

Package ‘NestedCohort’

February 19, 2015

Version 1.1-3

Date 2012-12-21

Title Survival Analysis for Cohorts with Missing Covariate Information

Author Hormuzd A. Katki <katkih@mail.nih.gov>

Maintainer Hormuzd A. Katki <katkih@mail.nih.gov>

Depends R (>= 2.15.2), survival, MASS

Description Estimate hazard ratios, survival curves and attributable risks for cohorts with missing covariates, using Cox models or Kaplan-Meier estimated for strata. This handles studies nested within cohorts, such as case-cohort studies with stratified sampling. See http://www.r-project.org/doc/Rnews/Rnews_2008-1.pdf

License GPL (>= 2)

URL <http://dceg.cancer.gov/about/staff-bios/katki-hormuzd>

Repository CRAN

Date/Publication 2013-02-07 20:16:59

NeedsCompilation no

R topics documented:

NestedCohort-package	2
nested.coxph	3
nested.km	6
nested.stdsurv	8
zinc	11

Index	12
--------------	-----------

NestedCohort-package *Survival Analysis of Cohort Studies With Missing Covariate Information* ~~ NestedCohort ~~

Description

NestedCohort fits Kaplan-Meier and Cox Models when you have missing covariate or strata data on a sample of a cohort. Missingness can be either by happenstance or by design (for example, the case-cohort and case-control within cohort designs). NestedCohort estimates standardized survival, survival differences, and attributable risks.

Details

Package: NestedCohort
Type: Package
Version: 1.1-3
Date: 2012-12-12
License: GPL (>= 2)

To fit Kaplan-Meier, use `nested.km()`. If you only want hazard ratios from a Cox model, used `nested.coxph()`. If you want standardized survival and attributable risk estimates, used `nested.stdsurv()`.

Author(s)

Author: Hormuzd A. Katki

Maintainer: Hormuzd A. Katki <katkih@mail.nih.gov>

References

Katki, H.A. and Mark, S.D. Survival Analysis for Cohorts with Missing Covariate Information. *The R Journal*, 2008, 8(1), 14-9.

Mark, S.D. and Katki, H.A. Specifying and Implementing Nonparametric and Semiparametric Survival Estimators in Two-Stage (sampled) Cohort Studies with Missing Case Data. *Journal of the American Statistical Association*, 2006, 101, 460-471.

See Also

The survival package and the survey package

Examples

```
# Get zinc dataset
data(zinc)
```

```

# Fit and plot Kaplan-Meier
mod <- nested.km(survfitformula="Surv(futime01,ec01==1)~znquartiles",
                 samplingmod="ec01*basehist",data=zinc)
plot(mod,ymin=.6,xlab="Time (Days)",ylab="Survival",main="Survival by Quartile of Zinc",lty=1:4,)
legend(2000,0.7,c("Q1","Q2","Q3","Q4"),lty=1:4)

# Fit Cox model, get hazard ratios
coxmod <- nested.coxph(coxformula="Surv(futime01,ec01==1)~
                       sex+agepill+smoke+drink+mildysp+moddysp+sevdysp+anyhist+zncent",
                       samplingmod="ec01*basehist",data=zinc)
summary(coxmod)

# Fit Cox model, get and plot standardized survivals, survival differences, and attributable risks
mod <- nested.stdsurv(outcome="Surv(futime01,ec01==1)",
                     exposures="znquartiles",
                     confounders="sex+agestr+smoke+drink+mildysp+moddysp+sevdysp+anyhist",
                     samplingmod="ec01*basehist",exposureofinterest="Q4",plot=TRUE,
                     main="Time to Esophageal Cancer by Quartiles of Zinc",data=zinc)

```

 nested.coxph

Estimate Cox model hazard ratios for covariates with missing data

Description

nested.coxph fits the Cox model to estimate hazard ratios for covariates that are missing data on some cohort members. All covariates may be continuous or categorical. nested.coxph requires knowledge of the variables that missingness depends on, with missingness probability modeled through a [glm](#) sampling model. Often, the data is in the form of a case-control sample taken within a cohort. nested.coxph allows cases to have missing data, and can extract efficiency from auxiliary variables by including them in the sampling model. nested.coxph requires [coxph](#) from the survival package.

Usage

```

nested.coxph(coxformula, samplingmod, data, outputsamplingmod = FALSE,
             glmLink = binomial(link = "logit"),
             glmcontrol = glm.control(epsilon = 1e-10, maxit = 10, trace = FALSE),
             coxphcontrol = coxph.control(eps = 1e-10, iter.max = 50),
             missvarwarn = TRUE, ...)

```

Arguments

Required arguments:

Standard coxph formula

samplingmod Right side of the formula for the [glm](#) sampling model that models the probability of missingness

data	Data Frame that all variables are in
	Optional arguments:
outputsamplingmod	Output the sampling model, default is false
glmLink	Sampling model link function, default is logistic regression
glmcontrol	See glm.control
coxphcontrol	See coxph.control
missvarwarn	Warn if there is missing data in the sampling variable. Default is TRUE
...	Any additional arguments to be passed on to glm or coxph

Details

If `nested.coxph` reports that the sampling model "failed to converge", the sampling model will be returned for your inspection. Note that if some sampling probabilities are estimated at 1, the model technically cannot converge, but you get very close to 1, and `nested.coxph` will not report non-convergence for this situation.

Note these issues. The data must be in a dataframe and specified in the data statement. No variable can be named 'o.b.s.e.r.v.e.d.' or 'p.i.h.a.t.'. Cases and controls cannot be finely matched on time, but matching on time within large strata is allowed. `cluster()` statements are not allowed in `coxformula`. Allows left truncation, staggered entry, open cohorts, and stratified baseline hazards. Must use Breslow Tie-Breaking.

Value

If `outputsamplingmod=FALSE`, the output are the hazard ratios and the `coxph` model. Any method for `coxph` objects will work for this so long as that method only requires consistent estimates of the parameters and their standard errors. If `outputsamplingmod=TRUE`, then the sampling model is also returned, and the output is a list with components:

<code>coxmod</code>	The Cox model of class <code>coxph</code>
<code>samplingmod</code>	The sampling model of class <code>glm</code>

Note

Requires the MASS library from the VR bundle that is available from the CRAN website.

Author(s)

Hormuzd A. Katki

References

Katki HA, Mark SD. Survival Analysis for Cohorts with Missing Covariate Information. R-News, 8(1) 14-9, 2008. http://www.r-project.org/doc/Rnews/Rnews_2008-1.pdf

Mark, S.D. and Katki, H.A. Specifying and Implementing Nonparametric and Semiparametric Survival Estimators in Two-Stage (sampled) Cohort Studies with Missing Case Data. Journal of the American Statistical Association, 2006, 101, 460-471.

Mark SD, Katki H. Influence function based variance estimation and missing data issues in case-cohort studies. *Lifetime Data Analysis*, 2001; 7; 329-342

Christian C. Abnet, Barry Lai, You-Lin Qiao, Stefan Vogt, Xian-Mao Luo, Philip R. Taylor, Zhi-Wei Dong, Steven D. Mark, Sanford M. Dawsey. Zinc concentration in esophageal biopsies measured by X-ray fluorescence and cancer risk. To Appear in *Journal of the National Cancer Institute*.

See Also

See Also: [nested.stdsurv](#), [zinc](#), [nested.km](#), [coxph](#), [glm](#)

Examples

```
## Simple analysis of zinc and esophageal cancer data:
## We sampled zinc (variable zncent) on a fraction of the subjects, with
## sampling fractions depending on cancer status and baseline histology.
## We observed the confounding variables on almost all subjects.
data(zinc)
coxmod <- nested.coxph(coxformula="Surv(futime01,ec01==1)~
  sex+agepill+smoke+drink+milddysp+moddysp+sevdysp+anyhist+zncent",
  samplingmod="ec01*basehist",data=zinc)
summary(coxmod)

# This is the output:
# Call:
# coxph(formula = as.formula(coxformula), data = data, weights = 1/p.i.h.a.t.,
# subset = TRUE, na.action = na.omit, control = coxphcontrol,
# x = TRUE, method = "breslow")

# n= 123, number of events= 56
# (308 observations deleted due to missingness)

#               coef exp(coef) se(coef)      z Pr(>|z|)
# sexMale         0.2953   1.3436  0.5558  0.53  0.5952
# agepill         0.0539   1.0554  0.0275  1.96  0.0499 *
# smokeEver       0.0145   1.0146  0.5870  0.02  0.9803
# drinkEver      -0.8548   0.4254  0.5896 -1.45  0.1471
# milddyspMild Dysplasia 0.9023   2.4653  0.3937  2.29  0.0219 *
# moddyspModerate Dysplasia 1.3309   3.7845  0.4212  3.16  0.0016 **
# sevdyspSevere Dysplasia 2.1334   8.4439  0.4615  4.62  3.8e-06 ***
# anyhistFamily History  0.0904   1.0946  0.3896  0.23  0.8165
# zncent         -0.2498   0.7789  0.1351 -1.85  0.0645 .

#               exp(coef) exp(-coef) lower .95 upper .95
# sexMale         1.344      0.744    0.452    3.99
# agepill         1.055      0.948    1.000    1.11
# smokeEver       1.015      0.986    0.321    3.21
# drinkEver       0.425      2.351    0.134    1.35
# milddyspMild Dysplasia 2.465      0.406    1.140    5.33
# moddyspModerate Dysplasia 3.784      0.264    1.658    8.64
# sevdyspSevere Dysplasia 8.444      0.118    3.417   20.86
# anyhistFamily History  1.095      0.914    0.510    2.35
# zncent          0.779      1.284    0.598    1.02
```

```
# Concordance= NA (se = NA )
# Rsquare= NA (max possible= NA )
# Likelihood ratio test= NA on 9 df, p=NA
# Wald test          = 65.1 on 9 df, p=1.36e-10
# Score (logrank) test = NA on 9 df, p=NA
```

nested.km	<i>Estimate non-parametric survival curves for each level of categorical variables with missing data.</i>
-----------	---

Description

The function `nested.km` gives non-parametric survival curve estimates (like Kaplan-Meier) for each level of categorical variables that have missing data on some cohort members. These variables must be factor variables. `nested.km` requires knowledge of the variables that missingness depends on, with missingness probability modeled through a `glm` sampling model. Often, the data is in the form of a case-control sample taken within a cohort. `nested.km` allows cases to have missing data, and can extract efficiency from auxiliary variables by including them in the sampling model. `nested.km` makes heavy use of the `survfit` function in the survival package.

Usage

```
nested.km(survfitformula, samplingmod, data, outputsamplingmod=FALSE,
          outputriskdiff = FALSE, exposureofinterest = "",
          timeofinterest = Inf, glmLink = binomial(link = "logit"),
          glmcontrol = glm.control(epsilon = 1e-10, maxit = 10, trace = FALSE),
          missvarwarn = TRUE, ...)
```

Arguments

	Required arguments:
	Legal formula for a <code>survfit</code> object
<code>samplingmod</code>	Right side of the formula for the <code>glm</code> sampling model that models the probability of missingness
<code>data</code>	Data Frame that all variables are in
	Optional arguments:
<code>outputsamplingmod</code>	Output the sampling model, default is false
<code>outputriskdiff</code>	Output risk differences, default is false
<code>exposureofinterest</code>	Exposure level to make risk differences with respect to
<code>timeofinterest</code>	Time at which to report risk differences, default is end of followup
<code>glmLink</code>	Sampling model link function, default is logistic regression
<code>glmcontrol</code>	See <code>glm.control</code>
<code>missvarwarn</code>	Warn if there is missing data in the sampling variable. Default is TRUE
<code>...</code>	Any additional arguments to be passed on to <code>survfit</code>

Details

nested.km provides survival estimates that are not standardized for confounders nor account for competing risks.

If nested.km reports that the sampling model "failed to converge", the sampling model will be returned for your inspection. Note that if some sampling probabilities are estimated at 1, the model technically cannot converge, but you get very close to 1, and nested.km will not report non-convergence for this situation.

Note these issues. The data must be in a dataframe and specified in the data statement. No variable in the dataframe can be named 'o.b.s.e.r.v.e.d.' or 'p.i.h.a.t.'. Cases and controls cannot be finely matched on time, but matching on time within large strata is allowed. Everyone must enter the cohort at the same time on the vital time scale. All covariates in the survfitformula must be factor even if binary. Never use '*' to mean interaction in the survfitformula, instead use [interaction](#).

Value

If outputpropmod=F, the output is the survival curves in the survfit model. Any method that works for survfit objects will work for this so long as the method only requires consistent estimates of the parameters and their standard errors. If outputpropmod=T, then the sampling model is also returned, and the output is a list with components:

survmod	The survfit model of class survfit
propmod	The sampling model of class glm

Note

Requires the MASS library from the VR bundle that is available from the CRAN website.

Author(s)

Hormuzd A. Katki

References

Katki HA, Mark SD. Survival Analysis for Cohorts with Missing Covariate Information. R-News, 8(1) 14-9, 2008. http://www.r-project.org/doc/Rnews/Rnews_2008-1.pdf

Mark, S.D. and Katki, H.A. Specifying and Implementing Nonparametric and Semiparametric Survival Estimators in Two-Stage (sampled) Cohort Studies with Missing Case Data. Journal of the American Statistical Association, 2006, 101, 460-471.

Mark SD, Katki H. Influence function based variance estimation and missing data issues in case-cohort studies. Lifetime Data Analysis, 2001; 7; 329-342

Christian C. Abnet, Barry Lai, You-Lin Qiao, Stefan Vogt, Xian-Mao Luo, Philip R. Taylor, Zhi-Wei Dong, Steven D. Mark, Sanford M. Dawsey. Zinc concentration in esophageal biopsies measured by X-ray fluorescence and cancer risk. To Appear in Journal of the National Cancer Institute.

See Also

See Also: [nested.stdsurv](#), [zinc](#), [nested.coxph](#), [coxph](#), [glm](#)

Examples

```
## Simple analysis of zinc and esophageal cancer data:
## We sampled zinc (variable znquartiles) on a fraction of the subjects, with
## sampling fractions depending on cancer status and baseline histology.
## We observed the confounding variables on almost all subjects.
data(zinc)
mod <- nested.km(survfitformula="Surv(futime01,ec01==1)~znquartiles",
                 samplingmod="ec01*basehist",data=zinc)

# This is the output
# Risk Differences vs. znquartiles=Q1 by time Inf
#      Risk Difference StdErr 95% CI Left 95% CI Right
# Q1 - Q2      -0.2262 0.1100   -0.4419   -0.01060
# Q1 - Q3      -0.1749 0.1145   -0.3993    0.04945
# Q1 - Q4      -0.2818 0.1042   -0.4859   -0.07760

plot(mod,ymin=.6,xlab="Time (Days)",ylab="Survival",main="Survival by Quartile of Zinc",lty=1:4,)
legend(2000,0.7,c("Q1","Q2","Q3","Q4"),lty=1:4)
```

nested.stdsurv

Estimate Standardized Survivals and Attributable Risks for covariates with missing data

Description

The function `nested.stdsurv` fits the Cox model to estimate standardized survival curves and attributable risks for covariates that are missing data on some cohort members. All covariates must be factor variables. `nested.stdsurv` requires knowledge of the variables that missingness depends on, with missingness probability modeled through a `glm` sampling model. Often, the data is in the form of a case-control sample taken within a cohort. `nested.stdsurv` allows cases to have missing data, and can extract efficiency from auxiliary variables by including them in the sampling model. `nested.stdsurv` requires `coxph` from the survival package.

Usage

```
nested.stdsurv(outcome, exposures, confounders, samplingmod, data,
              exposureofinterest = "", timeofinterest = Inf,cuminc=FALSE,
              plot = FALSE, plotfilename = "", glmLink = binomial(link = "logit"),
              glmcontrol = glm.control(epsilon = 1e-10, maxit = 10, trace = FALSE),
              coxphcontrol = coxph.control(eps = 1e-10, iter.max = 50),
              missvarwarn = TRUE, ...)
```

Arguments

Required arguments:

Survival outcome of interest, must be a `Surv` object

exposures	The part of the right side of the Cox model that parameterizes the exposures. Never use '*' for interaction, use interaction . Survival probabilities will be computed for each level of the exposures.
confounders	The part of the right side of the Cox model that parameterizes the confounders. Never use '*' for interaction, use interaction .
samplingmod	Right side of the formula for the glm sampling model that models the probability of missingness
data	Data Frame that all variables are in Optional arguments:
exposureofinterest	The name of the level of the exposures for which attributable risk is desired. Default is the first level of the exposure.
timeofinterest	The time at which survival probabilities and attributable risks are desired. Default is the last event time.
cuminc	Set to T if you want output as cumulative incidence, F for survival
plot	If T, plot the standardized survivals. Default is F.
plotfilename	A string for the filename to save the plot as
glmLink	Sampling model link function, default is logistic regression
glmcontrol	See glm.control
coxphcontrol	See coxph.control
missvarwarn	Warn if there is missing data in the sampling variable. Default is TRUE
...	Any additional arguments to be passed on to glm or coxph

Details

If `nested.stdsurv` reports that the sampling model "failed to converge", the sampling model will be returned for your inspection. Note that if some sampling probabilities are estimated at 1, the model technically cannot converge, but you get very close to 1, and `nested.stdsurv` will not report non-convergence for this situation.

Note the following issues.

The data must be in a dataframe and specified in the data statement. No variable can be named 'o.b.s.e.r.v.e.d.' or 'p.i.h.a.t.'. Cases and controls cannot be finely matched on time, but matching on time within large strata is allowed. `strata()`, `cluster()` or `offset()` statements in or confounders are not allowed. Everyone must enter the cohort at the same time on the vival time scale. Must use Breslow Tie-Breaking. All covariates must be factor variables, even if binary. Do not use '*' to mean interaction in exposures or confounders, use [interaction](#).

Value

A List with the following components:

coxmod	The fitted Cox model
samplingmod	The fitted glm sampling model
survtable	Standardized survival (and inference) for each exposure level

riskdifftable Standardized survival (risk) differences (and inference) for each exposure level, relative to the exposure of interest.

PARTable Population Attributable Risk (and inference) for the exposure of interest

If plot=T, then the additional component is included:

plotdata A matrix with data needed to plot the survivals: time, standardized survival for each exposure level, and crude survival. Name of each exposure level is converted to a proper R variable name (these are the column labels).

Note

Requires the MASS library from the VR bundle that is available from the CRAN website.

Author(s)

Hormuzd A. Katki

References

Katki HA, Mark SD. Survival Analysis for Cohorts with Missing Covariate Information. R-News, 8(1) 14-9, 2008. http://www.r-project.org/doc/Rnews/Rnews_2008-1.pdf

Mark, S.D. and Katki, H.A. Specifying and Implementing Nonparametric and Semiparametric Survival Estimators in Two-Stage (sampled) Cohort Studies with Missing Case Data. Journal of the American Statistical Association, 2006, 101, 460-471.

Mark SD, Katki H. Influence function based variance estimation and missing data issues in case-cohort studies. Lifetime Data Analysis, 2001; 7; 329-342

Christian C. Abnet, Barry Lai, You-Lin Qiao, Stefan Vogt, Xian-Mao Luo, Philip R. Taylor, Zhi-Wei Dong, Steven D. Mark, Sanford M. Dawsey. Zinc concentration in esophageal biopsies measured by X-ray fluorescence and cancer risk. Journal of the National Cancer Institute, 2005; 97(4) 301-306

See Also

See Also: [nested.coxph](#), [zinc](#), [nested.km](#), [coxph](#), [glm](#)

Examples

```
## Simple analysis of zinc and esophageal cancer data:
## We sampled zinc (variable znquartiles) on a fraction of the subjects, with
## sampling fractions depending on cancer status and baseline histology.
## We observed the confounding variables on almost all subjects.
data(zinc)
mod <- nested.stdsurv(outcome="Surv(futime01,ec01==1)",
                      exposures="znquartiles",
                      confounders="sex+agestr+smoke+drink+mildysp+moddysp+sevdysp+anyhist",
                      samplingmod="ec01*basehist",exposureofinterest="Q4",data=zinc)

# This is the output:
# Standardized Survival for znquartiles by time 5893
#      Survival StdErr 95% CI Left 95% CI Right
```

```

# Q1      0.5443 0.07232      0.3932      0.6727
# Q2      0.7595 0.07286      0.5799      0.8703
# Q3      0.7045 0.07174      0.5383      0.8203
# Q4      0.8911 0.06203      0.6863      0.9653
# Crude   0.7784 0.02491      0.7249      0.8228

# Standardized Risk Differences vs. znquartiles = Q4 by time 5893
#           Risk Difference StdErr 95% CI Left 95% CI Right
# Q4 - Q1      0.3468 0.10376   0.143412   0.5502
# Q4 - Q2      0.1316 0.09605  -0.056694   0.3198
# Q4 - Q3      0.1866 0.09355   0.003196   0.3699
# Q4 - Crude   0.1126 0.06353  -0.011871   0.2372

# PAR if everyone had znquartiles = Q4
# Estimate StdErr 95% PAR CI Left 95% PAR CI Right
# PAR      0.5084 0.2777      -0.4872      0.8375

```

zinc

*Example Study Nested within a Cohort: Zinc and Esophageal Cancer***Description**

zinc is a data frame, with some variables observed on all subjects, and some variables not. The outcome variable is ec01, indicating esophageal cancer or not at time futime01. The zinc variables are in zncent (continuous), znqt (ordinal zinc quartiles), or znquartiles (factor variable notating quartiles of zinc)

Usage

```
data(zinc)
```

Format

See [nested.coxph](#) for example of using this dataset.

Source

Steven D. Mark

References

Christian C. Abnet, Barry Lai, You-Lin Qiao, Stefan Vogt, Xian-Mao Luo, Philip R. Taylor, Zhi-Wei Dong, Steven D. Mark, Sanford M. Dawsey. Zinc concentration in esophageal biopsies measured by X-ray fluorescence and cancer risk. *Journal of the National Cancer Institute*, 2005; 97(4) 301-306

Index

*Topic **datasets**

zinc, [11](#)

*Topic **models**

nested.coxph, [3](#)

nested.km, [6](#)

nested.stdsurv, [8](#)

*Topic **survey**

NestedCohort-package, [2](#)

*Topic **survival**

NestedCohort-package, [2](#)

coxph, [3](#), [5](#), [7](#), [8](#), [10](#)

coxph.control, [4](#), [9](#)

glm, [3](#), [5–8](#), [10](#)

glm.control, [4](#), [6](#), [9](#)

interaction, [7](#), [9](#)

nested.coxph, [3](#), [7](#), [10](#), [11](#)

nested.km, [5](#), [6](#), [10](#)

nested.stdsurv, [5](#), [7](#), [8](#)

NestedCohort (NestedCohort-package), [2](#)

NestedCohort-package, [2](#)

Surv, [8](#)

survfit, [6](#)

zinc, [5](#), [7](#), [10](#), [11](#)