observations can be used for fitting a regression model with a generalized estimating equation. No missing values in either `time` or `event` vector are allowed.

Please note that the output of the function has changed and the usage is thus no longer the same as in the reference paper - the new usage is described in the example below.

### Value

A vector of pseudo-observations for each individual.

### References

Klein J.P., Gerster M., Andersen P.K., Tarima S., POHAR PERME, M.: "SAS and R Functions to Compute Pseudo-values for Censored Data Regression." *Comput. methods programs biomed.*, 2008, 89 (3): 289-300

### See Also

[pseudosurv](#), [pseudoci](#)

### Examples

```
library(KMsurv)
data(bmt)

#compute the pseudo-observations:
pseudo = pseudomean(time=bmt$t2, event=bmt$d3,tmax=2000)

#arrange the data
a <- cbind(bmt,pseudo = pseudo,id=1:nrow(bmt))

#fit a regression model for the mean time

library(geepack)
summary(fit <- geese(pseudo ~ z1 + as.factor(z8) + as.factor(group),
  data = a, id=id, jack = TRUE, family=gaussian,
  constr="independence", scale.fix=FALSE))

#rearrange the output
round(cbind(mean = fit$beta,SD = sqrt(diag(fit$vbeta.ajs)),
  Z = fit$beta/sqrt(diag(fit$vbeta.ajs)), PVal =
  2-2*pnorm(abs(fit$beta/sqrt(diag(fit$vbeta.ajs))))),4)
```

---

pseudosurv

*Pseudo-observations for the Kaplan-Meier estimate*

---

## Description

Computes pseudo-observations for modeling survival function based on the Kaplan-Meier estimator.

## Usage

```
pseudosurv(time,event, tmax)
```

## Arguments

time	the follow up time.
event	the status indicator: 0=alive, 1=dead.
tmax	a vector of time points at which the pseudo-observations are to be computed. If missing, the pseudo-observations are reported at each event time.

## Details

The function calculates the pseudo-observations for the value of the survival function at prespecified time-points for each individual. The pseudo-observations can be used for fitting a regression model with a generalized estimating equation. No missing values in either `time` or `event` vector are allowed.

Please note that the output of the function has changed and the usage is thus no longer the same as in the reference paper - the new usage is described in the example below. Similar (faster) version of the function is available in the R-package `prodlim` (`jackknife`).

## Value

A list containing the following objects:

time	The ordered time points at which the pseudo-observations are evaluated.
pseudo	A matrix. Each row belongs to one individual (ordered as in the original data set), each column presents a time point (ordered in time).

## References

Klein J.P., Gerster M., Andersen P.K., Tarima S., POHAR PERME, M.: "SAS and R Functions to Compute Pseudo-values for Censored Data Regression." *Comput. methods programs biomed.*, 2008, 89 (3): 289-300

## See Also

[pseudomean](#), [pseudoci](#), [pseudoyl](#)

**Examples**

```

library(KMsurv)
data(bmt)

#calculate the pseudo-observations
cutoffs <- c(50,105,170,280,530)
pseudo <- pseudosurv(time=bmt$t2,event=bmt$d3,tmax=cutoffs)

#rearrange the data into a long data set
b <- NULL
for(it in 1:length(pseudo$time)){
  b <- rbind(b,cbind(bmt,pseudo = pseudo$pseudo[,it],
    tpseudo = pseudo$time[it],id=1:nrow(bmt)))
}
b <- b[order(b$id),]

#fit a Cox model using GEE
library(geepack)
summary(fit <- geese(pseudo~as.factor(tpseudo)+as.factor(group)+
  as.factor(z8)+z1,data=b,scale.fix=TRUE,family=gaussian,
  jack=TRUE, mean.link="cloglog",corstr="independence"))

#The results using the AJ variance estimate
round(cbind(mean = fit$beta,SD = sqrt(diag(fit$vbeta.ajs)),
  Z = fit$beta/sqrt(diag(fit$vbeta.ajs)), PVal =
  2-2*pnorm(abs(fit$beta/sqrt(diag(fit$vbeta.ajs))))),4)

```

pseudoyl

*Pseudo-observations for the expected number of years lost***Description**

Computes pseudo-observations for modeling using the number of years lost.

**Usage**

```
pseudoyl(time,event, tmax)
```

**Arguments**

time	the follow up time.
event	the cause indicator, use 0 as censoring code and integers to name the other causes.
tmax	the maximum cut-off point time = the upper limit of the integral of the cumulative incidence function. If missing or larger than the maximum follow up time, it is replaced by the maximum follow up time.

## Details

The function calculates the pseudo-observations for the expected number of years lost for each individual. The pseudo-observations can be used for fitting a regression model with a generalized estimating equation. No missing values in either time or event vector are allowed.

## Value

A list containing the following objects:

cause	The ordered codes for different causes.
pseudo	A list of vectors- a vector for each of the causes, ordered by codes. Each value of a vector belongs to one individual (ordered as in the original data set).

## References

Andersen P.K.: "A note on the decomposition of number of life years lost according to causes of death." *Research report, Department of Biostatistics, University of Copenhagen, 2012 (2)*

## See Also

[pseudoci](#), [pseudomean](#), [pseudosurv](#)

## Examples

```
library(KMsurv)
data(bmt)
bmt$icr <- bmt$d1 + bmt$d3

#compute the pseudo-observations:
pseudo = pseudoyl(time=bmt$t2, event=bmt$icr, tmax=2000)

#arrange the data - use pseudo observations for cause 2
a <- cbind(bmt, pseudo = pseudo$pseudo[[2]], id=1:nrow(bmt))

#fit a regression model for cause 2

library(geepack)
summary(fit <- geese(pseudo ~ z1 + as.factor(z8) + as.factor(group),
  data = a, id=id, jack = TRUE, family=gaussian,
  constr="independence", scale.fix=FALSE))

#rearrange the output
round(cbind(mean = fit$beta, SD = sqrt(diag(fit$vbeta.ajs)),
  Z = fit$beta/sqrt(diag(fit$vbeta.ajs)), PVal =
  2-2*pnorm(abs(fit$beta/sqrt(diag(fit$vbeta.ajs))))), 4)
```

# Index

## \* survival

pseudoci, 2

pseudomean, 3

pseudosurv, 5

pseudoyl, 6

pseudoci, 2, 4, 5, 7

pseudomean, 3, 3, 5, 7

pseudosurv, 3, 4, 5, 7

pseudoyl, 3, 5, 6