Package 'InferenceSMR'

October 12, 2022

Type Package

Title Inference About the Standardized Mortality Ratio when Evaluating the Effect of a Screening Program on Survival

Version 1.0.1

Date 2022-05-04

Author Denis Talbot, Thierry Duchesne, Jacques Brisson, Nathalie Vandal

Maintainer Denis Talbot <denis.talbot@fmed.ulaval.ca>

Description Functions to make inference about the standardized mortality ratio (SMR) when evaluating the effect of a screening program. The package is based on methods described in Sasieni (2003) <doi:10.1097/00001648-200301000-00026> and Talbot et al. (2011) <doi:10.1002/sim.4334>.

License GPL (>= 2)

Depends survival, methods

Encoding latin1

NeedsCompilation no

Repository CRAN

Date/Publication 2022-05-09 07:20:08 UTC

R topics documented:

contrib *Weights calculation*

Description

contrib is a function that finds out how much time was contributed for every combination of the incidence covariates, the survival covariates, and follow up time. Using these contibutions makes the calculation of the variance of the expected number of deaths much more efficient than making the calculations on the raw data.

Usage

```
contrib(start_follow, end_follow, incid_cov, surv_cov, follow_up, increment)
```

Arguments

start_follow	A vector that contains the amount of follow-up time elapsed when this set of covariate values started.
end_follow	A vector that contains the amount of follow-up time that will have elapsed when this set of covariate values changes or when follow-up ends.
incid_cov	A vector that contains the values of the covariates for incidence.
surv_cov	A vector that contains the values of the covariates for survival.
follow_up	A vector that contains the total follow-up time for that individual.
increment	The value of the time increment that will be used for Sasieni's estimator and variance (a numeric).

Details

If only time independent covariates are used, then all vectors are of dimension n = sample size.

Otherwise, the function contrib needs an augmented dataset, in which each person-time corresponds to one row. Each row contains the values of the time-invariant covariates and the updated values of time-dependent variables. Therefore, the covariates must have a finite number of values (they must be discrete).

For example, if only the covariate "age" is used for incidence and survival, an individual followed for 3.4 years that was 54.6 years at the begining of the study should be entered as follow:

```
#start_follow, end_follow, incid_cov, surv_cov, follow_up
0 0.4 54 54 3.4
0.4 1.4 55 55 3.4
1.4 2.4 56 56 3.4
2.4 3.4 57 57 3.4
```

An example of the code required to do such a thing is provided in the screening dataset help page.

Value

The resulting list is directly usable in functions `est.expDeath`, `var.expDeath` and `inference.SMR`. The list contains the following elements :

`contrib`	A matrix giving the total time contributed for every combination of discretized follow-up times, incidence covariates and survival covariates.
`ncov.incid`	The number of covariates for incidence.
`ncov.surv`	The number of covariates for survival.
`increment`	The increment used for discretization of follow-up times.

Author(s)

Denis Talbot, Thierry Duchesne, Jacques Brisson, Nathalie Vandal.

See Also

`est.expDeath`, `var.expDeath`, `inference.SMR`, `screening`

`est.expDeath` *Estimation of the expected number of deaths.*

Description

Estimation of the expected number of deaths in a screening program using the method proposed by Sasieni (2003).

Usage

```
est.expDeath(contribution, incid, cox, fuzz, covnames)
```

Arguments

`contribution`	An object of contributions produced by the function `contrib`.
`incid`	A matrix containing: the incidences, the value of the covariates and the person-years at risk, in that order. It can be obtained with the function `incidences`.
`cox`	An oject of class `coxph` containing the model that was used to estimate the survival in the cohort of non-participants.
`fuzz`	Numerical precision is problematic when it comes to test equality between objects. The option `fuzz` is used to consider objects not differing by more than `fuzz` to be equal. The `fuzz` option should be chosen to be a small positive number, for instance 0.0001.
`covnames`	An alphanumeric vector containing the names of the covariates used to estimate the survival in the cohort of non-participants, that is, the names of the covariates used to obtain the `cox` object.

Value

Returns the expected number of deaths

Note

A complete example of usage is provided in the help page of the `screening` dataset.

Author(s)

Denis Talbot, Thierry Duchesne, Jacques Brisson, Nathalie Vandal.

References

Sasieni P. (2003) *On the expected number of cancer deaths during follow-up of an initially cancer-free cohort*. Epidemiology, 14, 108-110.

See Also

`var.expDeath`,`inference.SMR`, `screening`

Examples

```
#This example uses pre-built objects and shows the simple usage
#of the est.expDeath function when those objects already exists.
#For an example of how to built those object, refer to the
#help page of the screening dataset.

data(req.objects);
cox.data = req.objects$cox.data;
#Remove "#" to run example :
#est.expDeath(req.objects$contribution,req.objects$incid,req.objects$cox,fuzz = 0.01,
#req.objects$covnames);

#[1] 33.44264
```

`incidences` *Incidences calculations.*

Description

A function to calculate incidence rates for every combination of age and calendar years.

Usage

```
incidences(age_min, age_max, year_min, year_max, follow_up,
start_age, start_year, case)
```

Arguments

`age_min`	The age at which incidences should begin to be calculated.
`age_max`	The age at which incidences should stop to be calculated.
`year_min`	The calendar year at which indicidences should begin to be calculated.
`year_max`	The calendar year at which incidences should stop to be calculated.
`follow_up`	A vector of dimension `n` of follow-up times as non-participants.
`start_age`	A vector of dimension `n` of ages at the begining of the follow-up.
`start_year`	A vector of dimension `n` of calendar years at the begining of the follow-up.
`case`	A vector of dimension `n` where each component equals 1 if the follow-up ended because the individual was infected with the decease and 0 otherwise.

Details

This function can be used to obtain incidences for the functions `est.expDeath`, `var.expDeath` and `inference.SMR`. A complete example of usage is provided in the help page of the `screening` dataset.

Value

A matrix whose first column is the number of person-years at risk, the second column is the calendar years, the third column is the ages and the fourth column is the incidence rates.

Author(s)

Denis Talbot, Thierry Duchesne, Jacques Brisson, Nathalie Vandal.

See Also

`est.expDeath`, `var.expDeath`, `inference.SMR`, `screening`

`inference.SMR` *Inference about the standardized mortality ratio (SMR) when evaltuating the effect of a screening program on survival.*

Description

This function estimates the expected number of deaths, its variance, the SMR and confidence intervals about the SMR.

Usage

```
inference.SMR(obs.death, normal = "log-smr", alpha = 0.05, contribution,
incid, cox, fuzz = 0.01, Poisson = FALSE, covnames)
```

Arguments

`obs.death`	The observed number of deaths for the people participating in the screening program. A numeric value.
`normal`	Indicates at which level should the normality assumption be made, either at the SMR level, at the log-SMR level or at the root-SMR level. A character vector containing one or many of the following elements “smr”, “log-smr” and “root-smr”.
`alpha`	The nominal error rate of the confidence intervals. A numeric value between 0 and 1, e.g. 0.05 to obtain a 95
`contribution`	An object of contributions produced by the function `contrib`.
`incid`	A matrix containing: the incidences, the value of the covariates and the person-years at risk, in that order. It can be obtained with the function `incidences`.
`cox`	An oject of class `coxph` containing the model that was used to estimate the survival in the cohort of non-participants.
`fuzz`	Numerical precision is problematic when it comes to test equality between objects. The option `fuzz` is used to consider objects not differing by more than `fuzz` to be equal. The `fuzz` option should be chosen to be a small positive number, for instance 0.0001.
`Poisson`	Indicates whether the incidences’ variance should be estimated with a Poisson distribution (TRUE) or a binomial distribution (FALSE). The default is FALSE.
`covnames`	An alphanumeric vector containing the names of the covariates used to estimate the survival in the cohort of non-participants, that is, the names of the covariates used to obtain the `cox` object.

Details

The `inference.SMR` function estimates the expected number of deaths as in Sasieni (2003), estimates the variance of the expected number of deaths and builds confidence intervals as in Talbot et al. (2011). As suggested in the latter, the variance of the observed number of deaths is estimated by the observed number of deaths.

Value

`expected`	The expected number of deaths
`obs.death`	The observed number of deaths
`variance`	The variance of the expected number of deaths
`smr`	The standardized mortality ratio
`smr.var`	The variance of the SMR. Only returned if “smr” was given in the `normal` argument.
`smr.ci`	A 1-`alpha` confidence interval for the SMR. Only returned if “smr” was given in the `normal` argument.
`logSMR.var`	The variance of the natural logarithm of the SMR. Only returned if “log-smr” was given in the `normal` argument.

`logSMR.ci`	A 1-alpha confidence interval for the log-SMR. Only returned if "log-smr" was given in the `normal` argument.
`rootSMR.var`	The variance of the square root of the SMR. Only returned if "root-smr" was given in the `normal` argument.
`rootSMR.ci`	A 1-alpha confidence interval for the root-SMR. Only returned if "root-smr" was given in the `normal` argument.

Note

A complete example of usage is provided in the help page of the `screening` dataset.

Author(s)

Denis Talbot, Thierry Duchesne, Jacques Brisson, Nathalie Vandal.

References

Sasieni P. (2003) *On the expected number of cancer deaths during follow-up of an initially cancer-free cohort*. Epidemiology, 14, 108-110.

Talbot, D., Duchesne, T., Brisson, J., Vandal, N. (2011) *Variance estimation and confidence intervals for the standardized mortality ratio with application to the assessment of a cancer screening program*, Statistics in Medicine, 30, 3024-3037.

See Also

`est.expDeath`, `var.expDeath`, `screening`

Examples

```
#This example uses pre-built objects and shows the simple usage
#of the est.expDeath function when those objects already exist.
#For an example of how to build those objects, refer to the
#help page of the screening dataset.

#Estimating the variance can be very long even in this small sample example, e.g. a few hours.
#Remove "#" to run example :
#data(req.objects);
#cox.data = req.objects$cox.data;
#results = inference.SMR(obs.death = sum(screening$deathSCN),
# normal = c("smr", "log-smr", "root-smr"),
#  alpha = 0.05, req.objects$contribution, req.objects$incid,
#  cox = req.objects$cox, fuzz = 0.01, Poisson = TRUE, req.objects$covnames);


#********  INFERENCE ABOUT THE SMR  *********
#
#Observed =  18  Expected =  33.44264
#Obs.var. =  18  Exp.var. =  39.38153
#SMR =  0.5382351
#
```

```
# 95 % Confidence intervals with normality assumption at :
#
#The SMR level : ( 0.2204119 0.8560583 )
#
#The log-SMR level : ( 0.2982118 0.9714471 )
#
#The root-SMR level : ( 0.2673299 0.9029762 )

#results
#
#$expected
#[1] 33.44264
#
#$obs.death
#[1] 18
#
#$variance
#           2
#[1,] 39.38153
#
#$smr
#[1] 0.5400112
#
#$smr.var
#             2
#[1,] 0.02629511
#
#$smr.ci
#[1] 0.2204119 0.8560583
#
#$logSMR.var
#             2
#[1,] 0.09076763
#
#$logSMR.ci
#[1] 0.2982118 0.9714471
#
#$rootSMR.var
#             2
#[1,] 0.01221358
#
#$rootSMR.ci
#[1] 0.2673299 0.9029762
```

`screening` *A population of size 10,000 for a screening program*

Description

This dataset contains a simulated population of size 10,000. The population was simulated as described in Talbot et al (2011).

Usage

```
data(screening)
```

Format

A data frame with 10000 observations on the following 12 variables.

`yearSCN` Year at which the person would become a participant in the screening program

`ageSCN` Age at which the person would become a participant in the screening program

`yearONS` Year at which the disease onset would happen for a non-participant

`deathBC` Indicator variable that has a value of 1 if the non-participant dies from the screened disease, 0 otherwise.

`ageFL` Age at which the person is eligible to the screening program for the first time

`yearFL` Year at which the person is eligible to the screening program for the first time

`followONS` Follow-up time for a non-participant after the disease onset

`followSCN` Follow-up time as participant in the screening program

`particip` Indicator variable that has a value of 1 if the individual eventually became a participant in the screening program, 0 otherwise

`Onset` Indicator variable that has a value of 1 if the disease onset happens while the person is a non-participant

`end` Year at which the follow-up ends

`deathSCN` Indicator variable that has a value of 1 if the participants dies from the screened disease, 0 otherwise.

Details

Note that even though there are no missing values in the dataset, some events do not occur. For example, if `yearsSCN` has a greater value than `end`, then the inidividual never becomes a participant.

Examples

```
require(survival); #load survival package;
data(screening);
head(screening); #Data to be used in the example;
NB = nrow(screening); #Sample size

#Be careful with R round function. If it was used to obtain discrete value, then
#fuzz option should be used for expected and expected variance
i=1:NB
yearSCN<-screening[i,1]; #Year at which the woman started participating
ageSCN<-screening[i,2]; #Age at which the woman started participating
yearONS<-screening[i,3]; #Year of breast cancer diagnosis
```

```
deathBC<-screening[i,4]; #Death by breast cancer indicator for non-participating woman.
ageFL<-screening[i,5]; #Age at which the woman became eligible
yearFL<-screening[i,6]; #Year at which the woman became eligible
followONS<-screening[i,7]; #Follow-up time after disease onset for a non-participant
followSCN<-screening[i,8]; #Follow-up time as a participant in the screening program
particip<-screening[i,9]; #Indicator that the woman participated into the screening
                          #program at some point
Onset<-screening[i,10]; #Indicator that the non-participating woman got breast cancer
end<-screening[i,11]; #year of eligibility end.
deathSCN<-screening[i,12]; #Death by breast cancer indicator for participating woman.

nb_onset=length(Onset[Onset==1]); #Number of women with breast cancer

#Objects that will containt covariates for the Cox model
year<-numeric(nb_onset);
age1<-numeric(nb_onset);
age2<-numeric(nb_onset);
age3<-numeric(nb_onset);
age4<-numeric(nb_onset);
ti<-numeric(nb_onset);

year[yearONS[Onset==1] <= 3] = 1; #Indicator that diagnosis happened before year 3
year[yearONS[Onset==1] > 3] = 0; #Indicator that diagnosis happened before year 3
age1[ageSCN[Onset==1] < 55] = 1; #Indicator that age at diagnosis is smaller than 55
age1[ageSCN[Onset==1] >= 55] = 0; #Indicator that age at diagnosis is smaller than 55
age2[ageSCN[Onset==1] >= 55 & ageSCN[Onset==1] < 60] = 1; #... >= 55 and < 60
age2[ageSCN[Onset==1] < 55 | ageSCN[Onset==1] >= 60] = 0; #... >= 55 and < 60
age3[ageSCN[Onset==1] >= 60 & ageSCN[Onset==1] < 65] = 1; #... >= 60 and < 65
age3[ageSCN[Onset==1] < 60 | ageSCN[Onset==1] >= 65] = 0; #... >= 60 and < 65
age4[ageSCN[Onset==1] >= 65 & ageSCN[Onset==1] < 70] = 1; #... >= 65 and < 70
age4[ageSCN[Onset==1] < 65 | ageSCN[Onset==1] >= 70] = 0; #... >= 65 and < 70
cox.data = data.frame(followONS = followONS[Onset == 1], deathBC = deathBC[Onset == 1],
year, age1, age2, age3, age4);
x<-coxph(Surv(time = followONS, event = deathBC, type = 'right')~ year + age1 + age2 + age3 + age4,
data = cox.data, method="breslow",control=coxph.control(iter.max=100))

#Creating a matrix with many more lines than what will be used
new_data<-matrix(0,nrow=12*sum(particip),ncol=10);

#Creating a matrix containing data in a new form.
#Each line contains stable covariates, so that a
#given individual might be divided on many lines.
#For example, if only the covariate age is used for incidence and survival,
#an individual followed for 3.4 years that was 54.6 years at the begining
#of the study should be entered as follow:

#start_follow, end_follow, incid_cov, surv_cov, follow_up
#0   0.4 54 54 3.4
#0.4 1.4 55 55 3.4
#1.4 2.4 56 56 3.4
#2.4 3.4 57 57 3.4
```

```
r=1
for(i in seq(1,length(ageFL))[particip==1])
{
        X = followSCN[i];
        dep_t = yearSCN[i];
        age_t = ageSCN[i];
        while(dep_t - yearSCN[i] < X)
        {
                Y = min(floor(age_t) + 1 - age_t, floor(dep_t) + 1 - dep_t);
                if(dep_t - yearSCN[i] + Y >= X)
                {
                        new_data[r,]<-c(floor(age_t), floor(dep_t), age_t < 55,
                        (age_t >= 55 && age_t < 60),
                        (age_t >= 60 && age_t < 65), (age_t >=65 && age_t <70),
                        dep_t < 3, dep_t - yearSCN[i], X, followSCN[i]);
                }
                else
                {
                        new_data[r,]<-c(floor(age_t), floor(dep_t), age_t < 55,
                        (age_t >= 55 && age_t < 60),
                        (age_t >= 60 && age_t < 65), (age_t >=65 && age_t <70),
                    dep_t < 3, dep_t - yearSCN[i], dep_t - yearSCN[i] + Y, followSCN[i]);
                }
                dep_t = dep_t + Y;
                age_t = age_t + Y;
                r = r + 1;
        }
}
new_data<-new_data[new_data[,1]!=0,];
new_data[1:10,];

#Calculate incidences with incidences function:
#follow up time as non-participant:
follow_up = apply(cbind(end - yearFL,yearONS - yearFL,yearSCN - yearFL),1,min);
incid = incidences(50,75,0,5,follow_up,ageFL,yearFL,Onset);


#Calculate contributions with contrib function:
start_follow = new_data[,8];
end_follow = new_data[,9];
incid_cov = new_data[,c(2,1)];
surv_cov = data.frame(new_data[,c(7,3,4,5,6)]);
follow_up = new_data[,10];
increment = 0.5;


#Remove following "#" to run example :

#contribution = contrib(start_follow, end_follow, incid_cov, surv_cov,
#follow_up, increment);
#est.expDeath(contribution,incid,x,fuzz = 0.01,
#covnames = c("year", "age1", "age2", "age3", "age4"));
```

```
#Estimating the variance can be very long even in this small sample example, e.g. a few hours.
#Remove the "#" to run example:
#var.expDeath(contribution,incid,x,fuzz = 0.01,
#covnames = c("year", "age1", "age2", "age3", "age4"));

#Estimating the variance can be very long even in this small sample example, e.g. a few hours.
#Remove the "#" to run example:

#results = inference.SMR(obs.death = sum(deathSCN), normal = c("smr", "log-smr", "root-smr"),
#         alpha = 0.05, contribution, incid, cox = x, fuzz = 0.01, Poisson = TRUE,
#   covnames =  c("annees", "age1", "age2", "age3", "age4"));


#********  INFERENCE ABOUT THE SMR  *********
#
#Observed =  18  Expected =  33.44264
#Obs.var. =  18  Exp.var. =  39.38153
#SMR =  0.5382351
#
# 95 % Confidence intervals with normality assumption at :
#
#The SMR level : ( 0.2204119 0.8560583 )
#
#The log-SMR level : ( 0.2982118 0.9714471 )
#
#The root-SMR level : ( 0.2673299 0.9029762 )

#results
#
#$expected
#[1] 33.44264
#
#$obs.death
#[1] 18
#
#$variance
#            2
#[1,] 39.38153
#
#$smr
#[1] 0.5400112
#
#$smr.var
#              2
#[1,] 0.02629511
#
#$smr.ci
#[1] 0.2204119 0.8560583
#
#$logSMR.var
#              2
#[1,] 0.09076763
#
```

```
#$logSMR.ci
#[1] 0.2982118 0.9714471
#
#$rootSMR.var
#             2
#[1,] 0.01221358
#
#$rootSMR.ci
#[1] 0.2673299 0.9029762
```

var.expDeath *Variance estimation of the expected number of deaths*

Description

This function estimates the variance of the expected number of deaths when the latter is estimated using Sasieni's method.

Usage

```
var.expDeath(contribution, incid, cox, fuzz, Poisson = FALSE, covnames)
```

Arguments

`contribution`	An object of contributions produced by the function `contrib`.
`incid`	A matrix containing: the incidences, the value of the covariates and the person-years at risk, in that order. It can be obtained with the function `incidences`.
`cox`	An oject of class `coxph` containing the model that was used to estimate the survival in the cohort of non-participants.
`fuzz`	Numerical precision is problematic when it comes to test equality between objects. The option `fuzz` is used to consider objects not differing by more then `fuzz` to be equal. The `fuzz` option should be chosen to be a small positive number, for instance 0.0001.
`covnames`	An alphanumeric vector containing the names of the covariates used to estimate the survival in the cohort of non-participants, that is, the names of the covariates used to obtain the `cox` object.
`Poisson`	Indicates whether the incidences' variance should be estimated with a Poisson distribution (TRUE) or a binomial distribution (FALSE). The default is FALSE.

Value

The function returns the variance of the expected number of deaths

Note

A complete example of how to use this function is available in the help page of the `screening` dataset.

Author(s)

Denis Talbot, Thierry Duchesne, Jacques Brisson, Nathalie Vandal.

References

Talbot, D., Duchesne, T., Brisson, J., Vandal, N. (2011) *Variance estimation and confidence intervals for the standardized mortality ratio with application to the assessment of a cancer screening program*, Statistics in Medicine, 30, 3024-3037.

See Also

`est.expDeath`, `inference.SMR`, `screening`

Examples

```
#This example uses pre-built objects and shows the simple usage
#of the est.expDeath function when those objects already exists.
#For an example of how to built those object, refer to the
#help page of the screening dataset.

#Remove "#" to run example. The function can be quite long (a few hours) to run:
#data(req.objects);
#cox.data = req.objects$cox.data;
#var.expDeath(req.objects$contribution,req.objects$incid,req.objects$cox,fuzz = 0.01,
#req.objects$covnames);


#[1,] 39.31382
```

Index