Package: CaseCohortCoxSurvival (via r-universe)

February 21, 2025

Title Case-Cohort Cox Survival Inference

Version 0.0.36

Date 2024-09-23

Maintainer Lola Etievant <`lola.etievant@gmail.com`>

Description Cox model inference for relative hazard and covariate-specific pure risk estimated from stratified and unstratified case-cohort data as described in Etievant, L., Gail, M.H. (Lifetime Data Analysis, 2024) <doi:10.1007/s10985-024-09621-2>.

Depends R (>= 3.5.0), survival

License GPL-2

Imports nnet

NeedsCompilation yes

Author Lola Etievant [cre, aut], Mitchell H. Gail [aut], Bill Wheeler [aut]

Date/Publication 2024-09-23 22:00:25 UTC

Repository https://etievant.r-universe.dev

RemoteUrl https://github.com/cran/CaseCohortCoxSurvival

RemoteRef HEAD

RemoteSha 4bf498e51a7deb8bc79b447987625b8548461aad

Contents

`CaseCohortCoxSurvival-package`

Case-Cohort Cox Survival Inference

Description

This package uses case-cohort data to estimate log-relative hazard, baseline hazards at each unique event time, cumulative baseline hazard in a given time interval and pure risk on the time interval and for a given covariate profile, under the Cox model. For the corresponding variance estimation, it relies on influence functions and follows the complete variance decomposition, to enable correct analysis of case-cohort data with and without stratification, weight calibration or missing phase-two covariate data.

Details

The package provides functions implementing the methods described in Etievant and Gail (2024). More precisely, it includes

- a main driver function, `caseCohortCoxSurvival`.
- one function, `estimatePureRisk`, to estimate pure risks and the corresponding variances with additional covariate profiles.
- three functions, `estimation`, `estimation.CumBH` and `estimation.PR`, for parameters estimation.

- four functions, `influences`, `influences.RH`, `influences.CumBH` and `influences.PR`, for influence functions derivation when estimation is with design or calibrated weights and from a case-cohort consisting of the subcohort and cases not in the subcohort (i.e., case-cohort obtained from two phases of sampling).
- four functions, `influences.missingdata`, `influences.RH.missingdata`, `influences.CumBH.missingdata` and `influences.PR.missingdata`, for influence functions derivation when estimation is with design weights and from a case-cohort when covariate information was missing for certain individuals in the phase-two data (i.e., case-cohort obtained from three phases of sampling).
- two functions, `variance` and `variance.missingdata`, for variance estimation following complete variance decomposition (with design or calibrated weights and without missing phase-two data, or with design weights and missing phase-two covariate data).
- one function, `robustvariance`, for robust variance estimation.
- one function, `auxiliary.construction`, to compute the auxiliary variables proposed by Breslow et al. (Stat. Biosci., 2009), Breslow and Lumley (IMS, 2013), and Shin et al. (Biometrics, 2020),.
- one function, `calibration`, for weight calibration.
- one function, `estimation.weights.phase3`, for estimating the phase-three weights.

Author(s)

Lola Etievant, Mitchell H. Gail

References

Etievant, L., Gail, M. H. (2024). Cox model inference for relative hazard and pure risk from stratified weight-calibrated case-cohort data. Lifetime Data Analysis, 30, 572-599.

Etievant, L., Gail, M. H. (2024). Software Application Profile: CaseCohortCoxSurvival: an R package for case-cohort inference for relative hazard and pure risk under the Cox model. Submitted.

`auxiliary.construction` *auxiliary.construction*

Description

Creates the auxiliary variables proposed by Breslow et al. (Stat. Biosci., 2009), Breslow and Lumley (IMS, 2013), and proposed by Shin et al. (Biometrics, 2020).

Usage

```
auxiliary.construction(mod, Tau1 = NULL, Tau2 = NULL, method = "Breslow",
time.on.study = NULL, casecohort = NULL)
```

Arguments

`mod`	A cox model object, result of function coxph run on the cohort data with imputed covariate values.
`Tau1`	Left bound of the time interval considered for the cumulative baseline hazard. Default is the first event time.
`Tau2`	Right bound of the time interval considered for the cumulative baseline hazard. Default is the last event time.
`method`	"Breslow", "Breslow2013" or "Shin" to specify the algorithm to construct the auxiliary variables. The default is "Breslow".
`time.on.study`	Total folow-up time in `[Tau1, Tau2]`. Required for `method` = "Shin".
`casecohort`	Data frame containing the casecohort data. It must include columns "weights" containing the design weights and "id" as an id variable. Required for `method` = "Shin".

Details

Construction of the auxiliary variables can follow Breslow et al. (2009), Breslow and Lumley (2013), or Shin et al. (2020) (method). It relies on predictions of the phase-two covariates for all members of the cohort. The auxiliary variables are given by (i) the influences for the log-relative hazard parameters estimated from the Cox model with imputed cohort data; (ii) the influences for the cumulative baseline parameter estimated from the Cox model with imputed cohort data; (iii) the products of total follow-up time (on the time interval for which pure risk is to be estimated) with the estimated relative hazard for the imputed cohort data, where the log-relative hazard parameters are estimated from the Cox model with case-cohort data and weights calibrated with (i). When method = Breslow, calibration of the design weights is against (i), as proposed by Breslow et al. (2009) to improve efficiency of case-cohort estimates of relative hazard. When method = Breslow2013, calibration of the design weights is against (i) and (ii), as proposed by Breslow and Lumley (2013) to also improve efficiency of case-cohort estimates of cumulative baseline hazard. When method = Shin, calibration is against (i) and (iii), as proposed by Shin et al. (2020) to improve efficiency of relative hazard and pure risk estimates under the nested case-control design. See Etievant and Gail (2024).

Following Etievant and Gail (2024), in function `caseCohortCoxSurvival` we only provide calibration of the design weight as proposed by Breslow et al. (2009) or Shin et al. (2020).

Value

`A.RH.Breslow`: matrix with the influences on the log-relative hazard, estimated from the cohort with imputed phase-two covariate values for `method` = "Breslow" and `method` = "Breslow2013".

`A.CumBH.Breslow`: matrix with the influences on the cumulative baseline hazard in `[Tau1, Tau2]`, estimated from the cohort with imputed phase-two covariate values for `method` = "Breslow2013".

`A.RH.Shin`: matrix with the influences on the log-relative hazard, estimated from the cohort with imputed phase-two covariate values for `method` = "Shin".

`A.PR.Shin`: matrix with the products of total follow-up times in `[Tau1, Tau2]` and estimated relative hazards, estimated from the cohort with imputed phase-two covariate values for `method` = "Shin".

References

Breslow, N.E. and Lumley, T. (2013). Semiparametric models and two-phase samples: Applications to Cox regression. From Probability to Statistics and Back: High-Dimensional Models and Processes, 9, 65-78.

Breslow, N.E., Lumley, T., Ballantyne, C.M., Chambless, L.E. and Kulich, M. (2009). Improved Horvitz-Thompson Estimation of Model Parameters from Two-phase Stratified Samples: Applications in Epidemiology. Statistics in Biosciences, 1, 32-49.

Shin Y.E., Pfeiffer R.M., Graubard B.I., Gail M.H. (2020) Weight calibration to improve the efficiency of pure risk estimates from case-control samples nested in a cohort. Biometrics, 76, 1087-1097.

Etievant, L., Gail, M. H. (2024). Cox model inference for relative hazard and pure risk from stratified weight-calibrated case-cohort data. Lifetime Data Analysis, 30, 572-599.

See Also

`calibration`, `influences`, `influences.RH`, `influences.CumBH` and `influences.PR`.

Examples

```
data(dataexample.stratified, package="CaseCohortCoxSurvival")
cohort <- dataexample.stratified$cohort
Tau1   <- 0
Tau2   <- 8

# Running the coxph model on the imputed cohort data
mod.imputedcohort <- coxph(Surv(event.time, status) ~ X1.pred + X2 + X3.pred,
                           data = cohort, robust = TRUE)

# method = Breslow
ret <- auxiliary.construction(mod.imputedcohort)
# print auxiliary variables based on the log-relative hazard influences
ret$A.RH.Breslow[1:5,]

# Example for method = Shin, variables names must match with fitted model
casecohort <- cohort[which(cohort$status == 1 |
                     cohort$subcohort == 1),] # the stratified case-cohort
casecohort$weights <- casecohort$strata.n / casecohort$strata.m
casecohort$weights[which(casecohort$status == 1)] <- 1
casecohort[, "X1.pred"] <- casecohort[, "X1"]
casecohort[, "X3.pred"] <- casecohort[, "X3"]

time.on.study <- pmax(pmin(Tau2, cohort$event.time) - Tau1, 0)
ret <- auxiliary.construction(mod.imputedcohort, method = "Shin",
                              time.on.study = time.on.study, casecohort = casecohort)
ret$A.PR.Shin[1:5]
```

`calibration` *calibration*

Description

Calibrates the design weights using the raking procedure.

Usage

```
calibration(A.phase2, design.weights, total, eta0 = NULL, niter.max = NULL,
epsilon.stop = NULL)
```

Arguments

`A.phase2`	matrix with the values of the q auxiliary variables to be used for the calibration of the weights in the case-cohort (phase-two data).
`design.weights`	design weights to be calibrated.
`total`	vector of length q with un-weighted auxiliary variable totals in the whole cohort.
`eta0`	vector of length q with initial values for `eta` (the Lagrangian multipliers), to be used as seed in the iterative procedure. Default is (0, ... ,0).
`niter.max`	maximum number of iterations for the iterative optimization algorithm. Default is 10^4 iterations.
`epsilon.stop`	threshold for the difference between the estimated weighted total and the total in the whole cohort. If this difference is less than the value of epsilon.stop, no more iterations will be performed. Default is 10^(-10).

Details

Calibration matches the weighted total of the auxiliary variables in the case-cohort (with calibrated weights), to the un-weighted auxiliary variables total in the whole cohort. In other words, it solves in η $\sum_{j=1}^{J} \sum_{i=1}^{n^{(j)}} \{\xi_{i,j} w_{i,j} \text{exp}(\eta' A_{i,j}) A_{i,j} - A_{i,j}\} = 0$, with $\xi_{i,j}$ the sampling indicator and $w_{i,j}$ the design weight of individual i in stratum j, and with $\sum_{j=1}^{J} \sum_{i=1}^{n^{(j)}} A_{i,j}$ the total in the whole cohort. See Etievant and Gail (2024). The Newton Raphson method is used to solve the optimization problem. In the end, the calibrated weights of the case-cohort individuals are given by $w_{i,j} \text{exp}(\hat{\eta}' A_{i,j})$, and $\sum_{j=1}^{J} \sum_{i=1}^{n^{(j)}} \{\xi_{i,j} w_{i,j} \text{exp}(\hat{\eta}' A_{i,j}) A_{i,j}\}$ gives the estimated total.

Value

`eta.hat`: vector of length q with final eta values.

`calibrated.weights`: vector with the calibrated weights for the individuals in the case-cohort (phase-two data), computed from `design.weights`, `A.phase2` and `eta.hat`.

`estimated.total`: vector with the estimated totals, computed from the `calibrated.weights` and `A.phase2`.

References

Deville, J.C. and Sarndal, C.E. (1992). Calibration Estimators in Survey Sampling. Journal of the American Statistical Association, 87, 376-382.

Etievant, L., Gail, M. H. (2024). Cox model inference for relative hazard and pure risk from stratified weight-calibrated case-cohort data. Lifetime Data Analysis, 30, 572-599.

See Also

`auxiliary.construction`, `influences`, `influences.RH`, `influences.CumBH` and `influences.PR`.

Examples

```
data(dataexample.stratified, package="CaseCohortCoxSurvival")
cohort <- dataexample.stratified$cohort
casecohort <- cohort[which(cohort$status == 1 |
                     cohort$subcohort == 1),] # the stratified case-cohort
casecohort$weight <- casecohort$strata.n / casecohort$strata.m
casecohort$weight[which(casecohort$status == 1)] <- 1

A <- dataexample.stratified$A # auxiliary variables values in the cohort
indiv.phase2 <- casecohort$id
q <- ncol(A)
total <- colSums(A)
A.phase2 <- A[indiv.phase2,]
calib <- calibration(A.phase2 = A[indiv.phase2,], design.weights = casecohort$weight,
              total = total, eta0 = rep(0, q), niter.max = 10^3, epsilon.stop = 10^(-10))
#calib$calibrated.weights # print calibrated weights
```

`caseCohortCoxSurvival` *Parameter and variance estimation for case-cohort analyses under the Cox model*

Description

Function for estimating parameters (log-relative hazard, baseline hazards, cumulative baseline hazard, pure risks) and their variance (robust or the one accounting for sampling features) from cohort or case-cohort data, under the Cox model.

Usage

```
caseCohortCoxSurvival(data, status, time, cox.phase1 = NULL, cox.phase2 = NULL,
other.covars = NULL, strata = NULL, weights.phase2 = NULL, calibrated = FALSE,
subcohort = NULL, subcohort.strata.counts = NULL, predict = TRUE,
predicted.cox.phase2 = NULL, predictors.cox.phase2 = NULL, aux.vars = NULL,
aux.method = "Shin", phase3 = NULL, strata.phase3 = NULL, weights.phase3 = NULL,
weights.phase3.type = "both", Tau1 = NULL, Tau2 = NULL, x = NULL,
weights.op = NULL, print = 1)
```

Arguments

`data`	Data frame containing the cohort and all variables needed for the analysis.
`status`	Column name in `data` giving the case status for each individual in the cohort. This variable must be coded as 0 for non-cases and 1 for cases.
`time`	Column name(s) in `data` giving the time to event for each individual in the case-cohort. One variable is required for a time-on-study time scale, two variables for age-scale, with the first variable as the start age and second as the end age.
`cox.phase1`	Column name(s) in `data` giving the Cox model covariates measured on the entire cohort. See covariates and prediction in details.
`cox.phase2`	Column name(s) in `data` giving the Cox model covariates measured only on phase-two individuals. See covariates and prediction in details.
`other.covars`	Column name(s) in data giving other covariates measured on the entire cohort that might be useful, alone or in combination with `cox.phase1`, if predicted values of the phase-two covariates (`cox.phase2`) need to be obtained on the whole cohort for the weight calibration.
`strata`	NULL or column name in data with the stratum value for each individual in the cohort. The number of strata used for the sampling of the subcohort equals the number of different stratum values. For example, a stratum variable might take values 0,1,2,3 or 4. The default is NULL.
`weights.phase2`	NULL or column name in data giving the phase-two design weights for each individual in the cohort. For a whole cohort analysis (see `subcohort` below), weights are not used in the `coxph` call. If NULL but subcohort is not NULL, `subcohort.strata.counts` will be used to estimate `weights.phase2`. The default is NULL.
`calibrated`	TRUE or FALSE to calibrate the `weights`. Calibrated weights will be computed using the function `calibration`. If TRUE, then `phase3` (below) will be set to NULL. See calibration in details. The default is FALSE.
`subcohort`	NULL or column name in `data` giving the indicators of membership in the subcohort. The indicators are 1 if the individual belongs to the subcohort and 0 otherwise. Some cases might be in the subcohort and others not. If NULL, then a whole cohort analysis will be performed. The default is NULL.
`subcohort.strata.counts`	NULL or a list of the number of individuals sampled into the subcohort from each stratum of strata. The names in the list must be the strata values and the length of the list must be equal to the number of strata. If NULL, then the count for each stratum is estimated by the number of subcohort individuals in each stratum. The default is NULL.
`predict`	TRUE or FALSE to predict the phase-two covariates using `predictors.cox.phase2`. This option is not used if `calibrated=FALSE`. If `calibrated=TRUE`, `aux.vars=NULL` and `predict=FALSE`, then `predicted.cox.phase2` must be specified. See covariates and prediction in details. This option is only used when `calibrated=TRUE`, `aux.vars=NULL` and `predicted.cox.phase2=NULL`. The default is TRUE.
`predicted.cox.phase2`	NULL or a named list giving the predicted values of the phase-two covariates (`cox.phase2`) on the whole cohort. For example, if the phase-two covariates are

	`X1` and `X2`, then the list is of the form `list(X1=X1.pred, X2=X2.pred)`, where `X1.pred` and `X2.pred` are the predictions of `X1` and `X2` respectively. This option is only used when `calibrated=TRUE` and `aux.vars=NULL`. If `calibrated=TRUE`, `aux.vars=NULL` and `predict=FALSE`, then `predicted.cox.phase2` must be specified and must not contain missing values. The default is NULL.
`predictors.cox.phase2`	NULL, a vector, or a list specifying the columns in data to use as predictor variables for obtaining the predicted values on the whole cohort for the phase-two covariates (`cox.phase2`). A list allows for different proxy variables to be used for the different phase-two covariates. The selected predictor variables must be from among `cox.phase1` and `other.covars`. See examples and covariates and prediction in details. If NULL, then the phase-two covariates will be predicted using `cox.phase1` and `other.covars`. If NULL, `cox.phase1=NULL` and `other.covars=NULL`, then the calibrated analysis will not be performed. This option is only used when `calibrated=TRUE`, `aux.vars=NULL`, `predicted.cox.phase2=NULL` and `predict=TRUE`. The default is NULL.
`aux.vars`	NULL or column name(s) in data giving the auxiliary variables for each individual in the cohort. This option is only used when `calibrated=TRUE`. If NULL, then auxiliary variables will be constructed using method Breslow or Shin and predicted values on the whole cohort for the phase-two covariates (see `aux.method`, `predict`, `predicted.cox.phase2` and `predictors.cox.phase2`). `aux.vars` must not contain missing values. The default is NULL.
`aux.method`	"Breslow", or "Shin" to specify the algorithm to construct the auxiliary variables. This option is only used if `aux.vars=NULL` and `calibrated=TRUE`. The default is "Shin".
`phase3`	NULL or column name in data giving the indicators of membership in the in the phase-three sample. The indicators are 1 if the individual belongs to the phase-three sample and 0 otherwise. All individuals in the phase-three sample must also belong to the phase-two sample. This option is not used if `calibrated=TRUE`. The default is NULL.
`strata.phase3`	NULL or column name in `data` giving the phase-three stratification for each individual in phase-two. The number of strata used for the third phase of sampling equals the number of different stratum values. The default is NULL.
`weights.phase3`	NULL or column name in `data` giving the phase-three design weights for each individual in phase-two. If NULL but `phase3` is not NULL, then `phase3` and `subcohort` will be used to estimate `weights.phase3` (see details in `estimation.weights.phase3`). The default is NULL.
`weights.phase3.type`	One of NULL, "design", "estimated", or "both" to specify whether the phase-three weights are design weights (known), or to be estimated. The variance estimation differs for estimated and design weights. If set to "both", then both variance estimates will be computed. If not NULL, then only the first letter is matched for this option. The default is "both".
`Tau1`	NULL or left bound of the time interval considered for the cumulative baseline hazard and the pure risk. If NULL, then the first event time is used.
`Tau2`	NULL or right bound of the time interval considered for the cumulative baseline hazard and the pure risk. If NULL, then the last event time is used.

`x`	Data frame containing `cox.phase1` and `cox.phase2` variables for which pure risk is estimated. The default is NULL so that no pure risk estimates will be computed.
`weights.op`	NULL or a list of options for calibration of phase-two design weights or estimating phase-three design weights. The available options are `niter.max`, and `epsilon.stop` (see `calibration` or `estimation.weights.phase3`). The default is NULL.
`print`	0-3 to print information as the analysis is performed. The larger the value, the more information will be printed. To not print any information, set `print = 0`. The default is 1.

Details

The different scenarios covered by the function are:
1) Whole cohort (`subcohort = NULL`)

2) (stratified) case-cohort (= stratified phase-two sample with no missing covariate data)
a. With design weights (`subcohort`, `strata`, `calibrated = FALSE`)
b. With calibrated weights and proxies to predict phase-two covariates and the auxiliary variables (`subcohort`, `strata`, `calibrated=TRUE`, `predict=TRUE`, `predictors.cox.phase2`, `aux.method`)
c. With calibrated weights and externally supplied predicted values of phase-two covariates (`calibrated=TRUE`, `strata`, `predict=FALSE`, `predicted.cox.phase2`)

3) (unstratified) case-cohort (= unstratified phase-two sample with no missing covariate data)
a. With design weights (`subcohort`, `strata=NULL`, `calibrated=FALSE`)
b. With calibrated weights and proxies to predict phase-two covariates and obtain the auxiliary variables (`subcohort`, `strata=NULL`, `calibrated=TRUE`, `predict=TRUE`, `predictors.cox.phase2`, `aux.method`)
c. With calibrated weights and externally supplied predicted values of phase-two covariates (`calibrated=TRUE`, `strata=NULL`, `predict=FALSE`, `predicted.cox.phase2`)

4) Case-cohort (= phase-three sample, because of missing covariate information in phase-two data, with stratified or unstratified phase-two sampling)
a. With known phase-three design weights (`subcohort`, `strata`, `phase3`, `strata.phase3`, `weights.phase3.type="design"`)
b. With estimated phase-three design weights (`subcohort`, `strata`, `phase3`, `strata.phase3`, `weights.phase3.type="estimated"`)

covariates and prediction
Prediction of phase-two covariates is performed when `calibrated = TRUE`, `predict = TRUE`, `aux.vars = NULL` and `predicted.cox.phase2 = NULL`. If `predictors.cox.phase2 = NULL`, all the covariates measured on the entire cohort will be used for the prediction (see `cox.phase1` and `other.covars`). Prediction of phase-two covariates is performed by linear regression for a continuous variable, logistic regression for a binary variable and the function `multinom` for a categorical variable. Dummy variables should not be used for categorical covariates, because independent logistic (or linear) regressions will be performed using the dummy variables.

Alternatively, predicted values of phase-two covariates on the whole cohort can be specified with `predicted.cox.phase2`.

calibration
Calibrating the design weights against some informative auxiliary variables, measured on all cohort members, can increase efficiency. When `calibrated = TRUE`, the user can either provide the auxiliary variables (`aux.vars`), or let the driver function build the auxiliary variables (`aux.method`). Construction of the auxiliary variables follows Breslow et al. (2009) or Shin et al. (2020) (see `aux.method`), and relies on predictions of the phase-two covariates for all members of the cohort (see covariates and prediction above). The auxiliary variables are given by (i) the influences for the log-relative hazard parameters estimated from the Cox model with imputed cohort data; and (ii) the products of total follow-up time (on the time interval for which pure risk is to be estimated) with the estimated relative hazard for the imputed cohort data, where the log-relative hazard parameters are estimated from the Cox model with case-cohort data and weights calibrated with (i). When `aux.method = Breslow`, calibration of the design weights is against (i), as proposed by Breslow et al. (2009) to improve efficiency of case-cohort estimates of relative hazard. When `aux.method = Shin`, calibration is against (i) and (ii), as proposed by Shin et al. (2020) to improve efficiency of relative hazard and pure risk estimates under the nested case-control design.

Note
If `subcohort = NULL`, then a whole cohort analysis will be run and only robust variance estimates will be computed.

Value

A list with class `casecohortcoxsurv` containing:

- `beta` Estimated log-relative hazard estimates
- `Lambda0` Cumulative baseline hazard estimate in [Tau1, Tau2]
- `beta.var` Influence-based variance estimate for `beta`
- `Lambda0.var` Influence-based variance estimate for `Lambda0`
- `beta.var.estimated` Influence-based variance estimate for `beta` with estimated phase-three weights
- `Lambda0.var.estimated` Influence-based variance estimate for `Lambda0` with estimated phase-three weights
- `beta.var.design` Influence-based variance estimate for `beta` with design phase-three weights
- `Lambda0.var.design` Influence-based variance estimate for `Lambda0` with design phase-three weights
- `beta.robustvar` Robust variance estimate for `beta`
- `Lambda0.robustvar` Robust variance estimate for `Lambda0`
- `beta.robustvar.estimated` Robust variance estimate for `beta` with estimated phase-three weights
- `Lambda0.robustvar.estimated` Robust variance estimate for `Lambda0` with estimated phase-three weights
- `beta.robustvar.design` Robust variance estimate for `beta` with design phase-three weights

- `Lambda0.robustvar.design` Robust variance estimate for `Lambda0` with design phase-three weights
- `Pi.var` Matrix of pure risk estimates in [Tau1, Tau2] and variance estimates
- `Pi.var.estimated` Matrix of pure risk estimates in [Tau1, Tau2] and variance estimates with estimated phase-three weights
- `Pi.var.design` Matrix of pure risk estimates in [Tau1, Tau2] and variance estimates with design phase-three weights
- `coxph.fit` Return object from `coxph` of the model fit
- `changed.times` Matrix of original and new event times for individuals who had their event times changed due to ties. Will be NULL if event times were not changed.
- `args` List containing the values of the input arguments (except `data`)
- `risk.obj` List containing objects needed to compute pure risk estimates and variances for a different set of data

References

Etievant, L., Gail, M. H. (2024). Cox model inference for relative hazard and pure risk from stratified weight-calibrated case-cohort data. Lifetime Data Analysis, 30, 572-599.

Etievant, L., Gail, M. H. (2024). Software Application Profile: CaseCohortCoxSurvival: an R package for case-cohort inference for relative hazard and pure risk under the Cox model. Submitted.

Shin Y.E., Pfeiffer R.M., Graubard B.I., Gail M.H. (2020) Weight calibration to improve the efficiency of pure risk estimates from case-control samples nested in a cohort. Biometrics, 76, 1087-1097.

Breslow, N.E., Lumley, T., Ballantyne, C.M., Chambless, L.E. and Kulich, M. (2009). Improved Horvitz-Thompson Estimation of Model Parameters from Two-phase Stratified Samples: Applications in Epidemiology. Statistics in Biosciences, 1, 32-49.

Examples

```
data(dataexample.missingdata.stratified, package="CaseCohortCoxSurvival")
data <- dataexample.missingdata.stratified$cohort
cov1 <- "X2"
cov2 <- c("X1", "X3")

# Whole cohort, get pure risk estimate for every individual's profile in the
# cohort. Only robust variance estimates are computed for a whole cohort analysis
caseCohortCoxSurvival(data = data, status = "status", time = "event.time",
                      cox.phase1 = cov1, x = data)

# Stratified case-cohort analysis with missing covariate information in the
# phase-two data, and with phase-three strata based on W3
caseCohortCoxSurvival(data = data, status = "status", time = "event.time",
                      cox.phase1 = cov1, cox.phase2 = cov2, strata = "W",
                      subcohort = "subcohort", phase3 = "phase3",
                      strata.phase3 = "W3")
```

```
# Stratified case-cohort (phase-two) analysis with weight calibration specifying
# a different set of proxy variables to predict each phase-two covariate
data(dataexample.stratified, package="CaseCohortCoxSurvival")
data <- dataexample.stratified$cohort
cov1 <- "X2"
cov2 <- c("X1", "X3")

caseCohortCoxSurvival(data = data, status = "status", time = "event.time",
                      cox.phase1 = cov1, cox.phase2 = cov2, strata = "W",
                      subcohort = "subcohort", calibrated = TRUE,
                      predictors.cox.phase2 = list(X1 = c("X1.proxy", "W"),
                                                   X3 = c("X1.proxy", "X3.proxy", "X2")))

# Stratified case-cohort (phase-two) analysis with weight calibration, get pure
# risk estimate for one given covariate profile
est <- caseCohortCoxSurvival(data = data, status = "status", time = "event.time",
                             cox.phase1 = cov1, cox.phase2 = cov2, strata = "W",
                             subcohort = "subcohort", calibrated = TRUE,
                             predictors.cox.phase2 = list(X1 = c("X1.proxy", "W"),
                                                   X3 = c("X1.proxy", "X3.proxy", "X2")),
                             x = list(X1 = 1, X2 = -1, X3 = 0.6), Tau1 = 0, Tau2 = 8)

est$Pi.var

# Stratified case-cohort (phase-two) analysis with weight calibration, get pure
# risk estimate for two given covariate profiles
pr1 <- as.data.frame(cbind(X1 = -1, X2 = 1, X3 = -0.6))
pr2 <- as.data.frame(cbind(X1 = 1, X2 = -1, X3 = 0.6))

est <- caseCohortCoxSurvival(data = data, status = "status", time = "event.time",
                             cox.phase1 = cov1, cox.phase2 = cov2, strata = "W",
                             subcohort = "subcohort", calibrated = TRUE,
                             predictors.cox.phase2 = list(X1 = c("X1.proxy", "W"),
                                                   X3 = c("X1.proxy", "X3.proxy", "X2")),
                             x = rbind(pr1, pr2), Tau1 = 0, Tau2 = 8)

est$Pi.var

# Stratified case-cohort (phase-two) analysis with design weights, get pure
# risk estimate for one given covariate profile
est <- caseCohortCoxSurvival(data = data, status = "status", time = "event.time",
                      cox.phase1 = cov1, cox.phase2 = cov2, strata = "W",
                      subcohort = "subcohort",
                      x = list(X1 = 1, X2 = -1, X3 = 0.6), Tau1 = 0, Tau2 = 8)
est$beta
est$Pi.var

# Set the correct sampling counts in phase-two for each level of strata.
# The strata variable W has levels 0-3.
est <- caseCohortCoxSurvival(data = data, status = "status", time = "event.time",
                             cox.phase1 = cov1, cox.phase2 = cov2, strata = "W",
                             subcohort = "subcohort",
                             subcohort.strata.counts = list("0" = 97, "1" = 294,
```

```
                                                         "2" = 300, "3" = 380))

est$beta
```

`dataexample` *[Deprecated] Data for examples*

Description

[`dataexample` is deprecated and will be removed in the next version of the package].

Simulated cohort, case-cohort and set of auxiliary variables for examples. The case-cohort is a stratified phase-two sample with no missing covariate data.

See Also

`dataexample.stratified`, `dataexample.unstratified`

Examples

```
data(dataexample, package="CaseCohortCoxSurvival")

# Display some of the data
dataexample$cohort[1:5, ]

dataexample$A[1:5, ] # auxiliary variable values in the cohort
```

`dataexample.missingdata` *[Deprecated] Data for examples with missing data*

Description

[`dataexample.missingdata` is deprecated and will be removed in the next version of the package].

Simulated cohort and case-cohort for examples. The case-cohort is a stratified phase-three sample, because of missing covariate information in the stratified phase-two data.

See Also

`dataexample.missingdata.stratified`, `dataexample.missingdata.unstratified`

Examples

```
data(dataexample.missingdata, package="CaseCohortCoxSurvival")

# Display some of the data
dataexample.missingdata$cohort[1:5, ]
```

`dataexample.missingdata.stratified` *Example of case-cohort with stratified sampling of the subcohort and missing covariate information in phase-two data*

Description

List with `cohort`.

`cohort` is a simulated cohort with 20 000 subjects. It contains:

`id` is the subject identifier.

`X1` is a continuous baseline covariate. Its measurements are only available for subjects in the case-cohort, i.e., with `phase3 = 1`.

`X2` is a categorical baseline covariate, with categories 0, 1, and 2. It is measured on all cohort subjects.

`X3` is a continuous baseline covariate. Its measurements are only available for subjects in the case-cohort.

`W` is a baseline categorical variable, with categories 0, 1, 2, and 3. It depends on predictors of `X1` and `X2`. It is measured on all cohort subjects.

`status` indicates case status.

`event.time` gives the event or censoring time. `status` indicates whether the subject experienced the event of interest or was censored.

The stratified sampling of the subcohort was based on the 4 strata defined by `W`. 97, 294, 300, and 380 subjects were sampled (independently of case status) from the 4 strata, respectively. `subcohort` indicates all these subjects included in the subcohort.

The phase-two sample consisted of the subcohort and any other cases not in the subcohort. `phase2` indicates all these subjects included in the phase-two sample.

`W3` is a baseline binary variable, based on case status. It is measured on all cohort subjects.

The third phase of sampling was stratified based on the 2 strata defined by `W3`. Subjects were sampled from the 2 strata with sampling probabilities 0.9 and 0.8. `phase3` indicates all these subjects included in the case-cohort (phase-three sample).

`strata.n` gives the number of subjects in the stratum in the cohort.

`strata.m` gives the number of subjects sampled from each of the 4 phase-two strata to be included in the subcohort (i.e., 97, 294, 300, or 380).

`strata.m` and `strata.n` would be used to compute the phase-two design weights of non-cases. Because all the cases were included in the phase-two sample, they would be assigned a phase-two design weight of 1.

`strata.n.cases` gives the number of cases in each of the 4 phase-two strata in the cohort.

`n.cases` gives the number of cases in the entire cohort.

`strata.proba.missing` gives the the sampling probablity for the 2 phase-three strata based on `W3` and that were used for the third phase of sampling.

`weight.true` gives the true design weight (i.e., product of the phase-two and true phase-three design weight).

`weight.p2.true` gives true phase-two design weight. They are stratum-specific based on `W`.

`weight.p3.true` gives the true phase-three design weight. They are stratum-specific based on `W3`. `weight.p3.true` can be used with argument `weights.phase3` of function `caseCohortCoxSurvival`, along with argument `weights.phase3.type = "design"`.

`weight.p3.est` gives the estimated phase-three design weight. They were obtained from `W3`, `phase2` and `phase3`. `weight.p3.est` can be used with argument `weights.phase3` of function `caseCohortCoxSurvival`, along with argument `weights.phase3.type = "estimated"`. If in function `caseCohortCoxSurvival` `weights.phase3 = NULL` but `weights.phase3.type = "estimated"`, the phase-three design weights will be estimated from `W3`, `phase2` and `phase3` and should be identical.

`weight.est` gives the estimated design weight (i.e., product of the phase-two and estimated phase-three design weight).

References

Etievant, L., Gail, M. H. (2024). Cox model inference for relative hazard and pure risk from stratified weight-calibrated case-cohort data. Lifetime Data Analysis, 30, 572-599.

Etievant, L., Gail, M. H. (2024). Software Application Profile: CaseCohortCoxSurvival: an R package for case-cohort inference for relative hazard and pure risk under the Cox model. Submitted.

Examples

```
data(dataexample.missingdata.stratified, package="CaseCohortCoxSurvival")

# Display some of the data
dataexample.missingdata.stratified$cohort[1:5, ]
```

`dataexample.missingdata.unstratified`

Example of case-cohort with unstratified sampling of the subcohort and missing covariate information in phase-two data

Description

List with `cohort`.

`cohort` is a simulated cohort with 20 000 subjects. It contains:

`id` is the subject identifier.

`X1` is a continuous baseline covariate. Its measurements are only available for subjects in the case-cohort, i.e., with `phase3 = 1`.

`X2` is a categorical baseline covariate, with categories 0, 1, and 2. It is measured on all cohort subjects.

`X3` is a continuous baseline covariate. Its measurements are only available for subjects in the case-cohort.

`status` indicates case status.

`event.time` gives the event or censoring time. `status` indicates whether the subject experienced the event of interest or was censored.

The sampling of the subcohort was not stratified. 1053 subjects were sampled (independently of case status) from the cohort. `subcohort` indicates all these subjects included in the subcohort.

The phase-two sample consisted of the subcohort and any other cases not in the subcohort. `phase2` indicates all these subjects included in the phase-two sample.

`W3` is a baseline binary variable, based on case status. It is measured on all cohort subjects.

The third phase of sampling was stratified based on the 2 strata defined by `W3`. Subjects were sampled from the 2 strata with sampling probabilities 0.9 and 0.8. `phase3` indicates all these subjects included in the case-cohort (phase-three sample).

`n` gives the number of subjects in the cohort.

`m` gives the number of subjects sampled from the cohort (i.e., 1053).

`m` and `n` would be used to compute the design weights of non-cases. Because all the cases were included in the case-cohort, they would be assigned a design weight of 1.

`n.cases` gives the number of cases in the entire cohort.

`W3` is a baseline binary variable, based on case status. It is measured on all cohort subjects.

`strata.proba.missing` gives the the sampling probablity for the 2 phase-three strata based on `W3` and that were used for the third phase of sampling.

`weight.true` gives the true design weight (i.e., product of the phase-two and true phase-three design weight).

`weight.p2.true` gives true phase-two design weight. They are stratum-specific based on `W`.

`weight.p3.true` gives the true phase-three design weight. They are stratum-specific based on `W3`. `weight.p3.true` can be used with argument `weights.phase3` of function `caseCohortCoxSurvival`, along with argument `weights.phase3.type = "design"`.

`weight.p3.est` gives the estimated phase-three design weight. They were obtained from `W3`, `phase2` and `phase3`. `weight.p3.est` can be used with argument `weights.phase3` of function `caseCohortCoxSurvival`, along with argument `weights.phase3.type = "estimated"`. If in function `caseCohortCoxSurvival` `weights.phase3 = NULL` but `weights.phase3.type = "estimated"`, the phase-three design weights will be estimated from `W3`, `phase2` and `phase3` and should be identical.

`weight.est` gives the estimated design weight (i.e., product of the phase-two and estimated phase-three design weight).

References

Etievant, L., Gail, M. H. (2024). Cox model inference for relative hazard and pure risk from stratified weight-calibrated case-cohort data. Lifetime Data Analysis, 30, 572-599.

Etievant, L., Gail, M. H. (2024). Software Application Profile: CaseCohortCoxSurvival: an R package for case-cohort inference for relative hazard and pure risk under the Cox model. Submitted.

Examples

```
data(dataexample.missingdata.unstratified, package="CaseCohortCoxSurvival")

# Display some of the data
dataexample.missingdata.unstratified$cohort[1:5, ]
```

`dataexample.stratified` *Example of case-cohort with stratified sampling of the subcohort, and set of auxiliary variables*

Description

List with `cohort` and `A`.

`cohort` is a simulated cohort with 20 000 subjects. It contains:

`id` is the subject identifier.

`X1` is a continuous baseline covariate. Its measurements are only available for subjects in the case-cohort, i.e., on subjects with `subcohort = 1` and/or `status = 1`.

`X2` is a categorical baseline covariate, with categories 0, 1, and 2. It is measured on all cohort subjects.

`X3` is a continuous baseline covariate. Its measurements are only available for subjects in the case-cohort.

`W` is a baseline categorical variable, with categories 0, 1, 2, and 3. It depends on predictors of `X1` and `X2`. It is measured on all cohort subjects. The stratified sampling of the subcohort was based on the 4 strata defined by `W`.

`status` indicates case status.

`event.time` gives the event or censoring time. `status` indicates whether the subject experienced the event of interest or was censored.

97, 294, 300, and 380 subjects were sampled (independently of case status) from the 4 strata, respectively. `subcohort` indicates all these subjects included in the subcohort. The stratified case-cohort (phase-two sample) consists of the subcohort and any other cases not in the subcohort.

`strata.n` gives the number of subjects in the stratum in the cohort.

`strata.m` gives the number of subjects sampled from each of the 4 strata (i.e., 97, 294, 300, or 380). `strata.m` and `strata.n` would be used to compute the stratum-specific design weights of non-cases. Because all the cases were included in the case-cohort, they would be assigned a design weight of 1.

`strata.n.cases` gives the number of cases in each of the 4 strata.

`n.cases` gives the number of cases in the entire cohort.

`X1.proxy` is a continuous baseline covariate. It is a proxy of `X1`, with 0.8 correlation. It is measured on all cohort subjects. It can be used for design weights calibration in the argument `predictors.cox.phase2` of function `caseCohortCoxSurvival`, as one would need to predict `X1` on the entire cohort.

`X3.proxy` is a continuous baseline covariate. It is a proxy of `X3`, with 0.8 correlation. It is measured on all cohort subjects. It can be used for design weights calibration in the argument `predictors.cox.phase2` of function `caseCohortCoxSurvival`, as one would need to predict `X3` on the entire cohort.

`X1.pred` is a prediction of `X1`, available for all cohort subjects. The predictions were obtained by weighted linear regression on `X1.proxy` and `W`, with the design weights.

`X3.pred` is a prediction of `X3`, available for all cohort subjects. The predictions were obtained by weighted linear regression on `X1.proxy`, `X2`, and `X3.proxy`, with the design weights.

`A` contains auxiliary variables, obtained as proposed by Breslow et al. (2009) and Shin et al. (2020). `A` can be used with argument `aux.var` of function `caseCohortCoxSurvival`.

Predictions of `X1` were obtained by weighted linear regression on `X1.proxy` and `W`, with the design weights. Predictions of `X3` were obtained by weighted linear regression on `X1.proxy`, `X2`, and `X3.proxy`, with the design weights. Then the Cox model with `X2` and the predicted values of `X1` and `X3` (available for all cohort subjects) was run. `A.X1`, `A.X2`, and `A.X3` contain the influences on the estimated log-RHs (available for all cohort subjects).

Second, design weights were then calibrated based on `A.1`, `A.X1`, `A.X2`, and `A.X3`, with `A.1` that is identically equal to 1. The log-RH parameter was then estimated from the case-cohort data with these calibrated weights. Finally, the log-RH estimate was used with `X2` and the predicted values of `X1` and `X3` (available for all cohort subjects), and exponentiated. `A.Shin` contains the product of this quantity with the total follow-up time on interval (0,8].

References

Etievant, L., Gail, M. H. (2024). Cox model inference for relative hazard and pure risk from stratified weight-calibrated case-cohort data. Lifetime Data Analysis, 30, 572-599.

Etievant, L., Gail, M. H. (2024). Software Application Profile: CaseCohortCoxSurvival: an R package for case-cohort inference for relative hazard and pure risk under the Cox model. Submitted.

Shin Y.E., Pfeiffer R.M., Graubard B.I., Gail M.H. (2020) Weight calibration to improve the efficiency of pure risk estimates from case-control samples nested in a cohort. Biometrics, 76, 1087-1097

Breslow, N.E., Lumley, T., Ballantyne, C.M., Chambless, L.E. and Kulich, M. (2009). Improved Horvitz-Thompson Estimation of Model Parameters from Two-phase Stratified Samples: Applications in Epidemiology. Statistics in Biosciences, 1, 32-49.

Examples

```
data(dataexample.stratified, package="CaseCohortCoxSurvival")

# Display some of the data
dataexample.stratified$cohort[1:5, ]

dataexample.stratified$A[1:5, ] # auxiliary variable values in the cohort
```

`dataexample.unstratified`

Example of case-cohort with unstratified sampling of the subcohort, and set of auxiliary variables

---

Description

List with `cohort` and `A`.

`cohort` is a simulated cohort with 20 000 subjects. It contains:

`id` is the subject identifier.

`X1` is a continuous baseline covariate. Its measurements are only available for subjects in the case-cohort, i.e., on subjects with `subcohort = 1` and/or `status = 1`.

`X2` is a categorical baseline covariate, with categories 0, 1, and 2. It is measured on all cohort subjects.

`X3` is a continuous baseline covariate. Its measurements are only available for subjects in the case-cohort.

`status` indicates case status.

`event.time` gives the event or censoring time. `status` indicates whether the subject experienced the event of interest or was censored.

1053 subjects were sampled (independently of case status) from the cohort. `subcohort` indicates all these subjects included in the subcohort. The case-cohort (phase-two sample) consists of the subcohort and any other cases not in the subcohort.

`n` gives the number of subjects in the cohort.

`m` gives the number of subjects sampled from the cohort (i.e., 1053).

`m` and `n` would be used to compute the design weights of non-cases. Because all the cases were included in the case-cohort, they would be assigned a design weight of 1.

`n.cases` gives the number of cases in the entire cohort.

`X1.proxy` is a continuous baseline covariate. It is a proxy of `X1`, with 0.8 correlation. It is measured on all cohort subjects. It can be used for design weights calibration in the argument `predictors.cox.phase2` of function `caseCohortCoxSurvival`, as one would need to predict `X1` on the entire cohort.

`X3.proxy` is a continuous baseline covariate. It is a proxy of `X3`, with 0.8 correlation. It is measured on all cohort subjects. It can be used for design weights calibration in the argument `predictors.cox.phase2` of function `caseCohortCoxSurvival`, as one would need to predict `X3` on the entire cohort.

`X1.pred` is a prediction of `X1`, available for all cohort subjects. The predictions were obtained by weighted linear regression on `X1.proxy`, with the design weights.

`X3.pred` is a prediction of `X3`, available for all cohort subjects. The predictions were obtained by weighted linear regression on `X1.proxy`, `X2`, and `X3.proxy`, with the design weights.

`A` contains auxiliary variables, obtained as proposed by Breslow et al. (2009) and Shin et al. (2020). `A` can be used with argument `aux.var` of function `caseCohortCoxSurvival`.

Predictions of `X1` were obtained by weighted linear regression on `X1.proxy` and `X2`, with the design weights. Predictions of `X3` were obtained by weighted linear regression on `X1.proxy`, `X2`, and `X3.proxy`, with the design weights. Then the Cox model with `X2` and the predicted values of `X1` and `X3` (available for all cohort subjects) was run. `A.X1`, `A.X2`, and `A.X3` contain the influences on the estimated log-RHs (available for all cohort subjects).

Second, design weights were then calibrated based on `A.1`, `A.X1`, `A.X2`, and `A.X3`, with `A.1` that is identically equal to 1. The log-RH parameter was then estimated from the case-cohort data with these calibrated weights. Finally, the log-RH estimate was used with `X2` and the predicted values of `X1` and `X3` (available for all cohort subjects), and exponentiated. `A.Shin` contains the product of this quantity with the total follow-up time on interval (0,8].

References

Etievant, L., Gail, M. H. (2024). Cox model inference for relative hazard and pure risk from stratified weight-calibrated case-cohort data. Lifetime Data Analysis, 30, 572-599.

Etievant, L., Gail, M. H. (2024). Software Application Profile: CaseCohortCoxSurvival: an R package for case-cohort inference for relative hazard and pure risk under the Cox model. Submitted.

Shin Y.E., Pfeiffer R.M., Graubard B.I., Gail M.H. (2020) Weight calibration to improve the efficiency of pure risk estimates from case-control samples nested in a cohort. Biometrics, 76, 1087-1097

Breslow, N.E., Lumley, T., Ballantyne, C.M., Chambless, L.E. and Kulich, M. (2009). Improved Horvitz-Thompson Estimation of Model Parameters from Two-phase Stratified Samples: Applications in Epidemiology. Statistics in Biosciences, 1, 32-49.

Examples

```
data(dataexample.unstratified, package="CaseCohortCoxSurvival")

# Display some of the data
dataexample.unstratified$cohort[1:5, ]

dataexample.unstratified$A[1:5, ] # auxiliary variable values in the cohort
```

`deprecateddata` *Deprecated data sets in CaseCohortCoxSurvival*

Description

These data sets still work but will be removed (defuncted) in the next version of the package.

`dataexample` is deprecated and will be removed in the next version of the package.

`dataexample.missingdata` is deprecated and will be removed in the next version of the package.

See Also

`dataexample.stratified`, `dataexample.unstratified`, `dataexample.missingdata.stratified`, `dataexample.missingdata.unstratified`

`estimatePureRisk` *estimatePureRisk*

Description

Computes pure risk estimates and variances for new covariate values.

Usage

```
estimatePureRisk(obj, x)
```

Arguments

`obj`	Return object from `caseCohortCoxSurvival`.
`x`	Data frame or a list containing values of the covariates that were used when `caseCohortCoxSurvival` was called, and for which the pure risk is to be estimated.

Value

A list containing:

- `var` Matrix of pure risk estimates in [Tau1, Tau2] and variance estimates
- `var.estimated` Matrix of pure risk estimates in [Tau1, Tau2] and variance estimates when the phase-three weights are estimated
- `var.design` Matrix of pure risk estimates in [Tau1, Tau2] and variance estimates when the phase-three weights are known

Depending on the analysis run, some of the above objects will be NULL.

References

Etievant, L., Gail, M. H. (2024). Cox model inference for relative hazard and pure risk from stratified weight-calibrated case-cohort data. Lifetime Data Analysis, 30, 572-599.

See Also

`caseCohortCoxSurvival`

Examples

```
data(dataexample.stratified, package="CaseCohortCoxSurvival")

data <- dataexample.stratified$cohort
cov1 <- "X2"
cov2 <- c("X1", "X3")

obj <- caseCohortCoxSurvival(data = data, status = "status",
```

```
                                  time = "event.time", cox.phase1 = cov1,
                                  cox.phase2 = cov2, strata = "W",
                                  subcohort = "subcohort", Tau1 = 0, Tau2 = 8)

# get pure risk estimate for every individual's profile in the cohort
ret <- estimatePureRisk(obj, data)

# get pure risk estimate for one given covariate profile
ret <- estimatePureRisk(obj, list(X1 = 1, X2 = -1, X3 = 0.6))

# get pure risk estimates for two given covariate profiles
pr1 <- as.data.frame(cbind(X1 = -1, X2 = 1, X3 = -0.6))
pr2 <- as.data.frame(cbind(X1 = 1, X2 = -1, X3 = 0.6))
ret <- estimatePureRisk(obj, rbind(pr1, pr2))
ret$var
```

`estimation` *estimation*

Description

Estimates the log-relative hazard, baseline hazards at each unique event time, cumulative baseline hazard in a given time interval [Tau1, Tau2] and pure risk in [Tau1, Tau2] and for a given covariate profile x.

Usage

```
estimation(mod, Tau1 = NULL, Tau2 = NULL, x = NULL, missing.data = NULL,
riskmat.phase2 = NULL, dNt.phase2 = NULL, status.phase2 = NULL)
```

Arguments

`mod`	a Cox model object, result of function `coxph`.
`Tau1`	left bound of the time interval considered for the cumulative baseline hazard and pure risk. Default is the first event time.
`Tau2`	right bound of the time interval considered for the cumulative baseline hazard and pure risk. Default is the last event time.
`x`	vector of length p, specifying the covariate profile considered for the pure risk. Default is (0,...,0).
`missing.data`	was data on the p covariates missing for certain individuals in the phase-two data (i.e., was a third phase of sampling performed)? If `missing.data = TRUE`, the arguments below need to be provided. Default is `FALSE`.
`riskmat.phase2`	at risk matrix for the phase-two data at all of the case event times, even those with missing covariate data. Needs to be provided if `missing.data = TRUE`.
`dNt.phase2`	counting process matrix for failures in the phase-two data. Needs to be provided if `missing.data = TRUE` and `status.phase2 = NULL`.
`status.phase2`	vector indicating the case status in the phase-two data. Needs to be provided if `missing.data = TRUE` and `dNt.phase2 = NULL`.

Details

`estimation` returns the log-relative hazard estimates provided by `mod`, and estimates the baseline hazard point mass at any event time non-parametrically.

`estimation` works for estimation from a case-cohort with design weights or calibrated weights, when the case-cohort consists of the subcohort and cases not in the subcohort (i.e., case-cohort obtained from two phases of sampling), as well as with design weights when covariate data was missing for certain individuals in the phase-two data (i.e., case-cohort obtained from three phases of sampling).

Value

`beta.hat`: vector of length p with log-relative hazard estimates.

`lambda0.t.hat`: vector with baseline hazards estimates at each unique event time.

`Lambda0.Tau1Tau2.hat`: cumulative baseline hazard estimate in [Tau1, Tau2].

`Pi.x.Tau1Tau2.hat`: pure risk estimate in [Tau1, Tau2] and for covariate profile `x`.

References

Breslow, N. (1974). Covariance Analysis of Censored Survival Data. Biometrics, 30, 89-99.

Etievant, L., Gail, M. H. (2024). Cox model inference for relative hazard and pure risk from stratified weight-calibrated case-cohort data. Lifetime Data Analysis, 30, 572-599.

See Also

`estimation.CumBH`, `estimation.PR`, `influences`, `influences.RH`, `influences.CumBH`, `influences.PR`, `influences.missingdata`, `influences.RH.missingdata`, `influences.CumBH.missingdata`, and `influences.PR.missingdata`.

Examples

```
data(dataexample.missingdata.stratified, package="CaseCohortCoxSurvival")

cohort <- dataexample.missingdata.stratified$cohort
phase2 <- cohort[which(cohort$phase2 == 1),] # the phase-two sample
casecohort <- cohort[which(cohort$phase3 == 1),] # the stratified case-cohort

B.phase2 <- cbind(1 * (phase2$W3 == 0), 1 * (phase2$W3 == 1))
rownames(B.phase2)  <- cohort[cohort$phase2 == 1, "id"]
B.phase3 <- cbind(1 * (casecohort$W3 == 0), 1 * (casecohort$W3 == 1))
rownames(B.phase3)  <- cohort[cohort$phase3 == 1, "id"]
total.B.phase2 <- colSums(B.phase2)
J3 <- ncol(B.phase3)
n <- nrow(cohort)

# Quantities needed for estimation of the cumulative baseline hazard when
# covariate data is missing
mod.cohort <- coxph(Surv(event.time, status) ~ X2, data = cohort,
                    robust = TRUE) # X2 is available on all cohort members
mod.cohort.detail <- coxph.detail(mod.cohort, riskmat = TRUE)
```

```
riskmat.phase2 <- with(cohort, mod.cohort.detail$riskmat[phase2 == 1,])
rownames(riskmat.phase2) <- cohort[cohort$phase2 == 1, "id"]
observed.times.phase2 <- apply(riskmat.phase2, 1,
                               function(v) {which.max(cumsum(v))})
dNt.phase2 <- matrix(0, nrow(riskmat.phase2), ncol(riskmat.phase2))
dNt.phase2[cbind(1:nrow(riskmat.phase2), observed.times.phase2)] <- 1
dNt.phase2 <- sweep(dNt.phase2, 1, phase2$status, "*")
colnames(dNt.phase2) <- colnames(riskmat.phase2)
rownames(dNt.phase2) <- rownames(riskmat.phase2)

Tau1 <- 0 # given time interval for the pure risk
Tau2 <- 8
x <- c(-1, 1, -0.6) # given covariate profile for the pure risk

# Estimation using the stratified case cohort with true known design weights
mod.true <- coxph(Surv(event.time, status) ~ X1 + X2 + X3, data = casecohort,
                  weight = weight.true, id = id, robust = TRUE)

est.true <- estimation(mod.true, Tau1 = Tau1, Tau2 = Tau2, x = x,
                                    missing.data = TRUE,
                                    riskmat.phase2 = riskmat.phase2,
                                    dNt.phase2 = dNt.phase2)

# print the vector with log-relative hazard estimates
est.true$beta.hat

# print the cumulative baseline hazard estimate
est.true$Lambda0.Tau1Tau2.hat

# print the pure risk estimate
est.true$Pi.x.Tau1Tau2.hat
```

`estimation.CumBH` *estimation.CumBH*

Description

Estimates the log-relative hazard, baseline hazards at each unique event time and cumulative baseline hazard in a given time interval [Tau1, Tau2].

Usage

```
estimation.CumBH(mod, Tau1 = NULL, Tau2 = NULL, missing.data = FALSE,
riskmat.phase2 = NULL, dNt.phase2 = NULL, status.phase2 = NULL)
```

Arguments

`mod`	a Cox model object, result of function coxph.
`Tau1`	left bound of the time interval considered for the cumulative baseline hazard. Default is the first event time.
`Tau2`	right bound of the time interval considered for the cumulative baseline hazard. Default is the last event time.
`missing.data`	was data on the p covariates missing for certain individuals in the phase-two data (i.e., was a third phase of sampling performed)? If `missing.data = TRUE`, the arguments below need to be provided. Default is FALSE.
`riskmat.phase2`	at risk matrix for the phase-two data at all of the case event times, even those with missing covariate data. Needs to be provided if `missing.data = TRUE`.
`dNt.phase2`	counting process matrix for failures in the phase-two data. Needs to be provided if `missing.data = TRUE` and `status.phase2 = NULL`.
`status.phase2`	vector indicating the case status in the phase-two data. Needs to be provided if `missing.data = TRUE` and `dNt.phase2 = NULL`.

Details

`estimation.CumBH` returns the log-relative hazard estimates provided by `mod`, and estimates the baseline hazard point mass at any event time non-parametrically.

`estimation.CumBH` works for estimation from a case-cohort with design weights or calibrated weights, when the case-cohort consists of the subcohort and cases not in the subcohort (i.e., case-cohort obtained from two phases of sampling), as well as with design weights when covariate data was missing for certain individuals in the phase-two data (i.e., case-cohort obtained from three phases of sampling).

Value

`beta.hat`: vector of length p with log-relative hazard estimates.

`lambda0.t.hat`: vector with baseline hazards estimates at each unique event time.

`Lambda0.Tau1Tau2.hat`: cumulative baseline hazard estimate in [Tau1, Tau2].

References

Breslow, N. (1974). Covariance Analysis of Censored Survival Data. Biometrics, 30, 89-99.

Etievant, L., Gail, M. H. (2024). Cox model inference for relative hazard and pure risk from stratified weight-calibrated case-cohort data. Lifetime Data Analysis, 30, 572-599.

See Also

`estimation`, `estimation.PR`, `influences`, `influences.RH`, `influences.CumBH`, `influences.PR`, `influences.missingdata`, `influences.RH.missingdata`, `influences.CumBH.missingdata`, and `influences.PR.missingdata`

Examples

```
data(dataexample.missingdata.stratified, package="CaseCohortCoxSurvival")

cohort <- dataexample.missingdata.stratified$cohort
phase2 <- cohort[which(cohort$phase2 == 1),] # the phase-two sample
casecohort <- cohort[which(cohort$phase3 == 1),] # the stratified case-cohort

B.phase2 <- cbind(1 * (phase2$W3 == 0), 1 * (phase2$W3 == 1))
rownames(B.phase2)  <- cohort[cohort$phase2 == 1, "id"]
B.phase3 <- cbind(1 * (casecohort$W3 == 0), 1 * (casecohort$W3 == 1))
rownames(B.phase3)  <- cohort[cohort$phase3 == 1, "id"]
total.B.phase2 <- colSums(B.phase2)
J3 <- ncol(B.phase3)
n <- nrow(cohort)

# Quantities needed for estimation of the cumulative baseline hazard when
# covariate data is missing
mod.cohort <- coxph(Surv(event.time, status) ~ X2, data = cohort,
                    robust = TRUE) # X2 is available on all cohort members
mod.cohort.detail <- coxph.detail(mod.cohort, riskmat = TRUE)

riskmat.phase2 <- with(cohort, mod.cohort.detail$riskmat[phase2 == 1,])
rownames(riskmat.phase2) <- cohort[cohort$phase2 == 1, "id"]
observed.times.phase2 <- apply(riskmat.phase2, 1,
                               function(v) {which.max(cumsum(v))})
dNt.phase2 <- matrix(0, nrow(riskmat.phase2), ncol(riskmat.phase2))
dNt.phase2[cbind(1:nrow(riskmat.phase2), observed.times.phase2)] <- 1
dNt.phase2 <- sweep(dNt.phase2, 1, phase2$status, "*")
colnames(dNt.phase2) <- colnames(riskmat.phase2)
rownames(dNt.phase2) <- rownames(riskmat.phase2)

Tau1 <- 0 # given time interval for the pure risk
Tau2 <- 8
x <- c(-1, 1, -0.6) # given covariate profile for the pure risk

# Estimation using the stratified case cohort with true known design weights
mod.true <- coxph(Surv(event.time, status) ~ X1 + X2 + X3, data = casecohort,
                  weight = weight.true, id = id, robust = TRUE)

est.true <- estimation(mod.true, Tau1 = Tau1, Tau2 = Tau2, x = x,
                       missing.data = TRUE,
                       riskmat.phase2 = riskmat.phase2,
                       dNt.phase2 = dNt.phase2)

est.true <- estimation.CumBH(mod.true, Tau1 = Tau1, Tau2 = Tau2,
                                          missing.data = TRUE,
                                          riskmat.phase2 = riskmat.phase2,
                                          dNt.phase2 = dNt.phase2)

# print the cumulative baseline hazard estimate
est.true$Lambda0.Tau1Tau2.hat
```

`estimation.PR` *estimation.PR*

Description

Estimates the pure risk in the time interval [Tau1, Tau2] and for a covariate profile x, from the log-relative hazard and cumulative baseline hazard values.

Usage

```
estimation.PR(beta, Lambda0.Tau1Tau2, x = NULL)
```

Arguments

`beta`	vector of length p with log-relative hazard values.
`Lambda0.Tau1Tau2`	cumulative baseline hazard in [Tau1, Tau2].
`x`	vector of length p, specifying the covariate profile considered for the pure risk. Default is (0,...,0).

Value

`Pi.x.Tau1Tau2.hat`: pure risk estimate in [Tau1, Tau2] and for covariate profile x.

References

Etievant, L., Gail, M. H. (2024). Cox model inference for relative hazard and pure risk from stratified weight-calibrated case-cohort data. Lifetime Data Analysis, 30, 572-599.

See Also

`estimation`, `estimation.CumBH`, `influences`, `influences.RH`, `influences.CumBH`, `influences.PR`, `influences.missingdata`, `influences.RH.missingdata`, `influences.CumBH.missingdata`, and `influences.PR.missingdata`.

Examples

```
estimation.PR(beta = c(-0.2, 0.25, -0.3), Lambda0.Tau1Tau2 = 0.03,
              x = c(-1, 1, -0.6))
```

`estimation.weights.phase3` *estimation.weights.phase3*

Description

Estimates the weights for the third phase of sampling (due to missingness in phase two).

Usage

```
estimation.weights.phase3(B.phase3, total.phase2, gamma0 = NULL, niter.max = NULL,
epsilon.stop = NULL)
```

Arguments

`B.phase3`	matrix for the case-cohort (phase-three data), with phase-three sampling strata indicators. It should have as many columns as phase-three strata ($J^{(3)}$), with one 1 per row, to indicate the phase-three stratum position.
`total.phase2`	vector of length $J^{(3)}$ with un-weighted column totals for `B` in the phase-two data (i.e., using all the individuals, even the ones with missing covariate data).
`gamma0`	vector of length $J^{(3)}$ with initial values for γ (Lagrangian multipliers), to be used as seed in the iterative procedure. Default is (0,...,0).
`niter.max`	maximum number of iterations for the iterative optimization algorithm. Default is `10^4` iterations.
`epsilon.stop`	threshold for the difference between the estimated weighted total and the total in the whole cohort. If this difference is less than the value of `epsilon.stop`, no more iterations will be performed. Default is `10^(-10)`.

Details

`estimation.weights.phase3` estimates the phase-three sampling weights by solving in γ

$$\sum_{j=1}^{J}\sum_{i=1}^{n^{(j)}}\{\xi_{i,j}V_{i,j}\text{exp}(\gamma' B_{i,j})B_{i,j} - \xi_{i,j}B_{i,j}\} = 0,$$

with $\xi_{i,j}$ the phase-two sampling indicator and $V_{i,j}$ the phase-three sampling indicator of individual i in stratum j, and with $\sum_{j=1}^{J}\sum_{i=1}^{n^{(j)}} \xi_{i,j}B_{i,j}$ the total in the phase-two data. See Etievant and Gail (2024). The Newton Raphson method is used to solve the optimization problem.

In the end, the estimated weights are given by $\text{exp}(\hat{\gamma}' B_{i,j})$, and $\sum_{j=1}^{J}\sum_{i=1}^{n^{(j)}} \xi_{i,j}V_{i,j}\text{exp}(\hat{\gamma}' B_{i,j})B_{i,j}$ gives the estimated total.

Value

`gamma.hat`: vector of length $J^{(3)}$ with final gamma values.

`estimated.weights`: vector with the estimated phase-three weights for the individuals in the case-cohort (phase-three data), computed from `B.phase3` and `gamma.hat`.

`estimated.total`: vector with the estimated totals, computed from the `estimated.weights` and `B.phase3`.

References

Etievant, L., Gail, M. H. (2024). Cox model inference for relative hazard and pure risk from stratified weight-calibrated case-cohort data. Lifetime Data Analysis, 30, 572-599.

See Also

`influences.missingdata`, `influences.RH.missingdata`,`influences.CumBH.missingdata` and `influences.PR.missingdata`.

Examples

```
data(dataexample.missingdata.stratified, package="CaseCohortCoxSurvival")

cohort <- dataexample.missingdata.stratified$cohort
phase2 <- cohort[which(cohort$phase2 == 1),] # the phase-two sample
casecohort <- cohort[which(cohort$phase3 == 1),] # the stratified case-cohort

B.phase2 <- cbind(1 * (phase2$W3 == 0), 1 * (phase2$W3 == 1))
rownames(B.phase2) <- cohort[cohort$phase2 == 1, "id"]
B.phase3 <- cbind(1 * (casecohort$W3 == 0), 1 * (casecohort$W3 == 1))
rownames(B.phase3) <- cohort[cohort$phase3 == 1, "id"]
total.B.phase2 <- colSums(B.phase2)
J3 <- ncol(B.phase3)

estimation.weights.p3 <- estimation.weights.phase3(B.phase3 = B.phase3,
                                                    total.phase2 = total.B.phase2,
                                                    gamma0 = rep(0, J3),
                                                    niter.max = 10^(4),
                                                    epsilon.stop = 10^(-10))
```

`influences` *influences*

Description

Computes the influences on the log-relative hazard, baseline hazards at each unique event time, cumulative baseline hazard in a given time interval [Tau1, Tau2] and on the pure risk in [Tau1, Tau2] and for a given covariate profile x. Can take calibration of the design weights into account.

Usage

```
influences(mod, Tau1 = NULL, Tau2 = NULL, x = NULL, calibrated = NULL,
A = NULL)
```

Arguments

`mod`	a cox model object, result of function coxph.
`Tau1`	left bound of the time interval considered for the cumulative baseline hazard and pure risk. Default is the first event time.
`Tau2`	right bound of the time interval considered for the cumulative baseline hazard and pure risk. Default is the last event time.
`x`	vector of length p, specifying the covariate profile considered for the pure risk. Default is (0,...,0).
`calibrated`	are calibrated weights used for the estimation of the parameters? If `calibrated = TRUE`, the argument below needs to be provided. Default is `FALSE`.
`A`	$n \times q$ matrix with the values of the auxiliary variables used for the calibration of the weights in the whole cohort. Needs to be provided if `calibrated = TRUE`.

Details

`influences` works for estimation from a case-cohort with design weights or calibrated weights (case-cohort consisting of the subcohort and cases not in the subcohort, i.e., case-cohort obtained from two phases of sampling).

If covariate information is missing for certain individuals in the phase-two data (i.e., case-cohort obtained from three phases of sampling), use `influences.missingdata`.

`influences` uses the influence formulas provided in Etievant and Gail (2024).

If `calibrated = FALSE`, the infuences are only provided for the individuals in the case-cohort. If `calibrated = TRUE`, the influences are provided for all the individuals in the cohort.

Value

`infl.beta`: matrix with the overall influences on the log-relative hazard estimates.

`infl.lambda0.t`: matrix with the overall influences on the baseline hazards estimates at each unique event time.

`infl.Lambda0.Tau1Tau2.hat`: vector with the overall influences on the cumulative baseline hazard estimate in [Tau1, Tau2].

`infl.Pi.x.Tau1Tau2.hat`: vector with the overall influences on the pure risk estimate in [Tau1, Tau2] and for covariate profile `x`.

`infl2.beta`: matrix with the phase-two influences on the log-relative hazard estimates. Returned if `calibrated = TRUE`.

`infl2.lambda0.t`: matrix with the phase-two influences on the baseline hazards estimates at each unique event time. Returned if `calibrated = TRUE`.

`infl2.Lambda0.Tau1Tau2.hat`: vector with the phase-two influences on the cumulative baseline hazard estimate in [Tau1, Tau2]. Returned if `calibrated = TRUE`.

`infl2.Pi.x.Tau1Tau2.hat`: vector with the phase-two influences on the pure risk estimate in [Tau1, Tau2] and for covariate profile `x`. Returned if `calibrated` = `TRUE`.

`beta.hat`: vector of length p with log-relative hazard estimates.

`lambda0.t.hat`: vector with baseline hazards estimates at each unique event time.

`Lambda0.Tau1Tau2.hat`: cumulative baseline hazard estimate in [Tau1, Tau2].

`Pi.x.Tau1Tau2.hat`: pure risk estimate in [Tau1, Tau2] and for covariate profile `x`.

References

Etievant, L., Gail, M. H. (2024). Cox model inference for relative hazard and pure risk from stratified weight-calibrated case-cohort data. Lifetime Data Analysis, 30, 572-599.

See Also

`estimation`, `estimation.CumBH`, `estimation.PR`, `influences.RH`, `influences.CumBH`, `influences.PR`, `influences.missingdata`, `influences.RH.missingdata`, `influences.CumBH.missingdata`, `influences.PR.missing` `robustvariance` and `variance`.

Examples

```
data(dataexample.stratified, package="CaseCohortCoxSurvival")
cohort <- dataexample.stratified$cohort
casecohort <- cohort[which(cohort$status == 1 |
                     cohort$subcohort == 1),] # the stratified case-cohort
casecohort$weights <- casecohort$strata.n / casecohort$strata.m
casecohort$weights[which(casecohort$status == 1)] <- 1

Tau1 <- 0
Tau2 <- 8
x <- c(-1, 1, -0.6) # given covariate profile for the pure risk

# Estimation using the stratified case cohort with design weights
mod <- coxph(Surv(event.time, status) ~ X1 + X2 + X3, data = casecohort,
             weight = weights, id = id, robust = TRUE)
est <- influences(mod, Tau1 = Tau1, Tau2 = Tau2, x = x)

# print the vector with log-relative hazard estimates
est$beta.hat

# print the cumulative baseline hazard estimate
est$Lambda0.Tau1Tau2.hat

# print the pure risk estimate
est$Pi.x.Tau1Tau2.hat

# print the influences on the log-relative hazard estimates
# est$infl.beta

# print the influences on the cumulative baseline hazard estimate
# est$infl.Lambda0.Tau1Tau2
```

```
# print the influences on the pure risk estimate
# est$infl.Pi.x.Tau1Tau2
```

`influences.CumBH` *influences.CumBH*

Description

Computes the influences on the log-relative hazard, baseline hazards at each unique event time, and on the cumulative baseline hazard in a given time interval [Tau1, Tau2]. Can take calibration of the design weights into account.

Usage

```
influences.CumBH(mod, Tau1 = NULL, Tau2 = NULL, A=NULL, calibrated = NULL)
```

Arguments

`mod`	a cox model object, result of function coxph.
`Tau1`	left bound of the time interval considered for the cumulative baseline hazard and pure risk. Default is the first event time.
`Tau2`	right bound of the time interval considered for the cumulative baseline hazard and pure risk. Default is the last event time.
`A`	$n \times q$ matrix with the values of the auxiliary variables used for the calibration of the weights in the whole cohort. Needs to be provided if `calibrated = TRUE`.
`calibrated`	are calibrated weights used for the estimation of the parameters? If `calibrated = TRUE`, the argument below needs to be provided. Default is `FALSE`.

Details

`influences.CumBH` works for estimation from a case-cohort with design weights or calibrated weights (case-cohort consisting of the subcohort and cases not in the subcohort, i.e., case-cohort obtained from two phases of sampling).

If covariate information is missing for certain individuals in the phase-two data (i.e., case-cohort obtained from three phases of sampling), use `influences.CumBH.missingdata`.

`influences.CumBH` uses the influence formulas provided in Etievant and Gail (2024).

If `calibrated = FALSE`, the infuences are only provided for the individuals in the case-cohort. If `calibrated = TRUE`, the influences are provided for all the individuals in the cohort.

Value

`infl.beta`: matrix with the overall influences on the log-relative hazard estimates.

`infl.lambda0.t`: matrix with the overall influences on the baseline hazards estimates at each unique event time.

`infl.Lambda0.Tau1Tau2.hat`: vector with the overall influences on the cumulative baseline hazard estimate in [Tau1, Tau2].

`infl2.beta`: matrix with the phase-two influences on the log-relative hazard estimates. Returned if `calibrated` = `TRUE`.

`infl2.lambda0.t`: matrix with the phase-two influences on the baseline hazards estimates at each unique event time. Returned if `calibrated` = `TRUE`.

`infl2.Lambda0.Tau1Tau2.hat`: vector with the phase-two influences on the cumulative baseline hazard estimate in [Tau1, Tau2]. Returned if `calibrated` = `TRUE`.

`beta.hat`: vector of length p with log-relative hazard estimates.

`lambda0.t.hat`: vector with baseline hazards estimates at each unique event time.

`Lambda0.Tau1Tau2.hat`: cumulative baseline hazard estimate in [Tau1, Tau2].

References

Etievant, L., Gail, M. H. (2024). Cox model inference for relative hazard and pure risk from stratified weight-calibrated case-cohort data. Lifetime Data Analysis, 30, 572-599.

See Also

`estimation`, `estimation.CumBH`, `estimation.PR`, `influences`, `influences.RH`, `influences.PR`, `influences.missingdata`, `influences.RH.missingdata`, `influences.CumBH.missingdata`, `influences.PR.missingdata`, `robustvariance` and `variance`.

Examples

```
data(dataexample.stratified, package="CaseCohortCoxSurvival")
cohort <- dataexample.stratified$cohort
casecohort <- cohort[which(cohort$status == 1 |
                     cohort$subcohort == 1),] # the stratified case-cohort
casecohort$weights <- casecohort$strata.n / casecohort$strata.m
casecohort$weights[which(casecohort$status == 1)] <- 1

Tau1 <- 0
Tau2 <- 8
x <- c(-1, 1, -0.6) # given covariate profile for the pure risk

# Estimation using the stratified case cohort with design weights
mod <- coxph(Surv(event.time, status) ~ X1 + X2 + X3, data = casecohort,
             weight = weights, id = id, robust = TRUE)
est <- influences(mod, Tau1 = Tau1, Tau2 = Tau2, x = x)

# print the influences on the cumulative baseline hazard estimate
# est$infl.Lambda0.Tau1Tau2
```

`influences.CumBH.missingdata`

influences.CumBH.missingdata

Description

Computes the influences on the log-relative hazard, baseline hazards at each unique event time, and on the cumulative baseline hazard in a given time interval [Tau1, Tau2], when covariate data is missing for certain individuals in the phase-two data.

Usage

```
influences.CumBH.missingdata(mod, riskmat.phase2, dNt.phase2 = NULL,
status.phase2 = NULL, Tau1 = NULL, Tau2 = NULL, estimated.weights = FALSE,
B.phase2 = NULL)
```

Arguments

`mod`	a cox model object, result of function coxph.
`riskmat.phase2`	at risk matrix for the phase-two data at all of the cases event times, even those with missing covariate data.
`dNt.phase2`	counting process matrix for failures in the phase-two data. Needs to be provided if `status.phase2 = NULL`.
`status.phase2`	vector indicating the case status in the phase-two data. Needs to be provided if `dNt.phase2 = NULL`.
`Tau1`	left bound of the time interval considered for the cumulative baseline hazard and pure risk. Default is the first event time.
`Tau2`	right bound of the time interval considered for the cumulative baseline hazard and pure risk. Default is the last event time.
`estimated.weights`	are the weights for the third phase of sampling (due to missingness) estimated? If `estimated.weights = TRUE`, the argument below needs to be provided. Default is `FALSE`.
`B.phase2`	matrix for the phase-two data, with phase-three sampling strata indicators. It should have as many columns as phase-three strata ($J^{(3)}$), with one 1 per row, to indicate the phase-three stratum position. Needs to be provided if `estimated.weights = TRUE`.

Details

`influences.CumBH.missingdata` works for estimation from a case-cohort with design weights and when covariate data was missing for certain individuals in the phase-two data (i.e., case-cohort obtained from three phases of sampling).

If there are no missing covariates in the phase-two sample, use `influences.CumBH` with either design weights or calibrated weights.

`influences.CumBH.missingdata` uses the influence formulas provided in Etievant and Gail (2024).

Value

`infl.beta`: matrix with the overall influences on the log-relative hazard estimates.

`infl.lambda0.t`: matrix with the overall influences on the baseline hazards estimates at each unique event time.

`infl.Lambda0.Tau1Tau2.hat`: vector with the overall influences on the cumulative baseline hazard estimate in [Tau1, Tau2].

`infl2.beta`: matrix with the phase-two influences on the log-relative hazard estimates.

`infl2.lambda0.t`: matrix with the phase-two influences on the baseline hazards estimates at each unique event time.

`infl2.Lambda0.Tau1Tau2.hat`: vector with the phase-two influences on the cumulative baseline hazard estimate in [Tau1, Tau2].

`infl3.beta`: matrix with the phase-three influences on the log-relative hazard estimates.

`infl3.lambda0.t`: matrix with the phase-three influences on the baseline hazards estimates at each unique event time.

`infl3.Lambda0.Tau1Tau2.hat`: vector with the phase-three influences on the cumulative baseline hazard estimate in [Tau1, Tau2].

`beta.hat`: vector of length p with log-relative hazard estimates.

`lambda0.t.hat`: vector with baseline hazards estimates at each unique event time.

`Lambda0.Tau1Tau2.hat`: cumulative baseline hazard estimate in [Tau1, Tau2].

References

Etievant, L., Gail, M. H. (2024). Cox model inference for relative hazard and pure risk from stratified weight-calibrated case-cohort data. Lifetime Data Analysis, 30, 572-599.

See Also

`estimation`, `estimation.CumBH`, `estimation.PR`, `influences.missingdata`, `influences.RH.missingdata`, `influences.PR.missingdata`, `influences`, `influences.RH`, `influences.CumBH`, `influences.PR`, `robustvariance` and `variance`.

Examples

```
data(dataexample.missingdata.stratified, package="CaseCohortCoxSurvival")

cohort <- dataexample.missingdata.stratified$cohort
phase2 <- cohort[which(cohort$phase2 == 1),] # the phase-two sample
casecohort <- cohort[which(cohort$phase3 == 1),] # the stratified case-cohort

B.phase2 <- cbind(1 * (phase2$W3 == 0), 1 * (phase2$W3 == 1))
rownames(B.phase2)  <- cohort[cohort$phase2 == 1, "id"]
B.phase3 <- cbind(1 * (casecohort$W3 == 0), 1 * (casecohort$W3 == 1))
rownames(B.phase3)  <- cohort[cohort$phase3 == 1, "id"]
total.B.phase2 <- colSums(B.phase2)
J3 <- ncol(B.phase3)
n <- nrow(cohort)
```

```
# Quantities needed for estimation of the cumulative baseline hazard when
# covariate data is missing
mod.cohort <- coxph(Surv(event.time, status) ~ X2, data = cohort,
                    robust = TRUE) # X2 is available on all cohort members
mod.cohort.detail <- coxph.detail(mod.cohort, riskmat = TRUE)

riskmat.phase2 <- with(cohort, mod.cohort.detail$riskmat[phase2 == 1,])
rownames(riskmat.phase2) <- cohort[cohort$phase2 == 1, "id"]
observed.times.phase2 <- apply(riskmat.phase2, 1,
                               function(v) {which.max(cumsum(v))})
dNt.phase2 <- matrix(0, nrow(riskmat.phase2), ncol(riskmat.phase2))
dNt.phase2[cbind(1:nrow(riskmat.phase2), observed.times.phase2)] <- 1
dNt.phase2 <- sweep(dNt.phase2, 1, phase2$status, "*")
colnames(dNt.phase2) <- colnames(riskmat.phase2)
rownames(dNt.phase2) <- rownames(riskmat.phase2)

Tau1 <- 0 # given time interval for the pure risk
Tau2 <- 8
x <- c(-1, 1, -0.6) # given covariate profile for the pure risk

# Estimation using the stratified case cohort with true known design weights
mod.true <- coxph(Surv(event.time, status) ~ X1 + X2 + X3, data = casecohort,
                  weight = weight.true, id = id, robust = TRUE)

est.true <- influences.missingdata(mod = mod.true, riskmat.phase2 = riskmat.phase2,
                                   dNt.phase2 = dNt.phase2, Tau1 = Tau1,
                                   Tau2 = Tau2, x = x)

# print the influences on the cumulative baseline hazard estimate
# est.true$infl.Lambda0.Tau1Tau2
# print the phase-two influences on the cumulative baseline hazard estimate
# est.true$infl2.Lambda0.Tau1Tau2
# print the phase-three influences on the cumulative baseline hazard estimate
# est.true$infl3.Lambda0.Tau1Tau2

# Estimation using the stratified case cohort with estimated weights, and
# accounting for the estimation through the influences
mod.estimated <- coxph(Surv(event.time, status) ~ X1 + X2 + X3,
                       data = casecohort, weight = weight.est, id = id,
                       robust = TRUE)

est.estimated  <- influences.missingdata(mod.estimated,
                                         riskmat.phase2 = riskmat.phase2,
                                         dNt.phase2 = dNt.phase2,
                                         estimated.weights = TRUE,
                                         B.phase2 = B.phase2, Tau1 = Tau1,
                                         Tau2 = Tau2, x = x)

# print the influences on the cumulative baseline hazard estimate
# est.estimated$infl.Lambda0.Tau1Tau2
# print the phase-two influences on the cumulative baseline hazard estimate
# est.estimated$infl2.Lambda0.Tau1Tau2
```

```
# print the phase-three influences on the cumulative baseline hazard estimate
# est.estimated$infl3.Lambda0.Tau1Tau2
```

`influences.missingdata` *influences.missingdata*

Description

Computes the influences on the log-relative hazard, baseline hazards at each unique event time, cumulative baseline hazard in a given time interval [Tau1, Tau2] and on the pure risk in [Tau1, Tau2] and for a given covariate profile x, when covariate data is missing for certain individuals in the phase-two data.

Usage

```
influences.missingdata(mod, riskmat.phase2, dNt.phase2 = NULL,
status.phase2 = NULL, Tau1 = NULL, Tau2 = NULL, x = NULL,
estimated.weights = FALSE, B.phase2 = NULL)
```

Arguments

`mod`	a cox model object, result of function coxph.
`riskmat.phase2`	at risk matrix for the phase-two data at all of the cases event times, even those with missing covariate data.
`dNt.phase2`	counting process matrix for failures in the phase-two data. Needs to be provided if `status.phase2 = NULL`.
`status.phase2`	vector indicating the case status in the phase-two data. Needs to be provided if `dNt.phase2 = NULL`.
`Tau1`	left bound of the time interval considered for the cumulative baseline hazard and pure risk. Default is the first event time.
`Tau2`	right bound of the time interval considered for the cumulative baseline hazard and pure risk. Default is the last event time.
`x`	vector of length p, specifying the covariate profile considered for the pure risk. Default is (0,...,0).
`estimated.weights`	are the weights for the third phase of sampling (due to missingness) estimated? If `estimated.weights = TRUE`, the argument below needs to beprovided. Default is `FALSE`.
`B.phase2`	matrix for the phase-two data, with phase-three sampling strata indicators. It should have as many columns as phase-three strata ($J^{(3)}$), with one 1 per row, to indicate the phase-three stratum position. Needs to be provided if `estimated.weights = TRUE`.

Details

`influences.missingdata` works for estimation from a case-cohort with design weights and when covariate data was missing for certain individuals in the phase-two data (i.e., case-cohort obtained from three phases of sampling).

If there are no missing covariates in the phase- two sample, use `influences` with either design weights or calibrated weights.

When covariate information was missing for certain individuals in the phase-two data (i.e., case-cohort obtained from three phases of sampling), use `influences.missingdata`.

`influences.missingdata` uses the influence formulas provided in Etievant and Gail (2024).

Value

`infl.beta`: matrix with the overall influences on the log-relative hazard estimates.

`infl.lambda0.t`: matrix with the overall influences on the baseline hazards estimates at each unique event time.

`infl.Lambda0.Tau1Tau2.hat`: vector with the overall influences on the cumulative baseline hazard estimate in [Tau1, Tau2].

`infl.Pi.x.Tau1Tau2.hat`: vector with the overall influences on the pure risk estimate in [Tau1, Tau2] and for covariate profile `x`.

`infl2.beta`: matrix with the phase-two influences on the log-relative hazard estimates.

`infl2.lambda0.t`: matrix with the phase-two influences on the baseline hazards estimates at each unique event time.

`infl2.Lambda0.Tau1Tau2.hat`: vector with the phase-two influences on the cumulative baseline hazard estimate in [Tau1, Tau2].

`infl2.Pi.x.Tau1Tau2.hat`: vector with the phase-two influences on the pure risk estimate in [Tau1, Tau2] and for covariate profile `x`.

`infl3.beta`: matrix with the phase-three influences on the log-relative hazard estimates.

`infl3.lambda0.t`: matrix with the phase-three influences on the baseline hazards estimates at each unique event time.

`infl3.Lambda0.Tau1Tau2.hat`: vector with the phase-three influences on the cumulative baseline hazard estimate in [Tau1, Tau2].

`infl3.Pi.x.Tau1Tau2.hat`: vector with the phase-three influences on the pure risk estimate in [Tau1, Tau2] and for covariate profile `x`.

`beta.hat`: vector of length p with log-relative hazard estimates.

`lambda0.t.hat`: vector with baseline hazards estimates at each unique event time.

`Lambda0.Tau1Tau2.hat`: cumulative baseline hazard estimate in [Tau1, Tau2].

`Pi.x.Tau1Tau2.hat`: pure risk estimate in [Tau1, Tau2] and for covariate profile `x`.

References

Etievant, L., Gail, M. H. (2024). Cox model inference for relative hazard and pure risk from stratified weight-calibrated case-cohort data. Lifetime Data Analysis, 30, 572-599.

See Also

`estimation`, `estimation.CumBH`, `estimation.PR`, `influences.RH.missingdata`, `influences.CumBH.missingdata`, `influences.PR.missingdata`, `influences`, `influences.RH`, `influences.CumBH`, `influences.PR`, `robustvariance` and `variance`.

Examples

```
data(dataexample.missingdata.stratified, package="CaseCohortCoxSurvival")

cohort <- dataexample.missingdata.stratified$cohort
phase2 <- cohort[which(cohort$phase2 == 1),] # the phase-two sample
casecohort <- cohort[which(cohort$phase3 == 1),] # the stratified case-cohort

B.phase2 <- cbind(1 * (phase2$W3 == 0), 1 * (phase2$W3 == 1))
rownames(B.phase2)  <- cohort[cohort$phase2 == 1, "id"]
B.phase3 <- cbind(1 * (casecohort$W3 == 0), 1 * (casecohort$W3 == 1))
rownames(B.phase3)  <- cohort[cohort$phase3 == 1, "id"]
total.B.phase2 <- colSums(B.phase2)
J3 <- ncol(B.phase3)
n <- nrow(cohort)

# Quantities needed for estimation of the cumulative baseline hazard when
# covariate data is missing
mod.cohort <- coxph(Surv(event.time, status) ~ X2, data = cohort,
                    robust = TRUE) # X2 is available on all cohort members
mod.cohort.detail <- coxph.detail(mod.cohort, riskmat = TRUE)

riskmat.phase2 <- with(cohort, mod.cohort.detail$riskmat[phase2 == 1,])
rownames(riskmat.phase2) <- cohort[cohort$phase2 == 1, "id"]
observed.times.phase2 <- apply(riskmat.phase2, 1,
                               function(v) {which.max(cumsum(v))})
dNt.phase2 <- matrix(0, nrow(riskmat.phase2), ncol(riskmat.phase2))
dNt.phase2[cbind(1:nrow(riskmat.phase2), observed.times.phase2)] <- 1
dNt.phase2 <- sweep(dNt.phase2, 1, phase2$status, "*")
colnames(dNt.phase2) <- colnames(riskmat.phase2)
rownames(dNt.phase2) <- rownames(riskmat.phase2)

Tau1 <- 0 # given time interval for the pure risk
Tau2 <- 8
x <- c(-1, 1, -0.6) # given covariate profile for the pure risk

# Estimation using the stratified case cohort with true known design weights
mod.true <- coxph(Surv(event.time, status) ~ X1 + X2 + X3, data = casecohort,
                  weight = weight.true, id = id, robust = TRUE)

est.true <- influences.missingdata(mod = mod.true,
                                   riskmat.phase2 = riskmat.phase2,
                                   dNt.phase2 = dNt.phase2, Tau1 = Tau1,
                                   Tau2 = Tau2, x = x)

# print the influences on the log-relative hazard estimates
# est.true$infl.beta
```

```
# print the phase-two influences on the log-relative hazard estimates
# est.true$infl2.beta
# print the phase-three influences on the log-relative hazard estimates
# est.true$infl3.beta

# print the influences on the cumulative baseline hazard estimate
# est.true$infl.Lambda0.Tau1Tau2
# print the phase-two influences on the cumulative baseline hazard estimate
# est.true$infl2.Lambda0.Tau1Tau2
# print the phase-three influences on the cumulative baseline hazard estimate
# est.true$infl3.Lambda0.Tau1Tau2

# print the influences on the pure risk estimate
# est.true$infl.Pi.x.Tau1Tau2
# print the phase-two influences on the pure risk estimate
# est.true$infl2.Pi.x.Tau1Tau2
# print the phase-three influences on the pure risk estimate
# est.true$infl3.Pi.x.Tau1Tau2

# Estimation using the stratified case cohort with estimated weights, and
# accounting for the estimation through the influences
mod.estimated <- coxph(Surv(event.time, status) ~ X1 + X2 + X3,
                       data = casecohort, weight = weight.est, id = id,
                       robust = TRUE)

est.estimated  <- influences.missingdata(mod.estimated,
                                         riskmat.phase2 = riskmat.phase2,
                                         dNt.phase2 = dNt.phase2,
                                         estimated.weights = TRUE,
                                         B.phase2 = B.phase2, Tau1 = Tau1,
                                         Tau2 = Tau2, x = x)

# print the influences on the log-relative hazard estimates
# est.estimated$infl.beta
# print the phase-two influences on the log-relative hazard estimates
# est.estimated$infl2.beta
# print the phase-three influences on the log-relative hazard estimates
# est.estimated$infl3.beta

# print the influences on the cumulative baseline hazard estimate
# est.estimated$infl.Lambda0.Tau1Tau2
# print the phase-two influences on the cumulative baseline hazard estimate
# est.estimated$infl2.Lambda0.Tau1Tau2
# print the phase-three influences on the cumulative baseline hazard estimate
# est.estimated$infl3.Lambda0.Tau1Tau2

# print the influences on the pure risk estimate
# est.estimated$infl.Pi.x.Tau1Tau2
# print the phase-two influences on the pure risk estimate
# est.estimated$infl2.Pi.x.Tau1Tau2
# print the phase-three influences on the pure risk estimate
# est.estimated$infl3.Pi.x.Tau1Tau2
```

`influences.PR` *influences.PR*

Description

Computes the influences on the pure risk in the time interval [Tau1, Tau2] and for a given covariate profile `x`, from that on the log-relative hazard and cumulative baseline hazard. Can take calibration of the design weights into account.

Usage

```
influences.PR(beta, Lambda0.Tau1Tau2, x = NULL, infl.beta,
infl.Lambda0.Tau1Tau2, calibrated = NULL, infl2.beta = NULL,
infl2.Lambda0.Tau1Tau2 = NULL)
```

Arguments

`beta`	vector of length p with log-relative hazard values.
`Lambda0.Tau1Tau2`	cumulative baseline hazard in [Tau1, Tau2].
`x`	vector of length p, specifying the covariate profile considered for the pure risk. Default is (0,...,0).
`infl.beta`	matrix with the overall influences on the log-relative hazard estimates.
`infl.Lambda0.Tau1Tau2`	vector with the overall influences on the cumulative baseline hazard estimate in [Tau1, Tau2].
`calibrated`	are calibrated weights used for the estimation of the parameters? If `calibrated = TRUE`, the arguments below need to be provided. Default is `FALSE`.
`infl2.beta`	matrix with the phase-two influences on the log-relative hazard estimates. Needs to be provided if `missing.data = TRUE`.
`infl2.Lambda0.Tau1Tau2`	vector with the phase-two influences on the cumulative baseline hazard estimate in [Tau1, Tau2]. Needs to be provided if `missing.data = TRUE`.

Details

`influences.PR` works for estimation from a case-cohort with design weights or calibrated weights (case-cohort consisting of the subcohort and cases not in the subcohort, i.e., case-cohort obtained from two phases of sampling).

If covariate information is missing for certain individuals in the phase-two data (i.e., case-cohort obtained from three phases of sampling), use `influences.PR.missingdata`.

`influences` uses the influence formulas provided in Etievant and Gail (2024).

If `calibrated = FALSE`, the infuences are only provided for the individuals in the case-cohort. If `calibrated = TRUE`, the influences are provided for all the individuals in the cohort.

Value

`infl.Pi.x.Tau1Tau2.hat`: vector with the overall influences on the pure risk estimate in [Tau1, Tau2] and for covariate profile `x`.

`infl2.Pi.x.Tau1Tau2.hat`: vector with the phase-two influences on the pure risk estimate in [Tau1, Tau2] and for covariate profile `x`. Returned if `calibrated` = `TRUE`.

`Pi.x.Tau1Tau2.hat`: pure risk estimate in [Tau1, Tau2] and for covariate profile `x`.

References

Etievant, L., Gail, M. H. (2024). Cox model inference for relative hazard and pure risk from stratified weight-calibrated case-cohort data. Lifetime Data Analysis, 30, 572-599.

See Also

`estimation`, `estimation.CumBH`, `estimation.PR`, `influences`, `influences.RH`, `influences.CumBH`, `influences.missingdata`, `influences.RH.missingdata`, `influences.CumBH.missingdata`, `influences.PR.missing`
`robustvariance` and `variance`.

Examples

```
data(dataexample.stratified, package="CaseCohortCoxSurvival")
cohort <- dataexample.stratified$cohort
casecohort <- cohort[which(cohort$status == 1 |
                     cohort$subcohort == 1),] # the stratified case-cohort
casecohort$weights <- casecohort$strata.n / casecohort$strata.m
casecohort$weights[which(casecohort$status == 1)] <- 1

Tau1 <- 0
Tau2 <- 8
x <- c(-1, 1, -0.6) # given covariate profile for the pure risk

# Estimation using the stratified case cohort with design weights
mod <- coxph(Surv(event.time, status) ~ X1 + X2 + X3, data = casecohort,
             weight = weights, id = id, robust = TRUE)
est <- influences(mod, Tau1 = Tau1, Tau2 = Tau2, x = x)

# print the influences on the pure risk estimate
# est$infl.Pi.x.Tau1Tau2
```

`influences.PR.missingdata`

influences.PR.missingdata

Description

Computes the influences on the pure risk in the time interval [Tau1, Tau2] and for a given covariate profile x, from that on the log-relative hazard and cumulative baseline hazard, when covariate data is missing for certain individuals in the phase-two data.

Usage

```
influences.PR.missingdata(beta, Lambda0.Tau1Tau2, x = NULL, infl2.beta,
infl2.Lambda0.Tau1Tau2, infl3.beta, infl3.Lambda0.Tau1Tau2)
```

Arguments

`beta`	vector of length p with log-relative hazard values.
`Lambda0.Tau1Tau2`	cumulative baseline hazard in [Tau1, Tau2].
`x`	vector of length p, specifying the covariate profile considered for the pure risk. Default is (0,...,0).
`infl2.beta`	matrix with the overall influences on the log-relative hazard estimates.
`infl2.Lambda0.Tau1Tau2`	vector with the overall influences on the cumulative baseline hazard estimate in [Tau1, Tau2].
`infl3.beta`	matrix with the phase-three influences on the log-relative hazard estimates.
`infl3.Lambda0.Tau1Tau2`	vector with the phase-three influences on the cumulative baseline hazard estimate in [Tau1, Tau2].

Details

`influences.PR.missingdata` works for estimation from a case-cohort with design weights and when covariate data was missing for certain individuals in the phase-two data (i.e., case-cohort obtained from three phases of sampling).

If there are no missing covariates in the phase- two sample, use `influences.PR` with either design weights or calibrated weights.

`influences.PR.missingdata` uses the influence formulas provided in Etievant and Gail (2024).

Value

`infl.Pi.x.Tau1Tau2.hat`: vector with the overall influences on the pure risk estimate in [Tau1, Tau2] and for covariate profile `x`.

`infl2.Pi.x.Tau1Tau2.hat`: vector with the phase-two influences on the pure risk estimate in [Tau1, Tau2] and for covariate profile `x`.

`infl3.Pi.x.Tau1Tau2.hat`: vector with the phase-three influences on the pure risk estimate in [Tau1, Tau2] and for covariate profile `x`.

`Pi.x.Tau1Tau2.hat`: pure risk estimate in [Tau1, Tau2] and for covariate profile `x`.

References

Etievant, L., Gail, M. H. (2024). Cox model inference for relative hazard and pure risk from stratified weight-calibrated case-cohort data. Lifetime Data Analysis, 30, 572-599.

See Also

estimation, estimation.CumBH, estimation.PR, influences.missingdata, influences.RH.missingdata, influences.CumBH.missingdata, influences, influences.RH, influences.CumBH, influences.PR, robustvariance and variance.

Examples

```
data(dataexample.missingdata.stratified, package="CaseCohortCoxSurvival")

cohort <- dataexample.missingdata.stratified$cohort
phase2 <- cohort[which(cohort$phase2 == 1),] # the phase-two sample
casecohort <- cohort[which(cohort$phase3 == 1),] # the stratified case-cohort

B.phase2 <- cbind(1 * (phase2$W3 == 0), 1 * (phase2$W3 == 1))
rownames(B.phase2)  <- cohort[cohort$phase2 == 1, "id"]
B.phase3 <- cbind(1 * (casecohort$W3 == 0), 1 * (casecohort$W3 == 1))
rownames(B.phase3)  <- cohort[cohort$phase3 == 1, "id"]
total.B.phase2 <- colSums(B.phase2)
J3 <- ncol(B.phase3)
n <- nrow(cohort)

# Quantities needed for estimation of the cumulative baseline hazard when
# covariate data is missing
mod.cohort <- coxph(Surv(event.time, status) ~ X2, data = cohort,
                    robust = TRUE) # X2 is available on all cohort members
mod.cohort.detail <- coxph.detail(mod.cohort, riskmat = TRUE)

riskmat.phase2 <- with(cohort, mod.cohort.detail$riskmat[phase2 == 1,])
rownames(riskmat.phase2) <- cohort[cohort$phase2 == 1, "id"]
observed.times.phase2 <- apply(riskmat.phase2, 1,
                               function(v) {which.max(cumsum(v))})
dNt.phase2 <- matrix(0, nrow(riskmat.phase2), ncol(riskmat.phase2))
dNt.phase2[cbind(1:nrow(riskmat.phase2), observed.times.phase2)] <- 1
dNt.phase2 <- sweep(dNt.phase2, 1, phase2$status, "*")
colnames(dNt.phase2) <- colnames(riskmat.phase2)
rownames(dNt.phase2) <- rownames(riskmat.phase2)

Tau1 <- 0 # given time interval for the pure risk
Tau2 <- 8
x <- c(-1, 1, -0.6) # given covariate profile for the pure risk
v <- c(1, -1, 0.6) # over covariate profile

# Estimation using the stratified case cohort with true known design weights
mod.true <- coxph(Surv(event.time, status) ~ X1 + X2 + X3, data = casecohort,
                  weight = weight.true, id = id, robust = TRUE)

est.true <- influences.missingdata(mod = mod.true,
                                   riskmat.phase2 = riskmat.phase2,
                                   dNt.phase2 = dNt.phase2, Tau1 = Tau1,
                                   Tau2 = Tau2, x = x)

beta.true <- est.true$beta.hat
```

```
Lambda0.true <- est.true$Lambda0.Tau1Tau2.hat
infl2.beta.true <- est.true$infl2.beta
infl2.Lambda0.true <- est.true$infl2.Lambda0.Tau1Tau2
infl3.beta.true <- est.true$infl3.beta
infl3.Lambda0.true <- est.true$infl3.Lambda0.Tau1Tau2

est.PR2.true <- influences.PR.missingdata(beta = beta.true,
                                          Lambda0.Tau1Tau2 = Lambda0.true,
                                          x = v,
                                          infl2.beta = infl2.beta.true,
                                          infl2.Lambda0.Tau1Tau2 = infl2.Lambda0.true,
                                          infl3.beta = infl3.beta.true,
                                          infl3.Lambda0.Tau1Tau2 = infl3.Lambda0.true)

# print the influences on the pure risk estimate
# est.PR2.true$infl.Pi.x.Tau1Tau2
# print the phase-two influences on the pure risk estimate
# est.PR2.true$infl2.Pi.x.Tau1Tau2
# print the phase-three influences on the pure risk estimate
# est.PR2.true$infl3.Pi.x.Tau1Tau2

# Estimation using the stratified case cohort with estimated weights, and
# accounting for the estimation through the influences
mod.estimated <- coxph(Surv(event.time, status) ~ X1 + X2 + X3,
                       data = casecohort, weight = weight.est, id = id,
                       robust = TRUE)

est.estimated  <- influences.missingdata(mod.estimated,
                                         riskmat.phase2 = riskmat.phase2,
                                         dNt.phase2 = dNt.phase2,
                                         estimated.weights = TRUE,
                                         B.phase2 = B.phase2, Tau1 = Tau1,
                                         Tau2 = Tau2, x = x)

beta.estimated <- est.estimated$beta.hat
Lambda0.estimated <- est.estimated$Lambda0.Tau1Tau2.hat
infl2.beta.estimated <- est.estimated$infl2.beta
infl2.Lambda0.estimated <- est.estimated$infl2.Lambda0.Tau1Tau2
infl3.beta.estimated <- est.estimated$infl3.beta
infl3.Lambda0.estimated <- est.estimated$infl3.Lambda0.Tau1Tau2

est.PR2.estimated <- influences.PR.missingdata(beta = beta.estimated,
                                               Lambda0.Tau1Tau2 = Lambda0.estimated,
                                               x = v,
                                               infl2.beta = infl2.beta.estimated,
                                      infl2.Lambda0.Tau1Tau2 = infl2.Lambda0.estimated,
                                               infl3.beta = infl3.beta.estimated,
                                      infl3.Lambda0.Tau1Tau2 = infl3.Lambda0.estimated)

# print the influences on the pure risk estimate
# est.PR2.estimated$infl.Pi.x.Tau1Tau2
# print the phase-two influences on the pure risk estimate
# est.PR2.estimated$infl2.Pi.x.Tau1Tau2
```

```
# print the phase-three influences on the pure risk estimate
# est.PR2.estimated$infl3.Pi.x.Tau1Tau2
```

`influences.RH` *influences.RH*

Description

Computes the influences on the log-relative hazard. Can take calibration of the design weights into account.

Usage

```
influences.RH(mod, calibrated = NULL, A = NULL)
```

Arguments

`mod`	a cox model object, result of function coxph.
`calibrated`	are calibrated weights used for the estimation of the parameters? If `calibrated = TRUE`, the argument below needs to be provided. Default is `FALSE`.
`A`	$n \times q$ matrix with the values of the auxiliary variables used for the calibration of the weights in the whole cohort. Needs to be provided if `calibrated = TRUE`.

Details

`influences.RH` works for estimation from a case-cohort with design weights or calibrated weights (case-cohort consisting of the subcohort and cases not in the subcohort, i.e., case-cohort obtained from two phases of sampling).

If covariate information is missing for certain individuals in the phase-two data (i.e., case-cohort obtained from three phases of sampling), use `influences.RH.missingdata`.

`influence.RH` uses the influence formulas provided in Etievant and Gail (2024).

If `calibrated = FALSE`, the infuences are only provided for the individuals in the case-cohort. If `calibrated = TRUE`, the influences are provided for all the individuals in the cohort.

Value

`infl.beta`: matrix with the overall influences on the log-relative hazard estimates.

`infl2.beta`: matrix with the phase-two influences on the log-relative hazard estimates. Returned if `calibrated = TRUE`.

`beta.hat`: vector of length p with log-relative hazard estimates.

References

Etievant, L., Gail, M. H. (2024). Cox model inference for relative hazard and pure risk from stratified weight-calibrated case-cohort data. Lifetime Data Analysis, 30, 572-599.

See Also

`estimation`, `estimation.CumBH`, `estimation.PR`, `influences`, `influences.CumBH`, `influences.PR`, `influences.missingdata`, `influences.RH.missingdata`, `influences.CumBH.missingdata`, `influences.PR.missingdata`, `robustvariance` and `variance`.

Examples

```
data(dataexample.stratified, package="CaseCohortCoxSurvival")
cohort <- dataexample.stratified$cohort
casecohort <- cohort[which(cohort$status == 1 |
                     cohort$subcohort == 1),] # the stratified case-cohort
casecohort$weights <- casecohort$strata.n / casecohort$strata.m
casecohort$weights[which(casecohort$status == 1)] <- 1

Tau1 <- 0
Tau2 <- 8
x <- c(-1, 1, -0.6) # given covariate profile for the pure risk

# Estimation using the stratified case cohort with design weights
mod <- coxph(Surv(event.time, status) ~ X1 + X2 + X3, data = casecohort,
             weight = weights, id = id, robust = TRUE)
est <- influences(mod, Tau1 = Tau1, Tau2 = Tau2, x = x)

# print the influences on the log-relative hazard estimates
# est$infl.beta
```

`influences.RH.missingdata`

influences.RH.missingdata

Description

Computes the influences on the log-relative hazard, when covariate data is missing for certain individuals in the phase-two data.

Usage

```
influences.RH.missingdata(mod, riskmat.phase2, dNt.phase2 = NULL,
status.phase2 = NULL, estimated.weights = FALSE, B.phase2 = NULL)
```

Arguments

`mod`	a cox model object, result of function coxph.
`riskmat.phase2`	at risk matrix for the phase-two data at all of the cases event times, even those with missing covariate data.
`dNt.phase2`	counting process matrix for failures in the phase-two data. Needs to be provided if `status.phase2` = `NULL`.

`status.phase2`	vector indicating the case status in the phase-two data. Needs to be provided if `dNt.phase2 = NULL`.
`estimated.weights`	are the weights for the third phase of sampling (due to missingness) estimated? If `estimated.weights = TRUE`, the argument below needs to beprovided. Default is `FALSE`.
`B.phase2`	matrix for the phase-two data, with phase-three sampling strata indicators. It should have as many columns as phase-three strata ($J^{(3)}$), with one 1 per row, to indicate the phase-three stratum position. Needs to be provided if `estimated.weights = TRUE`.

Details

`influences.RH.missingdata` works for estimation from a case-cohort with design weights and when covariate data was missing for certain individuals in the phase-two data (i.e., case-cohort obtained from three phases of sampling and consisting of individuals in the phase-two data without missing covariate information).

If there are no missing covariates in the phase- two sample, use `influences.RH` with either design weights or calibrated weights.

`influences.RH.missingdata` uses the influence formulas provided in Etievant and Gail (2024).

Value

`infl.beta`: matrix with the overall influences on the log-relative hazard estimates.

`infl2.beta`: matrix with the phase-two influences on the log-relative hazard estimates.

`infl3.beta`: matrix with the phase-three influences on the log-relative hazard estimates.

`beta.hat`: vector of length p with log-relative hazard estimates.

References

Etievant, L., Gail, M. H. (2024). Cox model inference for relative hazard and pure risk from stratified weight-calibrated case-cohort data. Lifetime Data Analysis, 30, 572-599.

See Also

`estimation`, `estimation.CumBH`, `estimation.PR`, `influences.missingdata`, `influences.CumBH.missingdata`, `influences.PR.missingdata`, `influences`, `influences.RH`, `influences.CumBH`, `influences.PR`, `robustvariance` and `variance`.

Examples

```
data(dataexample.missingdata.stratified, package="CaseCohortCoxSurvival")

cohort <- dataexample.missingdata.stratified$cohort
phase2 <- cohort[which(cohort$phase2 == 1),] # the phase-two sample
casecohort <- cohort[which(cohort$phase3 == 1),] # the stratified case-cohort

B.phase2 <- cbind(1 * (phase2$W3 == 0), 1 * (phase2$W3 == 1))
```

```
rownames(B.phase2)  <- cohort[cohort$phase2 == 1, "id"]
B.phase3 <- cbind(1 * (casecohort$W3 == 0), 1 * (casecohort$W3 == 1))
rownames(B.phase3)  <- cohort[cohort$phase3 == 1, "id"]
total.B.phase2 <- colSums(B.phase2)
J3 <- ncol(B.phase3)
n <- nrow(cohort)

# Quantities needed for estimation of the cumulative baseline hazard when
# covariate data is missing
mod.cohort <- coxph(Surv(event.time, status) ~ X2, data = cohort,
                    robust = TRUE) # X2 is available on all cohort members
mod.cohort.detail <- coxph.detail(mod.cohort, riskmat = TRUE)

riskmat.phase2 <- with(cohort, mod.cohort.detail$riskmat[phase2 == 1,])
rownames(riskmat.phase2) <- cohort[cohort$phase2 == 1, "id"]
observed.times.phase2 <- apply(riskmat.phase2, 1,
                               function(v) {which.max(cumsum(v))})
dNt.phase2 <- matrix(0, nrow(riskmat.phase2), ncol(riskmat.phase2))
dNt.phase2[cbind(1:nrow(riskmat.phase2), observed.times.phase2)] <- 1
dNt.phase2 <- sweep(dNt.phase2, 1, phase2$status, "*")
colnames(dNt.phase2) <- colnames(riskmat.phase2)
rownames(dNt.phase2) <- rownames(riskmat.phase2)

Tau1 <- 0 # given time interval for the pure risk
Tau2 <- 8
x <- c(-1, 1, -0.6) # given covariate profile for the pure risk

# Estimation using the stratified case cohort with true known design weights
mod.true <- coxph(Surv(event.time, status) ~ X1 + X2 + X3, data = casecohort,
                  weight = weight.true, id = id, robust = TRUE)

est.true <- influences.missingdata(mod = mod.true, riskmat.phase2 = riskmat.phase2,
                                   dNt.phase2 = dNt.phase2, Tau1 = Tau1,
                                   Tau2 = Tau2, x = x)

# print the influences on the log-relative hazard estimates
# est.true$infl.beta
# print the phase-two influences on the log-relative hazard estimates
# est.true$infl2.beta
# print the phase-three influences on the log-relative hazard estimates
# est.true$infl3.beta

# Estimation using the stratified case cohort with estimated weights, and
# accounting for the estimation through the influences
mod.estimated <- coxph(Surv(event.time, status) ~ X1 + X2 + X3,
                       data = casecohort, weight = weight.est, id = id,
                       robust = TRUE)

est.estimated  <- influences.missingdata(mod.estimated,
                                         riskmat.phase2 = riskmat.phase2,
                                         dNt.phase2 = dNt.phase2,
                                         estimated.weights = TRUE,
                                         B.phase2 = B.phase2, Tau1 = Tau1,
```

```
                                           Tau2 = Tau2, x = x)

# print the influences on the log-relative hazard estimates
# est.estimated$infl.beta
# print the phase-two influences on the log-relative hazard estimates
# est.estimated$infl2.beta
# print the phase-three influences on the log-relative hazard estimates
# est.estimated$infl3.beta
```

`product.covar.weight` *product.covar.weight*

Description

Computes the product of joint design weights and joint sampling indicators covariances, needed for the phase-two component of the variance (with design or calibrated weights).

Usage

```
product.covar.weight(casecohort, stratified = NULL)
```

Arguments

`casecohort`	if `stratified = TRUE`, data frame with `status` (case status), `W` (the J strata), `strata.m` (vector of length J with the numbers of sampled individuals in the strata) and `strata.n` (vector of length J with the strata sizes), for each individual in the stratified case-cohort data. If `stratified = FALSE`, data frame with `status` (case status), `m` (number of sampled individuals) and `n` (cohort size), for each individual in the un-stratified case-cohort data.
`stratified`	was the sampling of the case-cohort stratified on `W`? Default is `FALSE`.

Details

`product.covar.weight` creates the matrix with the products of joint design weights and joint sampling indicator covariances, for the non-cases in the case cohort. In other words, it has as many rows and columns as non-cases in the case cohort, and contains the $w_{i,k,j}\sigma_{i,k,j}$, with

$w_{i,k,j} = \frac{n^{(j)}(n^{(j)}-1)}{m^{(j)}(m^{(j)}-1)}$ if individuals i and k in stratum j are both non-cases, and $w_{i,k,j} = \left(\frac{n^{(j)}}{m^{(j)}}\right)^2$ otherwise, $i \neq k \in \{1, \ldots, n^{(j)}\}$, $j \in \{1, \ldots, J\}$.

$w_{i,i,j} = \frac{n^{(j)}}{m^{(j)}}$ if individuals i in stratum j is a non-case, $i \in \{1, \ldots, n^{(j)}\}$, $j \in \{1, \ldots, J\}$.

$\sigma_{i,k,j} = \frac{m^{(j)}(m^{(j)}-1)}{n^{(j)}(n^{(j)}-1)} - \left(\frac{m^{(j)}}{n^{(j)}}\right)^2$ if individuals i and k in stratum j are both non-cases, $i \neq k \in \{1, \ldots, n^{(j)}\}$, $j \in \{1, \ldots, J\}$.

$\sigma_{i,i,j} = \frac{m^{(j)}}{n^{(j)}} - \left(1 - \frac{m^{(j)}}{n^{(j)}}\right)$ if individuals i in stratum j is a non-case, $i \in \{1, \ldots, n^{(j)}\}$, $j \in \{1, \ldots, J\}$.

See Etievant and Gail (2024).

Value

`product.covar.weight`: matrix with the products of joint design weights and joint sampling indicator covariances, for the non-cases in the case-cohort.

References

Etievant, L., Gail, M. H. (2024). Cox model inference for relative hazard and pure risk from stratified weight-calibrated case-cohort data. Lifetime Data Analysis, 30, 572-599.

See Also

`variance`, that uses `product.covar.weight` to compute the variance estimate that follows the complete variance decomposition (superpopulation and phase-two variance components).

Examples

```
data(dataexample.stratified, package="CaseCohortCoxSurvival")
cohort <- dataexample.stratified$cohort
casecohort <- cohort[which(cohort$status == 1 |
                     cohort$subcohort == 1),] # the stratified case-cohort

prod.covar.weight <- product.covar.weight(casecohort, stratified = TRUE)

sum(casecohort$status == 0) # number of non-cases in the case-cohort
```

`robustvariance` *robustvariance*

Description

Computes the robust variance estimate, i.e., the sum of the squared influence functions, for a parameter such as log-relative hazard, cumulative baseline hazard or covariate specific pure-risk.

Usage

```
robustvariance(infl)
```

Arguments

`infl`	overall influences on a parameter such as log-relative hazard, cumulative baseline hazard or covariate specific pure-risk.

Details

`robustvariance` works for estimation with design or calibrated weights from a case cohort obtained from two phases of sampling (i.e., case cohort consisting of the subcohort and cases not in the subcohort), or when covariate information was missing for certain individuals in the phase-two data (i.e., case cohort obtained from three phases of sampling and consisting of individuals in the phase-two data without missing covariate information).

Value

`robust.var`: robust variance estimate.

References

Barlow W. (1994). Robust Variance Estimation for the Case-Cohort Design. Biometrics, 50, 1064-1072.

Langholz B., Jiao J. (2007). Computational methods for case-cohort studies. Computational Statistics & Data Analysis, 51, 3737-37.

Etievant, L., Gail, M. H. (2024). Cox model inference for relative hazard and pure risk from stratified weight-calibrated case-cohort data. Lifetime Data Analysis, 30, 572-599.

See Also

`influences.RH`, `influences.CumBH`, `influences.PR`, `influences.missingdata`, `influences.RH.missingdata`, `influences.CumBH.missingdata`, `influences.PR.missingdata` and `variance`.

Examples

```
data(dataexample.stratified, package="CaseCohortCoxSurvival")
cohort <- dataexample.stratified$cohort
casecohort <- cohort[which(cohort$status == 1 |
                     cohort$subcohort == 1),] # the stratified case-cohort
casecohort$weights <- casecohort$strata.n / casecohort$strata.m
casecohort$weights[which(casecohort$status == 1)] <- 1

Tau1 <- 0
Tau2 <- 8
x <- c(-1, 1, -0.6) # given covariate profile for the pure risk

# Estimation using the stratified case cohort with design weights
mod <- coxph(Surv(event.time, status) ~ X1 + X2 + X3, data = casecohort,
                   weight = weights, id = id, robust = TRUE)
est <- influences(mod, Tau1 = Tau1, Tau2 = Tau2, x = x)

# robust variance estimate for the log-relative hazard
robustvariance(est$infl.beta)

# robust variance estimate for the cumulative baseline hazard estimate
robustvariance(est$infl.Lambda0.Tau1Tau2)

# robust variance estimate for the pure risk estimate
robustvariance(est$infl.Pi.x.Tau1Tau2)
```

`variance` *variance*

Description

Computes the variance estimate that follows the complete variance decomposition, for a parameter such as log-relative hazard, cumulative baseline hazard or covariate specific pure-risk.

Usage

```
variance(n, casecohort, weights = NULL, infl, calibrated = NULL,
infl2 = NULL, cohort = NULL, stratified = NULL,
variance.phase2 = NULL)
```

Arguments

`n`	number of individuals in the whole cohort.
`casecohort`	If `stratified = TRUE`, data frame with `status` (case status), `weights` (design, if they are not provided in the argument below), `W` (the J strata), `strata.m` (vector of length J with the numbers of sampled individuals in the strata) and `strata.n` (vector of length J with the strata sizes in the cohort), for each individual in the stratified case-cohort data. If `stratified = FALSE`, data frame with `weights` (design, if they are not provided in the argument below), `m` (number of sampled individuals) and `n` (cohort size), for each individual in the unstratified case-cohort data.
`weights`	vector with design weights for the individuals in the case-cohort data.
`infl`	matrix with the overall influences on the parameter.
`calibrated`	are calibrated weights used for the estimation of the parameters? If `calibrated = TRUE`, the arguments below need to be provided. Default is `FALSE`.
`infl2`	matrix with the phase-two influences on the parameter. Needs to be provided if `calibrated = TRUE`.
`cohort`	If `stratified = TRUE`, data frame with `status` (case status) and `subcohort` (subcohort sampling indicators) for each individual in the stratified case-cohort data. If `stratified = FALSE`, data frame with `status` (case status) and `unstrat.subcohort` (subcohort unstratified sampling indicators) for each individual in the unstratified case-cohort data. Needs to be provided if `calibrated = TRUE`.
`stratified`	was the sampling of the case-cohort stratified on `W`? Default is `FALSE`.
`variance.phase2`	should the phase-two variance component also be returned? Default is `FALSE`.

Details

`variance` works for estimation from a case-cohort with design weights or calibrated weights (case-cohort consisting of the subcohort and cases not in the subcohort, i.e., case-cohort obtained from two phases of sampling).

If covariate information is missing for certain individuals in the phase-two data (i.e., case-cohort obtained from three phases of sampling), use `variance.missingdata`.

`variance` uses the variance formulas provided in Etievant and Gail (2024).

Value

`variance`: variance estimate.

`variance.phase2`: phase-two variance component.

References

Etievant, L., Gail, M. H. (2024). Cox model inference for relative hazard and pure risk from stratified weight-calibrated case-cohort data. Lifetime Data Analysis, 30, 572-599.

See Also

`influences`, `influences.RH`, `influences.CumBH`, `influences.PR`, `robustvariance` and `variance.missingdata`.

Examples

```
data(dataexample.stratified, package="CaseCohortCoxSurvival")
cohort <- dataexample.stratified$cohort
casecohort <- cohort[which(cohort$status == 1 |
                     cohort$subcohort == 1),] # the stratified case-cohort
casecohort$weights <- casecohort$strata.n / casecohort$strata.m
casecohort$weights[which(casecohort$status == 1)] <- 1

Tau1 <- 0
Tau2 <- 8
x <- c(-1, 1, -0.6) # given covariate profile for the pure risk
n <- nrow(cohort)

# Estimation using the stratified case-cohort with design weights
mod <- coxph(Surv(event.time, status) ~ X1 + X2 + X3, data = casecohort,
             weight = weights, id = id, robust = TRUE)

# parameters and influences estimation
est           <- influences(mod, Tau1 = Tau1, Tau2 = Tau2, x = x)
beta.hat      <- est$beta.hat
Lambda0.hat   <- est$Lambda0.Tau1Tau2.hat
Pi.x.hat      <- est$Pi.x.Tau1Tau2.hat
infl.beta     <- est$infl.beta
infl.Lambda0  <- est$infl.Lambda0.Tau1Tau2
infl.Pi.x     <- est$infl.Pi.x.Tau1Tau2

# variance estimate for the log-relative hazard estimate
variance(n = n, casecohort = casecohort, infl = infl.beta, stratified = TRUE)

# variance estimate for the cumulative baseline hazard estimate
variance(n = n, casecohort = casecohort, infl = infl.Lambda0,
         stratified = TRUE)
```

```
# variance estimate for the pure risk estimate
variance(n = n, casecohort = casecohort, infl = infl.Pi.x, stratified = TRUE)
```

`variance.missingdata` *variance.missingdata*

Description

Computes the variance estimate that follows the complete variance decomposition, for a parameter such as log-relative hazard, cumulative baseline hazard or covariate specific pure-risk, when covariate information is missing for individuals in the phase-two sample.

Usage

```
variance.missingdata(n, casecohort, casecohort.phase2, weights,
weights.phase2, weights.p2.phase2, infl2, infl3, stratified.p2 = NULL,
estimated.weights = NULL)
```

Arguments

`n`	number of individuals in the whole cohort.
`casecohort`	If `stratified = TRUE`, data frame with `W` (the J phase-two strata), `strata.m` (vector of length J with the numbers of sampled individuals in the strata in the second phase of sampling) and `strata.n` (vector of length J with the strata sizes in the cohort), for each individual in the stratified case cohort data. If `stratified = FALSE`, data frame with `m` (number of sampled individuals in the second phase of sampling) and `n` (cohort size), for each individual in the unstratified case cohort data.
`casecohort.phase2`	If `stratified = TRUE`, data frame with `W` (the J phase-two strata), `strata.m` (vector of length J with the numbers of sampled individuals in the strata in the second phase of sampling), `strata.n` (vector of length J with the strata sizes in the cohort) and `phase3` (phase-three sampling indicator), for each individual in the phase-two sample. If `stratified = FALSE`, data frame with `m` (number of sampled individuals in the second phase of sampling), `n` (cohort size) and unstrat.phase3 (phase-three sampling indicator), for each individual in the phase-two sample.
`weights`	vector with design weights for the individuals in the case cohort data.
`weights.phase2`	vector with design weights for the individuals in the phase-two sample.
`weights.p2.phase2`	vector with phase-two design weights for the individuals in the phase-two sample.
`infl2`	matrix with the phase-two influences on the parameter.
`infl3`	matrix with the phase-three influences on the parameter.

`stratified.p2`	was the second phase of sampling stratified on `W`? Default is `FALSE`.
`estimated.weights`	were the phase-three weights estimated? Default is `FALSE`.

Details

`variance.missingdata` works for estimation from a case cohort with design weights and when covariate information was missing for certain individuals in the phase-two data (i.e., case cohort obtained from three phases of sampling and consisting of individuals in the phase-two data without missing covariate information).

If there are no missing covariates in the phase- two sample, use `variance` with either design weights or calibrated weights.

`variance.missingdata` uses the variance formulas provided in Etievant and Gail (2024).

Value

`variance`: variance estimate.

References

Etievant, L., Gail, M. H. (2024). Cox model inference for relative hazard and pure risk from stratified weight-calibrated case-cohort data. Lifetime Data Analysis, 30, 572-599.

See Also

`influences.missingdata`, `influences.RH.missingdata`, `influences.CumBH.missingdata`, `influences.PR.missingdata`, `robustvariance` and `variance`.

Examples

```
data(dataexample.missingdata.stratified, package="CaseCohortCoxSurvival")

cohort <- dataexample.missingdata.stratified$cohort
phase2 <- cohort[which(cohort$phase2 == 1),] # the phase-two sample
casecohort <- cohort[which(cohort$phase3 == 1),] # the stratified case-cohort

B.phase2 <- cbind(1 * (phase2$W3 == 0), 1 * (phase2$W3 == 1))
rownames(B.phase2)  <- cohort[cohort$phase2 == 1, "id"]
B.phase3 <- cbind(1 * (casecohort$W3 == 0), 1 * (casecohort$W3 == 1))
rownames(B.phase3)  <- cohort[cohort$phase3 == 1, "id"]
total.B.phase2 <- colSums(B.phase2)
J3 <- ncol(B.phase3)
n <- nrow(cohort)

# Quantities needed for estimation of the cumulative baseline hazard when
# covariate data is missing
mod.cohort <- coxph(Surv(event.time, status) ~ X2, data = cohort,
                    robust = TRUE) # X2 is available on all cohort members
mod.cohort.detail <- coxph.detail(mod.cohort, riskmat = TRUE)
```

```
riskmat.phase2 <- with(cohort, mod.cohort.detail$riskmat[phase2 == 1,])
rownames(riskmat.phase2) <- cohort[cohort$phase2 == 1, "id"]
observed.times.phase2 <- apply(riskmat.phase2, 1,
                               function(v) {which.max(cumsum(v))})
dNt.phase2 <- matrix(0, nrow(riskmat.phase2), ncol(riskmat.phase2))
dNt.phase2[cbind(1:nrow(riskmat.phase2), observed.times.phase2)] <- 1
dNt.phase2 <- sweep(dNt.phase2, 1, phase2$status, "*")
colnames(dNt.phase2) <- colnames(riskmat.phase2)
rownames(dNt.phase2) <- rownames(riskmat.phase2)

Tau1 <- 0 # given time interval for the pure risk
Tau2 <- 8
x <- c(-1, 1, -0.6) # given covariate profile for the pure risk

# Estimation using the stratified case cohort with true known design weights

mod.true <- coxph(Surv(event.time, status) ~ X1 + X2 + X3, data = casecohort,
                  weight = weight.true, id = id, robust = TRUE)

est.true <- influences.missingdata(mod = mod.true,
                                   riskmat.phase2 = riskmat.phase2,
                                   dNt.phase2 = dNt.phase2, Tau1 = Tau1,
                                   Tau2 = Tau2, x = x)
infl.beta.true <- est.true$infl.beta
infl.Lambda0.true <- est.true$infl.Lambda0.Tau1Tau2
infl.Pi.x.true <- est.true$infl.Pi.x.Tau1Tau2
infl2.beta.true <- est.true$infl2.beta
infl2.Lambda0.true <- est.true$infl2.Lambda0.Tau1Tau2
infl2.Pi.x.true <- est.true$infl2.Pi.x.Tau1Tau2
infl3.beta.true <- est.true$infl3.beta
infl3.Lambda0.true <- est.true$infl3.Lambda0.Tau1Tau2
infl3.Pi.x.true <- est.true$infl3.Pi.x.Tau1Tau2

# variance estimate for the log-relative hazard estimate
variance.missingdata(n = n, casecohort = casecohort,
                     casecohort.phase2 = phase2,
                     weights = casecohort$weight.true,
                     weights.phase2 = phase2$weight.true,
                     weights.p2.phase2 = phase2$weight.p2.true,
                     infl2 = infl2.beta.true, infl3 = infl3.beta.true,
                     stratified.p2 = TRUE)

# variance estimate for the cumulative baseline hazard estimate
variance.missingdata(n = n, casecohort = casecohort,
                     casecohort.phase2 = phase2,
                     weights = casecohort$weight.true,
                     weights.phase2 = phase2$weight.true,
                     weights.p2.phase2 = phase2$weight.p2.true,
                     infl2 = infl2.Lambda0.true, infl3 = infl3.Lambda0.true,
                     stratified.p2 = TRUE)

# variance estimate for the pure risk estimate
variance.missingdata(n = n, casecohort = casecohort,
```

```
                      casecohort.phase2 = phase2,
                      weights = casecohort$weight.true,
                      weights.phase2 = phase2$weight.true,
                      weights.p2.phase2 = phase2$weight.p2.true,
                      infl2 = infl2.Pi.x.true, infl3 = infl3.Pi.x.true,
                      stratified.p2 = TRUE)


# Estimation using the stratified case cohort with estimated weights, and
# accounting for the estimation through the influences

mod.estimated <- coxph(Surv(event.time, status) ~ X1 + X2 + X3,
                       data = casecohort, weight = weight.est, id = id,
                       robust = TRUE)

est.estimated  <- influences.missingdata(mod.estimated,
                                         riskmat.phase2 = riskmat.phase2,
                                         dNt.phase2 = dNt.phase2,
                                         estimated.weights = TRUE,
                                         B.phase2 = B.phase2, Tau1 = Tau1,
                                         Tau2 = Tau2, x = x)

infl.beta.estimated <- est.estimated$infl.beta
infl.Lambda0.estimated <- est.estimated$infl.Lambda0.Tau1Tau2
infl.Pi.x.estimated <- est.estimated$infl.Pi.x.Tau1Tau2
infl2.beta.estimated <- est.estimated$infl2.beta
infl2.Lambda0.estimated <- est.estimated$infl2.Lambda0.Tau1Tau2
infl2.Pi.x.estimated <- est.estimated$infl2.Pi.x.Tau1Tau2
infl3.beta.estimated <- est.estimated$infl3.beta
infl3.Lambda0.estimated <- est.estimated$infl3.Lambda0.Tau1Tau2
infl3.Pi.x.estimated <- est.estimated$infl3.Pi.x.Tau1Tau2

# variance estimate for the log-relative hazard
variance.missingdata(n = n, casecohort = casecohort,
                     casecohort.phase2 = phase2,
                     weights = casecohort$weight.est,
                     weights.phase2 = phase2$weight.est,
                     weights.p2.phase2 = phase2$weight.p2.true,
                     infl2 = infl2.beta.estimated,
                     infl3 = infl3.beta.estimated,
                     stratified.p2 = TRUE, estimated.weights = TRUE)

# variance estimate for the cumulative baseline hazard estimate
variance.missingdata(n = n, casecohort = casecohort,
                     casecohort.phase2 = phase2,
                     weights = casecohort$weight.est,
                     weights.phase2 = phase2$weight.est,
                     weights.p2.phase2 = phase2$weight.p2.true,
                     infl2 = infl2.Lambda0.estimated,
                     infl3 = infl3.Lambda0.estimated,
                     stratified.p2 = TRUE, estimated.weights = TRUE)

# variance estimate for the pure risk estimate
```

```
variance.missingdata(n = n, casecohort = casecohort,
                     casecohort.phase2 = phase2,
                     weights = casecohort$weight.est,
                     weights.phase2 = phase2$weight.est,
                     weights.p2.phase2 = phase2$weight.p2.true,
                     infl2 = infl2.Pi.x.estimated,
                     infl3 = infl3.Pi.x.estimated,
                     stratified.p2 = TRUE, estimated.weights = TRUE)
```

Index