Package ‘weightQuant’

May 8, 2026

Type Package

Title Weights for Incomplete Longitudinal Data and Quantile Regression

Version 1.0.1

Date 2022-01-05

Author Viviane Philipps

Maintainer Viviane Philipps <Viviane.Philipps@u-bordeaux.fr>

Description Estimation of observation-specific weights for incomplete longitudinal data and bootstrap procedure for weighted quantile regressions. See Jacqmin-Gadda, Rouanet, Mba, Philipps, Dartigues (2020) for details <doi:10.1177/0962280220909986>.

License GPL (>= 2.0)

Depends quantreg, foreach, doParallel, stringr

BugReports https://github.com/VivianePhilipps/weightQuant/issues

LazyLoad yes

LazyData true

NeedsCompilation no

Repository CRAN

Date/Publication 2022-01-05 12:00:02 UTC

Contents

`weightQuant-package` *Weights for incomplete longitudinal data and quantile regression*

Description

Functions for the estimation of observation-specific weights for incomplete longitudinal data. A bootstrap method is also provided to obtain standard erros of weighted quantile regressions.

Details

Package:	weightQuant
Type:	Package
Version:	1.0.1
Date:	2022-01-05
License:	GPL (>= 2.0)

Index:

```
bootwrq                 Bootstrap procedure for weighted quantile
                        regressions
simdata                 Simulated dataset
summary.bootwrq         Summary of a quantile regression model
test.bootwrq            Test of covariate effects between different
                        quantiles
weightQuant-package     Weights for incomplete longitudinal data and
                        quantile regression
weightsIMD              Estimation of observation-specific weights with
                        intermittent missing data
weightsMMD              Estimation of observation-specific weights with
                        monotone missing data
```

Author(s)

Viviane Philipps

References

Jacqmin-Gadda H, Rouanet A, Mba RD, Philipps V, Dartigues J-F. Quantile regression for incomplete longitudinal data with selection by death. Statistical Methods in Medical Research. 2020;29(9):2697-2716. doi:10.1177/0962280220909986

bootwrq *Bootstrap procedure for weighted quantile regressions*

Description

A subject-level bootstrap method for weighted quantile regressions is implemented in this function. Quantile regressions are estimated in a generalized estimating equation framework with independent working covariance matrix. Weights are estimated using `weightsIMD` or `weightsMMD` functions.

Usage

```
bootwrq(B, form, tau, data, Y, X1 = NULL, X2 = NULL, subject,
death, time, interval.death = NULL, impute = NULL, weight = NULL,
wcompute = 2, seed = NULL, intermittent, file = NULL,
nproc = 1, MPI = FALSE)
```

Arguments

`B`	integer, number of bootstrap samples
`form`	formula indicating the quantile regression model to be estimated
`tau`	numeric vector indicating the quantiles to be estimated
`data`	data frame containing the data
`Y`	character indicating the name of the response outcome
`X1`	optional character vector passed to the weight functions
`X2`	optional character vector passed to the weight functions
`subject`	character indicating the name of the subject identifier
`death`	optional character passed to the weight functions
`time`	optional character passed to the weight functions
`interval.death`	optional numeric vector passed to the weight function weightsMMD
`impute`	optional numeric vector passed to the weight function weightsIMD
`weight`	character indicating the name of the weight variable in data
`wcompute`	integer indicating if weights should be estimated in each bootstrap sample. If wcompute=0, weights are supposed to be known. If wcompute=1, weights are re-estimated in each bootstrp sample. If wcompute=2, both results are returned.
`seed`	optional integer vector of length B indicating the seeds.
`intermittent`	logical indicating if data contains intermittent missing data. If intermittent=TRUE, the weights are estimated using weightsIMD function, if intermittent=FALSE, the weights are estimated using weightsMMD function.
`file`	optional character indicating the name of the results file. If file=NULL, no results file is created.
`nproc`	number of processors to be used for parallel computing. Default to 1, sequential computation.
`MPI`	logical indicating if MPI parallelization should be used. Default to FALSE.

Value

a matrix with B columns containing the results on each bootstrap sample.

Author(s)

Viviane Philipps, Robert Darlin Mba

See Also

summary.bootwrq, test.bootwrq

Examples

```
## Not run:
## computation of the weights with intermittent missing data
w_simdata <- weightsIMD(data=simdata,Y="Y",X1="X",X2=NULL,subject="id",
death="death",time="time",impute=20,name="w_imd")$data

## estimation of the weighted quantile regressions
## for the first quartile and the median
m_simdata <- rq(Y~time*X,data=w_simdata,weights=w_imd,tau=c(0.25,0.5))

## estimation of the standard erros using the bootstrap procedure
boot_simdata <- bootwrq(B=1000, form=Y~time*X, tau=c(0.25,0.5),
data=w_simdata, Y="Y",X1="X",X2=NULL,subject="id",
death="death",time="time",impute=20,wcompute=0,intermittent=TRUE)

## the summary of the results
summary(boot_simdata,m_simdata)

## comparison of the covariate effects
## between the first quartile and the median
test.bootwrq(boot_simdata,m_simdata)

## End(Not run)
```

---

simdata *Simulated dataset*

Description

The data were simulated from a linear mixed model. Repeated data of the longitudinal outcome were simulated for 500 subjects. Death time was simulated depending on the (observed and unobserved) longitudinal outcome and on the binary covariate. Missing data before death were simulated using a logistic regression model including the binary covariate, the outcome at the previous visit and the observation status at the previous visit.

Usage

```
simdata
```

Format

A data frame with 2123 observations over 500 different subjects and 7 variables.

`id` subject identification number

`X` binary covariate

`death` death time (missing for subjects alive)

`time` measurement time

`age` age at measurement time

`Y` longitudinal outcome

`Ytrunc` longitudinal outcome truncated at the first missing value

---

`summary.bootwrq` *Summary of a quantile regression model*

Description

The function provides a summary of quantile regression estimation. Standard erros and p values are obtained from a bootstrap procedure.

Usage

```
## S3 method for class 'bootwrq'
summary(object, ...)
```

Arguments

`object`	results from bootstrap estimations obtained with bootwrq function
`...`	additional arguments. If a quantile regression model estimated with rq function from quantreg package is specified, the function uses these estimated coefficients as results. Otherwise, the coefficients are obtained as the mean over the B estimated coefficients from the bootstrap results.

Value

A list containing :

`results0`	a matrix with 3 columns containing the results (coefficients, standard erros and p-values) without computing the weights in each bootstrap sample. Or NULL if the bootstrap results are obtained with wcompute=1.
`results1`	a matrix with 3 columns containing the results (coefficients, standard erros and p-values) with re-estimated weights on each bootstrap sample. Or NULL if the bootstrap results are obtained with wcompute=0.

Author(s)

Viviane Philipps

`test.bootwrq` *Test of covariate effects between different quantiles*

Description

This function provides a test for the covariate effects estimated for different quantiles.

Usage

```
test.bootwrq(x, m)
```

Arguments

`x`	results from bootstrap estimations obtained with bootwrq function
`m`	a quantile regression model estimated with rq function from quantreg package. At least 2 quantiles should be specified in rq function.

Details

For 2 quantiles tau1 and tau2, the test of the null hypothesis H0 : b_tau1 = b_tau2 is obtained with the following procedure : 1. estimate the difference diff = b_tau1 - b_tau2 on the initial sample (ie from model m) 2. estimate the difference diff_b = b_tau1^b - b_tau2^b on each of the B bootstrap samples 3. compute se_diff, the empirical standard error of these B differences 4. the associated p-value is obtained with the Gaussian assumption (p-value = 2*P(N(0,1) > abs(diff/se_diff)))

Value

A list containing :

`results0`	a matrix with 3 columns containing the results (difference of the coefficients, standard erros of the diffrence and assocated p-values) without computing the weights in each bootstrap sample. Or NULL if the bootstrap results are obtained with wcompute=1.
`results1`	a matrix with 3 columns containing the results (difference of the coefficients, standard erros of the diffrence and assocated p-values) with re-estimated weights on each bootstrap sample. Or NULL if the bootstrap results are obtained with wcompute=0.

Author(s)

Viviane Philipps

`weightsIMD` *Estimation of observation-specific weights with intermittent missing data*

Description

This function provides stabilized weights for incomplete longitudinal data selected by death. The procedure allows intermittent missing data and assumes a missing at random (MAR) mechanism. Weights are defined as the inverse of the probability of being observed. These are obtained by pooled logistic regressions.

Usage

```
weightsIMD(data, Y, X1, X2, subject, death, time, impute = 0, name = "weight")
```

Arguments

`data`	data frame containing the observations and all variables named in `Y`, `X1`, `X2`, `subject`, `death` and `time` arguments.
`Y`	character indicating the name of the response outcome
`X1`	character vector indicating the name of the covariates with interaction with the outcome Y in the logistic regressions
`X2`	character vector indicating the name of the covariates without interaction with the outcome Y in the logistic regressions
`subject`	character indicating the name of the subject identifier
`death`	character indicating the time of death variable
`time`	character indicating the measurement time variable. Time should be 1 for the first (theoretical) visit, 2 for the second (theoretical) visit, etc.
`impute`	numeric indicating the value to impute if the outcome Y is missing
`name`	character indicating the name of the weight variable that will be added to the data

Details

Denoting T_i the death time, R_ij the observation indicator for subject i and occasion j, t the time, Y the outcome and X1 and X2 the covariates, we propose weights for intermittent missing data defined as :

w_ij = P(R_ij = 1 | T_i > t_ij, X1_ij, X2_ij) / P(R_ij = 1 | T_i > t_ij, X1_ij, X2_ij, Y_ij-1)

The numerator corresponds to the conditional probability of being observed in the population currently alive under the MCAR assumption.

The denominator is computed by recurrence :

P(R_ij = 1 | T_i > t_ij, X1_ij, X2_ij, Y_ij-1) =

P(R_ij = 1 | T_i > t_ij-1, X1_ij, X2_ij, Y_ij-1, R_ij-1 = 0) * P(R_ij-1 = 0 | T_i > t_ij, X1_ij, X2_ij, Y_ij-1) + P(R_ij = 1 | T_i > t_ij-1, X1_ij, X2_ij, Y_ij-1, R_ij-1 = 1) * P(R_ij-1 = 1 | T_i > t_ij, X1_ij, X2_ij, Y_ij-1)

Under the MAR assumption, the conditional probabilities lambda_ij = P(R_ij = 1 | T_i > t_ij, X1_ij, X2_ij, Y_ij-1, R_ij-1) are obtained from the logistic regression :

logit(lambda_ij) = b_0j + b_1 X1_ij + b_2 X2_ij + b_3 Y_i(j-1) + b_4 X1_ij Y_i(j-1) + b_5 (1-R_ij-1)

Value

A list containing :

`data`	the data frame with initial data and estimated weights as last column
`coef`	a list containing the estimates of the logistic regressions. The first element of coef contains the estimates under the MCAR assumption, the second contains the estimates under the MAR assumption.
`se`	a list containing the standard erros of the estimates contained in coef, in the same order.

Author(s)

Viviane Philipps, Marion Medeville, Anais Rouanet, Helene Jacqmin-Gadda

See Also

`weightsMMD`

Examples

```
w_simdata <- weightsIMD(data=simdata,Y="Y",X1="X",X2=NULL,subject="id",
death="death",time="time",impute=20,name="w_imd")$data
```

---

weightsMMD   *Estimation of observation-specific weights with monotone missing data*

Description

This function provides stabilized weights for incomplete longitudinal data selected by death. The procedure assumes monotne missing data and a MAR-S mechanism, that is the probability of being observed depends also on further death. Weights are defined as the inverse of the probability of being observed. These are obtained by pooled logistic regressions.

Usage

```
weightsMMD(data, Y, X1, X2, subject, death, time, interval.death = 0, name = "weight")
```

Arguments

`data`	data frame containing the observations and all variables named in `Y`, `X1`, `X2`, `subject`, `death` and `time` arguments.
`Y`	character indicating the name of the response outcome
`X1`	character vector indicating the name of the covariates with interaction with the outcome Y in the logistic regressions
`X2`	character vector indicating the name of the covariates without interaction with the outcome Y in the logistic regressions
`subject`	character indicating the name of the subject identifier
`death`	character indicating the time of death variable
`time`	character indicating the measurement time variable. Time should be 1 for the first visit, 2 for the second visit, etc.
`interval.death`	integer vector, intervals (j-k) to consider for the MAR-S hypothesis (see details). By default, interval.death=0, estimation under the MAR assumption.
`name`	character indicating the name of the weight variable that will be added to the data

Details

In longitudinal studies, follow-up can be truncated by death. Different missingness mechanism can be assumed. Missing data can be : 1. MCAR (completely at random) if the missingness probability is independent from the outcome and the death time 2. MAR (missing at random) if the probability is independent from the unobserved values of the outcome and from the death time 3. MAR-S if the probability is independent from the unobserved values but is different according to the death time 4. MNAR (missing not at random) if the probability may depend on unobserved values.

Denoting T_i the death time, R_ij the observation indicator for subject i and occasion j, t the time, Y the outcome and X1 and X2 the covariates, we propose weights for monotone missing data defined as :

w_ij = P(R_ij = 1 | T_i > t_ij, X1_ij, X2_ij) / P(R_ij = 1 | T_i > t_ij, X1_ij, X2_ij, Y_ij-1)

The numerator corresponds to the conditional probability of being observed in the population currently alive under the MCAR assumption.

The denominator is computed as a telescoping product :

P(R_ij = 1 | T_i > t_ij, X1_ij, X2_ij, Y_ij-1) =

prod_k=2^j P(R_ik = 1 | R_ik-1 = 1, T_i > t_ij, X1_ij, X2_ij, Y_ij-1) =

prod_k=2^j lambda_ijk

The probability lambda_ijk are obtained by logistic regressions.

Under the MAR-S assumption, the regression model is :

logit(lambda_ijk) = b_0k(j-k) + b_1(j-k) X1_ik + b_2(j-k) Y_i(k-1) + b_3(j-k) X1_ik Y_i(k-1) + b_4(j-k) X2_ik

For each interval (j-k), one logistic regression is performed.

Under the MAR assumption, one logistic regression is performed :

logit(lambda_ikk) = b_0k + b_1 X1_ik + b_2 X2_ik + b_3 Y_i(k-1) + b_4 X1_ik Y_i(k-1)

Value

A list containing :

- `data` — the data frame with initial data and estimated weights as last column
- `coef` — a list containing the estimates of the logistic regressions. The first element of coef contains the estimates under the MCAR assumption, the further contain the estimates under the MAR or MAR-S assumption.
- `se` — a list containing the standard erros of the estimates contained in coef, in the same order.

Author(s)

Viviane Philipps, Marion Medeville, Anais Rouanet, Helene Jacqmin-Gadda

See Also

weightsIMD

Examples

```
w_simdata <- weightsMMD(data=simdata,Y="Ytrunc",X1="X", X2=NULL,
subject="id", death="death", time="time", interval.death = 0)$data
```

Index