Package 'SurvMI'

October 12, 2022

Type Package

Title Multiple Imputation Method in Survival Analysis

Version 0.1.0

Depends R(>= 3.4.0)

Imports survival (>= 3.1.11), zoo, stats, graphics, base

Maintainer Yiming Chen <yimingc1208@gmail.com>

Description In clinical trials, endpoints are sometimes evaluated with uncertainty. Adjudication is commonly adopted to ensure the study integrity. We propose to use multiple imputation (MI) introduced by Robin (1987) <doi:10.1002/9780470316696> to incorporate these uncertainties if reasonable event probabilities were provided. The method has been applied to Cox Proportional Hazard (PH) model, Kaplan-Meier (KM) estimation and Log-rank test in this package. Moreover, weighted estimations discussed in Cook (2004) <doi:10.1016/S0197-2456(00)00053-2> were also implemented with weights calculated from event probabilities. In conclusion, this package can handle time-to-event analysis if events presented with uncertainty by different methods.

License GPL-2

Encoding UTF-8

LazyData TRUE

RoxygenNote 7.1.1

NeedsCompilation no

Author Yiming Chen [aut, cre],
John Lawrence [ctb]

Repository CRAN

Date/Publication 2020-07-13 08:50:03 UTC

R topics documented:

`CoxMI` *Cox PH model with MI method*

Description

CoxMI function estimated Cox model with uncertain endpoints by using MI method. Users have to provide survival data in a long format with rows for all potential events, together with corresponding event probabilities. The long format data should be transformed by the uc_data_transform function into a data list before feed into the function.

Usage

```
CoxMI(data_list,nMI=1000,covariates=NULL,id=NULL,...)
```

Arguments

`data_list`	The data list which has been transformed from the long format by the uc_data_transform function.
`nMI`	Number of imputations (>1).
`covariates`	Vector of covariates on the RHS of Cox model. Categorical variables need to be encoded as factor variables before entering the model. This encoding has to be done before the data transform step.
`id`	Vector of id variable if Andersen-Gill model is required.
`...`	Other arguments passed on to coxph().

Details

Calculates the estimated parameters as in the usual Cox proportional hazards model when event uncertainties present. The data are assumed to consist of potential event times with probabilities or weights between 0 and 1 corresponding to the probability that an event occurred at each time.

Value

`est`	Estimated vector of coefficients in the model
`var`	Estimated variance of the coefficients
`betamat`	Matrix containing estimate of coefficient from each imputed dataset
`Var_mat`	Array containing variances for each imputed dataset
`Between Var`	Between imputation variance

Within Var	Mean within imputed dataset variance
nMI	Number of imputed datasets
pvalue	Estimated two-sided p-value
en	Expected events count - mean event count of imputed datasets

Author(s)

Yiming Chen, John Lawrence

References

[1] Rubin DB. Multiple Imputation for Nonresponse in Surveys. New York: Wiley; 1987

See Also

Coxwt, CoxMI.summ.

Examples

```
set.seed(128)
df_x<-data_sim(n=500,true_hr=0.8,haz_c=0.5/365)
df_x$f.trt<-as.factor(df_x$trt_long)
data_intrim<-uc_data_transform(data=df_x,
                               var_list=c("id_long","f.trt"),
                               var_list_new=c("id","trt"),
                               time="time_long",
                               prob="prob_long")
#nMI=10 used in the example below to reduce the time needed
#but a large number as nMI=1000 is recommended in practice

fit<-CoxMI(data_list=data_intrim,nMI=10,covariates=c("trt"))
CoxMI.summ(fit)


fit<-CoxMI(data_list=data_intrim,nMI=1000,covariates=c("trt"),id=c("id"))
CoxMI.summ(fit)
```

CoxMI.summ *Summary function for the Cox MI model*

Description

Prints the fitting results from the CoxMI function.

Usage

```
CoxMI.summ(x,digits=3)
```

Arguments

x	An object returned by the CoxMI function.
digits	Digits of output

Details

Print a summary table of Cox regression result with MI implemented.

Value

A summary table of Cox regression result with MI implemented.

Author(s)

Yiming Chen

See Also

CoxMI.

Coxwt *Weighted Cox PH model estimation*

Description

Estimate the Cox PH model by weighted partial likelihood. Event weights are calcualted with respect to event probabilities.

Usage

```
Coxwt(data_list,covariates,init=NULL,BS=FALSE,nBS=1000)
```

Arguments

data_list	The data list which has been transformed from the long format by the uc_data_transform function.
covariates	The vector of varaible on the RHS of the Cox model.
init	The initial value of covariates vector in the likelihood, length matches the length of covariates.
BS	T/F, whether conduct estimation via the Bootstrap method.
nBS	Number of BS, only effective if BS=TRUE.

Value

coefficients	Estimated vector of coefficients in the model
var	Estimated variance of the coefficients
hr	Estimated hazard ratios in the model
z	Wald test statistics
pvalue	Estimated two-sided p-value
coefficients_bs	Bootstrapped coefficient estimation
var_bs	Bootstrapped variance estimation
column_name	Column name

Author(s)

Yiming Chen, John Lawrence

References

[1]Cook TD. Adjusting survival analysis for the presence of unadjudicated study events. Controlled clinical trials. 2000;21(3):208-222.

[2]Cook TD, Kosorok MR. Analysis of time-to-event data with incomplete event adjudication. Journal of the american statistical association. 2004;99(468):1140-1152.

[3]Snapinn SM. Survival analysis with uncertain endpoints. Biometrics. 1998;54(1):209-218.

See Also

CoxMI, Coxwt.summ.

Examples

```
df_x<-data_sim(n=500,0.8,haz_c=0.5/365)
data_intrim<-uc_data_transform(data=df_x,
                               var_list=c("id_long","trt_long"),
                               var_list_new=c("id","trt"),
                               time="time_long",
                               prob="prob_long")
 fit<-Coxwt(data_list=data_intrim,covariates=c("trt"),init=c(1),BS=FALSE)
 Coxwt.summ(fit)

##an example if we would like to check the BS variance

fit2<-Coxwt(data_list=data_intrim,covariates=c("trt"),init=c(1),BS=TRUE, nBS = 100)
Coxwt.summ(fit2)
```

Coxwt.summ — *Summary function for the weighted Cox model*

Description

Print the fitting results from the weighted Cox regression.

Usage

```
Coxwt.summ(x,digits=3)
```

Arguments

x	An object returned by the Coxwt function
digits	Digits of output

Value

A summary table of weighted Cox regression result.

Author(s)

Yiming Chen

See Also

Coxwt, CoxMI.

data_sim — *Simulated survival data with uncertain endpoints from exponential distribution.*

Description

data_sim function simulates data from a hypothetic 1:1 two-arms clinical trial, with one year uniform accrual period and three years follow-up.

data_sim2 function simplifies data list generated from above function to a more events only case. Note this function is only used for demonstration purpose.

Usage

```
data_sim(n=200,true_hr=0.8,haz_c=1/365)
data_sim2(data_list,covariates,percentage)
```

Arguments

n	Total number of subject.
true_hr	True hazard ratio between trt and control.
haz_c	True event rate in the control arm.
data_list	The data list which has been transformed from the long format by uc_data_transform function.
covariates	The covariate we pose the true HR.
percentage	The percentage of censored subjects with potential events we would like to ultilize in the analysis. Ideally, with more potential events added, more power gain of imputation.

Value

Dataframe. Simulated datasets with event probabilities and potential event date.

Author(s)

Yiming Chen, John Lawrence

Examples

```
df_x<-data_sim(n=500,true_hr=0.8,haz_c=1/365)
data_intrim<-uc_data_transform(data=df_x,
                               var_list=c("id_long","trt_long"),
                               var_list_new=c("id","trt"),
                               time="time_long",
                               prob="prob_long")
df_y<-data_sim2(data_list=data_intrim,covariates=c("trt"),percentage=0.2)
```

KMMI *Kaplan-Meier estimation with event uncertainty*

Description

KM estimation for survival data when event uncertainty presents. KM plot will be output if plot=TRUE specfied.

Usage

```
KMMI(data_list,nMI,covariates,data_orig = NULL,plot = TRUE,
time_var=NULL,event_var=NULL)
```

Arguments

data_list	The data list which has been transformed from the long format by uc_data_transform function.
nMI	Number of imputations (>1). If missing, weighted statistics would be output instead.
covariates	The grouping varaible, no need to be factorized. If missing then the overall KM is returned.
plot	T/F, whether output a KM plot, the plot potentially contains KM curves from original dataset and imputed/weighted dataset.
data_orig	The original data without any uncertain events. If supplies then user can compare results from certain events only and all possible events.
time_var	Time variable in data_orig. If user provides the orig dataset then user need to specify the time and event indicator variable in the orignal dataset.
event_var	Event indicator variable in the original data set.

Value

KM_mi	A dataset contains MI estimation and variance at all potential event time
KM_cook	A dataset contains weighted KM estimation and variance at all potential event time
ngroup	Number of groups
cate_level	Values of the categorical variable
nMI	Number of imputed datasets

Author(s)

Yiming Chen

References

[1]Cook TD. Adjusting survival analysis for the presence of unadjudicated study events. Controlled clinical trials. 2000;21(3):208-222.

[2]Cook TD, Kosorok MR. Analysis of time-to-event data with incomplete event adjudication. Journal of the american statistical association. 2004;99(468):1140-1152.

[3]Klein JP, Moeschberger ML. Survival Analysis : Techniques for Censored and Truncated Data. New York: Springer; 1997.

[4]Rubin DB. Multiple Imputation for Nonresponse in Surveys. New York: Wiley; 1987

See Also

uc_data_transform

Examples

```
##an example with more potential event case
##data_orig was created as keeping the event with largest weights for individuals
df_x<-data_sim(n=500,0.8,haz_c=0.5/365)
data_intrim<-uc_data_transform(data=df_x,
                               var_list=c("id_long","trt_long"),
                               var_list_new=c("id","trt"),
                               time="time_long",
                               prob="prob_long")
df_y<-data_sim2(data_list=data_intrim,covariates=c("trt"),percentage=1)
data_orig<-df_y[df_y$prob==0|df_y$prob==1,]
data_orig<-data_orig[!duplicated(data_orig$id),]
data_orig$cens<-data_orig$prob


##weighted estimation
KM_res<-KMMI(data_list=data_intrim,nMI=NULL,covariates=c("trt"),plot=TRUE,data_orig=NULL)

##MI estimation
KMMI(data_list=data_intrim,nMI=1000,covariates=c("trt"),plot=TRUE,data_orig=NULL)

data_intrim2<-uc_data_transform(data=df_y, var_list=c("id","trt"),
                               var_list_new=NULL,time="time", prob="prob")

KMMI(data_list=data_intrim2,nMI=1000,covariates=c("trt"),plot=TRUE,data_orig=data_orig,
time_var=c("time"),event_var=c("cens"))
```

LRMI *Log-rank test with events uncertainty*

Description

This function conducts the Log-rank test with respect to uncertain endpoints, by MI or weighted method.

Usage

```
LRMI(data_list, nMI, covariates, strata = NULL,...)
```

Arguments

data_list	The data list which has been transformed from the long format by uc_data_transform function.
nMI	Number of imputation (>1). If missing, weighted statistics would be output instead.
covariates	The categorical variable used in the Log-rank test. No need to factorlize numeric variables.

strata	Strata variable may required by the Log-rank test
...	Other arguments passed on to survdiff().

Value

est	Estimated LR statistics, either from the MI method or weighted method
var	Estimated variance matrix
est_mat	Matrix containing estimate of statistics from each imputed dataset
Var_mat	Array containing variances for each imputed dataset
Between Var	Between imputation variance
Within Var	Mean within imputed dataset variance
nMI	Number of imputed datasets
pvalue	Estimated two-sided Chi-square test p-value
df	Degree of freedom
covariates	covariates
ngroup	Number of groups
obsmean	Mean of observed events count across imputations
expmean	Mean of expected events count across imputations

Author(s)

Yiming Chen

References

[1]Cook TD. Adjusting survival analysis for the presence of unadjudicated study events. Controlled clinical trials. 2000;21(3):208-222.

[2]Cook TD, Kosorok MR. Analysis of time-to-event data with incomplete event adjudication. Journal of the american statistical association. 2004;99(468):1140-1152.

[3]Klein JP, Moeschberger ML. Survival Analysis : Techniques for Censored and Truncated Data. New York: Springer; 1997.

[4]Rubin DB. Multiple Imputation for Nonresponse in Surveys. New York: Wiley; 1987

See Also

uc_data_transform, LRMI.summ

Examples

```
df_x<-data_sim(n=500,0.8,haz_c=0.5/365)
data_intrim<-uc_data_transform(data=df_x,
                               var_list=c("id_long","trt_long"),
                               var_list_new=c("id","trt"),
                               time="time_long",
                               prob="prob_long")
```

```
#nMI=10 used in the example below to reduce the time needed
#but a large number as nMI=1000 is recommended in practice
fit<-LRMI(data_list=data_intrim,nMI=10,covariates=c("trt"),strata=NULL)
LRMI.summ(fit)
```

`LRMI.summ` *Prints the test results output by the LRMI function*

Description

Summary function for the Log-rank test either by the MI method or the weighted method.

Usage

```
LRMI.summ(x,digits=3)
```

Arguments

`x`	An object returned by the LRMI function.
`digits`	Digits of output

Value

A summary table of LR test result with MI implemented.

Author(s)

Yiming Chen

See Also

`LRMI`

`uc_data_transform` *Transform long formatted time-to-event data into a data list*

Description

This function transforms data from long format (one record per event) to a datalist with length as unique subject number. The transformation is required before fitting other models from the package.

Usage

```
uc_data_transform(data,var_list,var_list_new,time,prob)
```

Arguments

data	The dataset in long format with a row for each potential event. For ceonsoring record, the event prob should be 0. It should include id, time and prob variables at a minimum. If any covariates are included in the call to the function, then these variables should also be included. A censoring record is required for each subject. Categorical variables need to be encoded as factor varaible before transformationif they are expected to be in the Cox model.
var_list	The list of identification variables, such as: c("id_long","trt_long").
time	The time variable need to be transofirmed, e.g. time_long.
prob	The prob variable need to be transformed, e.g. prob_long.
var_list_new	The character vector contains the new names for the id variables defined in the var_list, if missing, previous variable names would be used.

Value

time	The list of all potential event time
prob	The list of all potential event probabilities
weights	The list of all potential event weights
e	The list of individual potential event count
s	The list of all survival probabilities
data_uc	The dataset contains unique information of each subject
data_long	The dataset contains the original data in long format

Author(s)

Yiming Chen

Examples

```
df_x<-data_sim(n=1000,true_hr=0.8,haz_c=0.5/365)
df_x$f.trt<-as.factor(df_x$trt_long)
data_intrim<-uc_data_transform(data=df_x,
                               var_list=c("id_long","f.trt"),
                               var_list_new=c("id","trt"),
                               time="time_long",
                               prob="prob_long")
```

Index