

Package ‘timereg’

January 2, 2012

Title timereg package for Flexible regression models for survival data.

Version 1.6-0

Date 2011-17-11

Author Thomas Scheike with contributions from Torben Martinussen and Jeremy Silver

Maintainer Thomas Scheike <ts@biostat.ku.dk>

Description Programs for Martinussen and Scheike (2006), ‘Dynamic Regression Models for Survival Data’, Springer Verlag. Plus more recent developments. Additive survival model, semi-parametric proportional odds model, cumulative residuals, excess risk models and more. Flexible competing risks regression including GOF-tests. Two-stage frailty modelling. PLS for the additive risk model. Lasso in ahaz package.

Depends R (>= 2.10), survival

LazyLoad yes

License GPL (>= 2)

Repository CRAN

Repository/R-Forge/Project timereg

Repository/R-Forge/Revision 84

Date/Publication 2011-11-21 08:06:26

R topics documented:

aalen	2
additive.pls	5
bmt	7
cd4	8
comp.risk	9
const	14
cox	14

cox.aalen	14
csl	18
cum.residuals	19
diabetes	20
dynreg	21
Gprop.odds	24
krylow.pls	27
mela.pop	28
melanoma	29
pe.sasieni	30
plot.aalen	31
plot.cum.residuals	32
plot.dynreg	34
predict.timereg	35
print.aalen	37
prop	38
prop.excess	38
prop.odds	40
pval	43
summary.aalen	43
summary.cum.residuals	44
timecox	44
TRACE	47
two.stage	48

Index **51**

aalen	<i>Fit additive hazards model</i>
-------	-----------------------------------

Description

Fits both the additive hazards model of Aalen and the semi-parametric additive hazards model of McKeague and Sasieni. Estimates are un-weighted. Time dependent variables and counting process data (multiple events per subject) are possible.

Resampling is used for computing p-values for tests of time-varying effects.

The modelling formula uses the standard survival modelling given in the **survival** package.

Usage

```
aalen(formula,data=sys.parent(),start.time=0,max.time=NULL,robust=1,
id=NULL,clusters=NULL,residuals=0,n.sim=1000,weighted.test=0,
covariance=0,resample.iid=0,deltaweight=1,silent=1,weights=NULL,
max.clust=1000,gamma=NULL,offsets=0)
```

Arguments

<code>formula</code>	a formula object with the response on the left of a '~' operator, and the independent terms on the right as regressors. The response must be a survival object as returned by the 'Surv' function. Time- invariant regressors are specified by the wrapper <code>const()</code> , and cluster variables (for computing robust variances) by the wrapper <code>cluster()</code> .
<code>data</code>	a <code>data.frame</code> with the variables.
<code>start.time</code>	start of observation period where estimates are computed.
<code>max.time</code>	end of observation period where estimates are computed. Estimates thus computed from <code>[start.time, max.time]</code> . Default is max of data.
<code>robust</code>	to compute robust variances and construct processes for resampling. May be set to 0 to save memory.
<code>id</code>	For timevarying covariates the variable must associate each record with the id of a subject.
<code>clusters</code>	cluster variable for computation of robust variances.
<code>n.sim</code>	number of simulations in resampling.
<code>weighted.test</code>	to compute a variance weighted version of the test-processes used for testing time-varying effects.
<code>residuals</code>	to returns residuals that can be used for model validation in the function <code>cum.residuals</code>
<code>covariance</code>	to compute covariance estimates for nonparametric terms rather than just the variances.
<code>resample.iid</code>	to return i.i.d. representation for nonparametric and parametric terms.
<code>deltaweight</code>	uses weights to estimate semiparametric model, under construction, default=1 is standard least squares estimates
<code>silent</code>	set to 0 to print warnings for non-inverible design-matrices for different time-points, default is 1.
<code>weights</code>	weights for estimating equations.
<code>max.clust</code>	sets the total number of i.i.d. terms in i.i.d. decomposition. This can limit the amount of memory used by coarsening the clusters. When NULL then all clusters are used. Default is 1000 to save memory and time.
<code>gamma</code>	fixes gamma at this value for estimation.
<code>offsets</code>	offsets for the additive model, to make excess risk modelling.

Details

The data for a subject is presented as multiple rows or 'observations', each of which applies to an interval of observation (`start, stop`]. For counting process data with the `(start,stop]` notation is used the 'id' variable is needed to identify the records for each subject. The program assumes that there are no ties, and if such are present random noise is added to break the ties.

Value

returns an object of type "aalen". With the following arguments:

cum	cumulative timevarying regression coefficient estimates are computed within the estimation interval.
var.cum	the martingale based pointwise variance estimates for cumulatives.
robvar.cum	robust pointwise variances estimates for cumulatives.
gamma	estimate of parametric components of model.
var.gamma	variance for gamma.
robvar.gamma	robust variance for gamma.
residuals	list with residuals. Estimated martingale increments (dM) and corresponding time vector (time).
obs.testBeq0	observed absolute value of supremum of cumulative components scaled with the variance.
pval.testBeq0	p-value for covariate effects based on supremum test.
sim.testBeq0	resampled supremum values.
obs.testBeqC	observed absolute value of supremum of difference between observed cumulative process and estimate under null of constant effect.
pval.testBeqC	p-value based on resampling.
sim.testBeqC	resampled supremum values.
obs.testBeqC.is	observed integrated squared differences between observed cumulative and estimate under null of constant effect.
pval.testBeqC.is	p-value based on resampling.
sim.testBeqC.is	resampled supremum values.
conf.band	resampling based constant to construct robust 95% uniform confidence bands.
test.procBeqC	observed test-process of difference between observed cumulative process and estimate under null of constant effect over time.
sim.test.procBeqC	list of 50 random realizations of test-processes under null based on resampling.
covariance	covariances for nonparametric terms of model.
B.iid	Resample processes for nonparametric terms of model.
gamma.iid	Resample processes for parametric terms of model.
deviance	Least squares of increments.

Author(s)

Thomas Scheike

References

Martinussen and Scheike, Dynamic Regression Models for Survival Data, Springer (2006).

Examples

```

library(survival)
data(sTRACE)
# Fits Aalen model
out<-aalen(Surv(time,status==9)~age+sex+diabetes+chf+vf,
sTRACE,max.time=7,n.sim=100)

summary(out)
par(mfrow=c(2,3))
plot(out)

# Fits semi-parametric additive hazards model
out<-aalen(Surv(time,status==9)~const(age)+const(sex)+const(diabetes)+chf+vf,
sTRACE,max.time=7,n.sim=100)

summary(out)
par(mfrow=c(2,3))
plot(out)

## Excess risk additive modelling
data(mela.pop)
dummy<-rnorm(nrow(mela.pop));

# Fits Aalen model with offsets
out<-aalen(Surv(start,stop,status==1)~age+sex+const(dummy),
mela.pop,max.time=7,n.sim=100,offsets=mela.pop$rate,id=mela.pop$id,
gamma=0)
summary(out)
par(mfrow=c(2,3))
plot(out,main="Additive excess risks model")

# Fits semi-parametric additive hazards model with offsets
out<-aalen(Surv(start,stop,status==1)~age+const(sex),
mela.pop,max.time=7,n.sim=100,offsets=mela.pop$rate,id=mela.pop$id)
summary(out)
plot(out,main="Additive excess risks model")

```

additive.pls

Fits PLS for additive hazards model

Description

Fits the partial least squares estimator for the additive risk model. Time dependent variables and counting process data (multiple events per subject) are possible.

The modelling formula uses the standard survival modelling given in the **survival** package.

Covariates Z_1, \dots, Z_p , fixed covariates $F(t)$. Algorithm : 1) For pls components X_1, \dots, X_K , fits

$$\lambda_0(t) + \alpha^T(t)F(t) + \sum_{j=1}^K X_j(t)\gamma_j(t) + Z_i(t)\beta_i^{K+1}$$

for $i = 1, \dots, p$ 2) compute new pls components $X_{K+1} = \sum \beta_i^{K+1} Z_i(t)$ and iterate.

Usage

```
additive.pls(formula = formula(data), data = sys.parent(),
start.time=0,max.time=NULL,id=NULL, pls.dim=1,
silent=1)
```

Arguments

formula	a formula object with the response on the left of a '~' operator, and the independent terms on the right as regressors. The response must be a survival object as returned by the 'Surv' function. The const terms are kept as fixed covariates that are not involved in the pls variable reduction. This may be covariates that are know to be of clinical importance.
data	a data.frame with the variables.
start.time	start of observation period where estimates are computed.
max.time	end of observation period where estimates are computed. Estimates thus computed from [start.time, max.time]. Default is max of data.
id	For timevarying covariates the variable must associate each record with the id of a subject.
pls.dim	number of pls components
silent	set to 1 to avoid printing of warnings for non-inverible design-matrices for different timepoints, default is 0.

Details

const() specifies the $F(t)$ covariates in the above model.

Value

returns an object with the following arguments:

baseline	baseline of the semparametric additive risk model
pls.comp	the pls components, i.e. the covariates multiplied on the beta coefficients.
beta	risk regression coefficients related to the pls components.
beta.pls	regression coefficients that defines the pls componets.
tbeta.pls	the combined regression coefficients from the pls components and the regression coefficients, these leads to risk predictions when applied to new covariates.

Author(s)

Thomas Scheike

References

- Martinussen and Scheike, Dynamic Regression Models for Survival Data, Springer (2006).
- Martinussen and Scheike, The Aalen additive hazards model with high-dimensional regressors, (2009), Lifetime Data Anal.

Examples

```

data(mypbc)
pbc<-mypbc
pbc$time<-pbc$time+runif(418)*0.1; pbc$time<-pbc$time/365
pbc<-subset(pbc,complete.cases(pbc));
covs<-as.matrix(pbc[, -c(1:3,6)])
covs<-cbind(covs[,c(1:6,16)],log(covs[,7:15]))
covs<-scale(covs);

### 5 PLS components for the 16 covariates
### based on Gui-Li approach with constant effects
out<-additive.pls(Surv(time,status>=1)~covs,pbc,max.time=7,pls.dim=5)

### survival predictions
par(mfrow=c(1,3))
pout=predict.pls(out,Z=covs)
matplot(pout$times,t(pout$surv[1:20,]),type="l",ylim=c(0,1),xlab="time",ylab="survival")

### cross-validated number of pls components

outcv<-pls.surv.cv(Surv(time,status>=1)~covs,pbc,max.time=7,pls.dims=0:2)
plot(outcv$pls.dims,outcv$cv,type="l")

out2<-additive.pls(Surv(time,status>=1)~covs,pbc,max.time=7,
pls.dim=outcv$cv.dim)

pout=predict.pls(out2, Z= covs)
matplot(pout$times,t(pout$surv[1:20,]),type="l",ylim=c(0,1),
xlab="time",ylab="survival")

### 2 PLS components for the 3 covariates
### age and sex are fixed covariates
data(sTRACE)
out<-additive.pls(Surv(time,status==9)~const(age)+const(sex)+
vf+diabetes+chf,sTRACE,max.time=7,pls.dim=2,silent=1)

```

Description

Bone marrow transplant data with 408 rows and 5 columns.

Format

The data has 408 rows and 5 columns.

cause a numeric vector code. Survival status. 1: dead from treatment related causes, 2: relapse , 0: censored.

time a numeric vector. Survival time.

platelet a numeric vector code. Platelet 1: more than 100×10^9 per L, 0: less.

tcell a numeric vector. T-cell depleted BMT 1:yes, 0:no.

age a numeric vector code. Age of patient, scaled and centered $((age-35)/15)$.

Source

Simulated data

References

NN

Examples

```
data(bmt)
names(bmt)
```

cd4

The multicenter AIDS cohort study

Description

CD4 counts collected over time.

Format

This data frame contains the following columns:

obs a numeric vector. Number of observations.

id a numeric vector. Id of subject.

visit a numeric vector. Timings of the visits in years.

smoke a numeric vector code. 0: non-smoker, 1: smoker.

age a numeric vector. Age of the patient at the start of the trial.

cd4 a numeric vector. CD4 percentage at the current visit.

cd4.prev a numeric vector. CD4 level at the preceding visit.

precd4 a numeric vector. Post-infection CD4 percentage.

lt a numeric vector. Gives the starting time for the time-intervals.

rt a numeric vector. Gives the stopping time for the time-interval.

Source

MACS Public Use Data Set Release PO4 (1984-1991). See reference.

References

Kaslow et al. (1987), The multicenter AIDS cohort study: rationale, organization and selected characteristics of the participants. *Am. J. Epidemiology* 126, 310–318.

Examples

```
data(cd4)
names(cd4)
```

comp.risk

Competing Risks Regression

Description

Fits a semiparametric model for the cause-specific quantities :

$$P(T < t, \text{cause} = 1 | x, z) = P_1(t, x, z) = h(g(t, x, z))$$

for a known link-function $h(\cdot)$ and known prediction-function $g(t, x, z)$ for the probability of dying from cause 1 in a situation with competing causes of death.

We consider the following models : 1) the additive model where $h(x) = 1 - \exp(-x)$ and

$$g(t, x, z) = x^T A(t) + (\text{diag}(t^p)z)^T \beta$$

2) the proportional setting that includes the Fine & Gray (FG) "prop" model and some extensions where $h(x) = 1 - \exp(-\exp(x))$ and

$$g(t, x, z) = \exp(x^T A(t) + (\text{diag}(t^p)z)^T \beta)$$

The FG model is obtained when $x = 1$.

3) a "logistic" model where $h(x) = \exp(x)/(1 + \exp(x))$ and

$$g(t, x, z) = x^T A(t) + (\text{diag}(t^p)z)^T \beta$$

4) the relative cumulative incidence function "rcif" model where $h(x) = \exp(x)$ and

$$g(t, x, z) = x^T A(t) + (\text{diag}(t^p)z)^T \beta$$

) Where p by default is 1 for the additive model and 0 for the other models. In general p may be powers of the same length as z .

Usage

```
comp.risk(formula,data=sys.parent(),cause,times=NULL,Nit=50,
clusters=NULL,est=NULL,fix.gamma=0,gamma=0,n.sim=500,weighted=0,model="additive",
causeS=1,cens.code=0,detail=0,interval=0.01,resample.iid=1,
cens.model="KM",time.pow=NULL,time.pow.test=NULL,silent=1,conv=1e-6,
weights=NULL,max.clust=1000,n.times=50,first.time.p=0.05,
trunc.p=NULL,entry.time=NULL,cens.weight=NULL,admin.cens=NULL)
```

Arguments

formula	a formula object, with the response on the left of a '~' operator, and the terms on the right. The response must be a survival object as returned by the 'Surv' function. The status indicator is not important here. Time-invariant regressors are specified by the wrapper const(), and cluster variables (for computing robust variances) by the wrapper cluster().
data	a data.frame with the variables.
cause	specifies the causes related to the death times, the value 0 is the censoring value.
times	specifies the times at which the estimator is considered. The default is cause "1" jump times, with the first 10 percent or max 20 jump points removed for numerical stability in simulations.
Nit	number of iterations for Newton-Raphson algorithm.
clusters	specifies cluster structure, for backwards compability.
est	possible starting value for nonparametric component of model.
fix.gamma	to keep gamma fixed, possibly at 0.
gamma	starting value for constant effects.
n.sim	number of simulations in resampling.
weighted	Not implemented. To compute a variance weighted version of the test-processes used for testing time-varying effects.
model	"additive", "prop"ortional, "rcif", or "logistic".
causeS	specificies which cause we consider.
cens.code	specificies the code for the censoring.
detail	if 0 no details are printed during iterations, if 1 details are given.
interval	specifies that we only consider timepoints where the Kaplan-Meier of the censoring distribution is larger than this value.
resample.iid	to return the iid decomposition, that can be used to construct confidence bands for predictions
cens.model	specified which model to use for the ICPW, KM is Kaplan-Meier alternatively it may be "cox"
time.pow	specifies that the power at which the time-arguments is transformed, for each of the arguments of the const() terms, default is 1 for the additive model and 0 for the proportional model.

time.pow.test	specifies that the power the time-arguments is transformed for each of the arguments of the non-const() terms. This is relevant for testing if a coefficient function is consistent with the specified form $A_1(t)=\beta_1 t^{\text{time.pow.test}(l)}$. Default is 1 for the additive model and 0 for the proportional model.
silent	if 0 information on convergence problems due to non-invertible derviates of scores are printed.
conv	gives convergence criterie in terms of sum of absolute change of parameters of model
weights	weights for estimating equations.
max.clust	sets the total number of i.i.d. terms in i.i.d. decompositiion. This can limit the amount of memory used by coarsening the clusters. When NULL then all clusters are used. Default is 1000 to save memory and time.
first.time.p	first point for estimation is pth percentile of causeS jump times.
n.times	only uses 50 points for estimation, if NULL then uses all points, subject to p.start condition.
trunc.p	truncation weight for delayed entry, $P(T > \text{entry.time} Z_i)$, typically Cox model.
entry.time	entry times
cens.weight	censoring weights can be given here rather than calculated using the KM, cox or aalen models.
admin.cens	censoring times for the administrative censoring

Value

returns an object of type 'comprisk'. With the following arguments:

cum	cumulative timevarying regression coefficient estimates are computed within the estimation interval.
var.cum	pointwise variances estimates.
gamma	estimate of proportional odds parameters of model.
var.gamma	variance for gamma.
score	sum of absolute value of scores.
gamma2	estimate of constant effects based on the non-parametric estimate. Used for testing of constant effects.
obs.testBeq0	observed absolute value of supremum of cumulative components scaled with the variance.
pval.testBeq0	p-value for covariate effects based on supremum test.
obs.testBeqC	observed absolute value of supremum of difference between observed cumulative process and estimate under null of constant effect.
pval.testBeqC	p-value based on resampling.
obs.testBeqC.is	observed integrated squared differences between observed cumulative and estimate under null of constant effect.

pval.testBeqC.is p-value based on resampling.
 conf.band resampling based constant to construct 95% uniform confidence bands.
 B.iid list of iid decomposition of non-parametric effects.
 gamma.iid matrix of iid decomposition of parametric effects.
 test.procBeqC observed test process for testing of time-varying effects
 sim.test.procBeqC 50 resample processes for for testing of time-varying effects
 conv information on convergence for time points used for estimation.

Author(s)

Thomas Scheike

References

Scheike, Zhang and Gerds (2008), Predicting cumulativeincidence probability by direct binomial regression, *Biometrika*, 95, 205-220.

Scheike and Zhang (2007), Flexible competing risks regression modelling and goodness of fit, *LIDA*, 14, 464-483.

Martinussen and Scheike (2006), *Dynamic regression models for survival data*, Springer.

Examples

```
library(timereg)
data(bmt);

add<-comp.risk(Surv(time,cause>0)~platelet+age+tcell,data=bmt,
bmt$cause,causeS=1,resample.iid=1,n.sim=100)
summary(add)

par(mfrow=c(2,4))
plot(add); plot(add,score=1)

ndata<-data.frame(platelet=c(1,0,0),age=c(0,1,0),tcell=c(0,0,1))
par(mfrow=c(2,3))
out<-predict(add,ndata,uniform=1,n.sim=100)
par(mfrow=c(2,2))
plot(out,multiple=0,uniform=1,col=1:3,lty=1,se=1)

## fits additive model with some constant effects
add.sem<-comp.risk(Surv(time,cause>0)~
const(platelet)+const(age)+const(tcell),data=bmt,
bmt$cause,causeS=1,resample.iid=1,n.sim=100)
summary(add.sem)

out<-predict(add.sem,ndata,uniform=1,n.sim=100)
par(mfrow=c(2,2))
plot(out,multiple=0,uniform=1,col=1:3,lty=1,se=0)
```

```

## Fine & Gray model
fg<-comp.risk(Surv(time,cause>0)~
const(platelet)+const(age)+const(tcell),data=bmt,
bmt$cause,causeS=1,resample.iid=1,model="prop",n.sim=100)
summary(fg)

out<-predict(fg,ndata,uniform=1,n.sim=100)

par(mfrow=c(2,2))
plot(out,multiple=1,uniform=0,col=1:3,lty=1,se=0)

## extended model with time-varying effects
fg.npar<-comp.risk(Surv(time,cause>0)~platelet+age+const(tcell),
data=bmt,bmt$cause,causeS=1,resample.iid=1,model="prop",n.sim=100)
summary(fg.npar);

out<-predict(fg.npar,ndata,uniform=1,n.sim=100)
head(out$P1[,1:5]); head(out$se.P1[,1:5])

par(mfrow=c(2,2))
plot(out,multiple=1,uniform=0,col=1:3,lty=1,se=0)

## Fine & Gray model with alternative parametrization
fg<-comp.risk(Surv(time,cause>0)~
const(platelet)+const(age)+const(tcell),data=bmt,
bmt$cause,causeS=1,resample.iid=1,model="fg",n.sim=100)
summary(fg)

## Left truncated cumulative incidence estimation
## truncation weights computed first, by Cox model
tt <- runif(408)*bmt$time*rbinom(408,1,0.5)*0.5
fgt<- cox.aalen(Surv(tt,time,cause>0)~prop(platelet)+prop(age)+prop(tcell),data=bmt,n.sim=0,robust=0)
cumt <- Cpred(fgt$cum,tt)[,2]
psurv <- exp(-cumt);

fgt<-comp.risk(Surv(time,cause>0)~ const(platelet)+const(age)+const(tcell),data=bmt,
bmt$cause,causeS=1,model="fg",n.sim=0,trunc.p=psurv,entry.time=tt)
summary(fgt)
pfgt <- predict(fgt,X=1,Z=c(0,0,0))
pfgt <- predict(fgt,X=1)
plot(pfgt)

## Left truncated cumulative incidence estimation
## without covariates but using dummy covariate, equivalent with Aalen-Johansen estimator
fgt<- aalen(Surv(tt,time,cause>0)~+1,data=bmt,n.sim=0,robust=0)
cumt <- Cpred(fgt$cum,tt)[,2]
psurv <- exp(-cumt);

pcif<-comp.risk(Surv(time,cause>0)~const(platelet),data=bmt,
bmt$cause,fix.gamma=1,causeS=1,model="fg",n.sim=0,trunc.p=psurv,entry.time=tt)
pout <- predict(pcif,X=1)

```

const *Identifies parametric terms of model*

Description

Specifies which of the regressors that have constant effect.

Author(s)

Thomas Scheike

cox *Identifies proportional excess terms of model*

Description

Specifies which of the regressors that lead to proportional excess hazard

Author(s)

Thomas Scheike

cox.aalen *Fit Cox-Aalen survival model*

Description

Fits an Cox-Aalen survival model. Time dependent variables and counting process data (multiple events per subject) are possible.

$$\lambda_i(t) = Y_i(t)(X_i^T(t)\alpha(t)) \exp(Z_i^T \beta)$$

The model thus contains the Cox's regression model as special case.

Resampling is used for computing p-values for tests of time-varying effects. Test for proportionality is considered by considering the score processes for the proportional effects of model.

The modelling formula uses the standard survival modelling given in the **survival** package.

Usage

```
cox.aalen(formula=formula(data), data=sys.parent(), beta=NULL, Nit=10, detail=0,
start.time=0, max.time=NULL, id=NULL, clusters=NULL, n.sim=500, residuals=0,
robust=1, weighted.test=0, covariance=0, resample.iid=0, weights=NULL, rate.sim=1,
beta.fixed=0, max.clust=1000, exact.deriv=1, silent=0, max.timepoint.sim=100)
```

Arguments

formula	a formula object with the response on the left of a '~' operator, and the independent terms on the right as regressors. The response must be a survival object as returned by the 'Surv' function. Terms with a proportional effect are specified by the wrapper prop(), and cluster variables (for computing robust variances) by the wrapper cluster().
data	a data.frame with the variables.
start.time	start of observation period where estimates are computed.
max.time	end of observation period where estimates are computed. Estimates thus computed from [start.time, max.time]. Default is max of data.
robust	to compute robust variances and construct processes for resampling. May be set to 0 to save memory.
id	For timevarying covariates the variable must associate each record with the id of a subject.
clusters	cluster variable for computation of robust variances.
n.sim	number of simulations in resampling.
weighted.test	to compute a variance weighted version of the test-processes used for testing time-varying effects.
residuals	to returns residuals that can be used for model validation in the function cum.residuals. Estimated martingale increments (dM) and corresponding time vector (time).
covariance	to compute covariance estimates for nonparametric terms rather than just the variances.
resample.iid	to return i.i.d. representation for nonparametric and parametric terms.
beta	starting value for relative risk estimates.
Nit	number of iterations for Newton-Raphson algorithm.
detail	if 0 no details is printed during iterations, if 1 details are given.
weights	weights for weighted analysis.
rate.sim	rate.sim=1 such that resampling of residuals is based on estimated martingales and thus valid in rate case, rate.sim=0 means that resampling is based on counting processes and thus only valid in intensity case.
beta.fixed	option for computing score process for fixed relative risk parameter
max.clust	sets the total number of i.i.d. terms in i.i.d. decomposition. This can limit the amount of memory used by coarsening the clusters. When NULL then all clusters are used. Default is 1000 to save memory and time.
exact.deriv	if 1 then uses exact derivative in last iteration, if 2 then uses exact derivate for all iterations, and if 0 then uses approximation for all computations and there may be a small bias in the variance estimates. For Cox model always exact and all options give same results.
silent	if 1 then oppresses some output.
max.timepoint.sim	considers only this resolution on the time scale for simulations, see time.sim.resolution argument

Details

The data for a subject is presented as multiple rows or 'observations', each of which applies to an interval of observation (start, stop]. For counting process data with the $(start, stop]$ notation is used the 'id' variable is needed to identify the records for each subject. The program assumes that there are no ties, and if such are present random noise is added to break the ties.

Value

returns an object of type "cox.aalen". With the following arguments:

cum	cumulative timevarying regression coefficient estimates are computed within the estimation interval.
var.cum	the martingale based pointwise variance estimates.
robvar.cum	robust pointwise variances estimates.
gamma	estimate of parametric components of model.
var.gamma	variance for gamma.
robvar.gamma	robust variance for gamma.
residuals	list with residuals.
obs.testBeq0	observed absolute value of supremum of cumulative components scaled with the variance.
pval.testBeq0	p-value for covariate effects based on supremum test.
sim.testBeq0	resampled supremum values.
obs.testBeqC	observed absolute value of supremum of difference between observed cumulative process and estimate under null of constant effect.
pval.testBeqC	p-value based on resampling.
sim.testBeqC	resampled supremum values.
obs.testBeqC.is	observed integrated squared differences between observed cumulative and estimate under null of constant effect.
pval.testBeqC.is	p-value based on resampling.
sim.testBeqC.is	resampled supremum values.
conf.band	resampling based constant to construct robust 95% uniform confidence bands.
test.procBeqC	observed test-process of difference between observed cumulative process and estimate under null of constant effect over time.
sim.test.procBeqC	list of 50 random realizations of test-processes under null based on resampling.
covariance	covariances for nonparametric terms of model.
B.iid	Resample processes for nonparametric terms of model.
gamma.iid	Resample processes for parametric terms of model.
loglike	approximate log-likelihood for model, similar to Cox's partial likelihood. Only computed when robust=1.

D2linv	inverse of the derivative of the score function.
score	value of score for final estimates.
test.procProp	observed score process for proportional part of model.
var.score	variance of score process (optional variation estimator for beta.fixed=1 and robust estimator otherwise).
pval.Prop	p-value based on resampling.
sim.supProp	re-sampled absolute supremum values.
sim.test.procProp	list of 50 random realizations of test-processes for proportionality under the model based on resampling.

Author(s)

Thomas Scheike

References

Martinussen and Scheike, Dynamic Regression Models for Survival Data, Springer (2006).

Examples

```
library(timereg)
data(sTRACE)
# Fits Cox model
out<-cox.aalen(Surv(time,status==9)~prop(age)+prop(sex)+
prop(vf)+prop(chf)+prop(diabetes),data=sTRACE,max.time=7,n.sim=100)

# makes Lin, Wei, Ying test for proportionality
summary(out)
par(mfrow=c(2,3))
plot(out,score=1)

# Fits Cox-Aalen model
out<-cox.aalen(Surv(time,status==9)~prop(age)+prop(sex)+
vf+chf+prop(diabetes),data=sTRACE,max.time=7,n.sim=100)

summary(out)
par(mfrow=c(2,3))
plot(out)

# Fits model Cox-Aalen model and
# simulates based on time-grouping to save computing time
out<- cox.aalen(Surv(time,status==9)~prop(sex)+prop(age)+prop(wmi),data=sTRACE,
max.timepoint.sim=30,n.sim=100)
summary(out)
par(mfrow=c(2,2))
plot(out)
plot(out,score=1)
```

`csl`*CSL liver cirrhosis data*

Description

Survival status for the liver cirrhosis patients of Schlichting et al.

Format

This data frame contains the following columns:

id a numeric vector. Id of subject.

time a numeric vector. Time of measurement.

prot a numeric vector. Prothrombin level at measurement time.

dc a numeric vector code. 0: censored observation, 1: died at eventT.

eventT a numeric vector. Time of event (death).

treat a numeric vector code. 0: active treatment of prednisone, 1: placebo treatment.

sex a numeric vector code. 0: female, 1: male.

age a numeric vector. Age of subject at inclusion time subtracted 60.

prot.base a numeric vector. Prothrombin base level before entering the study.

prot.prev a numeric vector. Level of prothrombin at previous measurement time.

lt a numeric vector. Gives the starting time for the time-intervals.

rt a numeric vector. Gives the stopping time for the time-intervals.

Source

P.K. Andersen

References

Schlichting, P., Christensen, E., Andersen, P., Fauerholds, L., Juhl, E., Poulsen, H. and Tygstrup, N. (1983), The Copenhagen Study Group for Liver Diseases, *Hepatology* 3, 889–895

Examples

```
data(csl)
names(csl)
```

cum.residuals *Model validation based on cumulative residuals*

Description

Computes cumulative residuals and approximative p-values based on resampling techniques.

Usage

```
cum.residuals(object,data=sys.parent(),modelmatrix=0,cum.resid=1,
              n.sim=500,weighted.test=1,max.point.func=50)
```

Arguments

object	an object of class 'aalen', 'timecox', 'cox.aalen' where the residuals are returned ('residuals=1')
data	data frame based on which residuals are computed.
modelmatrix	specifies a grouping of the data that is used for cumulating residuals. Must have same size as data and be ordered in the same way.
n.sim	number of simulations in resampling.
weighted.test	to compute a variance weighted version of the test-processes used for testing constant effects of covariates.
cum.resid	to compute residuals versus each of the continuous covariates in the model.
max.point.func	limits the amount of computations, only considers a max of 50 points on the covariate scales.

Value

returns an object of type "cum.residuals" with the following arguments:

cum	cumulative residuals versus time for the groups specified by modelmatrix.
var.cum	the martingale based pointwise variance estimates.
robvar.cum	robust pointwise variances estimates of cumulatives.
obs.testBeq0	observed absolute value of supremum of cumulative components scaled with the variance.
pval.testBeq0	p-value covariate effects based on supremum test.
sim.testBeq0	resampled supremum value.
conf.band	resampling based constant to construct robust 95% uniform confidence bands for cumulative residuals.
obs.test	absolute value of supremum of observed test-process.
pval.test	p-value for supremum test statistic.
sim.test	resampled absolute value of supremum cumulative residuals.
proc.cumz	observed cumulative residuals versus all continuous covariates of model.
sim.test.proccumz	list of 50 random realizations of test-processes under model for all continuous covariates.

Author(s)

Thomas Scheike

References

Martinussen and Scheike, Dynamic Regression Models for Survival Data, Springer (2006).

Examples

```
library(survival)
data(sTRACE)
# Fits Aalen model and returns residuals
fit<-aalen(Surv(time,status==9)~age+sex+diabetes+chf+vf,
sTRACE,max.time=7,n.sim=0,residuals=1)

# constructs and simulates cumulative residuals versus age groups
fit.mg<-cum.residuals(fit,sTRACE,n.sim=100,
modelmatrix=model.matrix(~-1+factor(cut(age,4)),sTRACE))

par(mfrow=c(1,4))
# cumulative residuals with confidence intervals
plot(fit.mg);
# cumulative residuals versus processes under model
plot(fit.mg,score=1);
summary(fit.mg)

# cumulative residuals vs. covariates Lin, Wei, Ying style
fit.mg<-cum.residuals(fit,sTRACE,cum.resid=1,n.sim=100)

par(mfrow=c(2,4))
plot(fit.mg,score=2)
summary(fit.mg)
```

diabetes

The Diabetic Retinopathy Data

Description

The data was collected to test a laser treatment for delaying blindness in patients with diabetic retinopathy. The subset of 197 patients given in Huster et al. (1989) is used.

Format

This data frame contains the following columns:

id a numeric vector. Patient code.

agedx a numeric vector. Age of patient at diagnosis.

time a numeric vector. Survival time: time to blindness or censoring.

status a numeric vector code. Survival status. 1: blindness, 0: censored.

trteye a numeric vector code. Random eye selected for treatment. 1: left eye 2: right eye.

treat a numeric vector. 1: treatment 0: untreated.

adult a numeric vector code. 1: younger than 20, 2: older than 20.

Source

Huster W.J. and Brookmeyer, R. and Self. S. (1989) MOdelling paired survival data with covariates, Biometrics 45, 145-56.

Examples

```
data(diabetes)
names(diabetes)
```

dynreg	<i>Fit time-varying regression model</i>
--------	--

Description

Fits time-varying regression model with partly parametric components. Time-dependent variables for longitudinal data. The model assumes that the mean of the observed responses given covariates is a linear time-varying regression model :

$$E(Z_{ij}|X_{ij}(t)) = \beta^T(t)X_{ij}^1(t) + \gamma^T X_{ij}^2(t)$$

where Z_{ij} is the j 'th measurement at time t for the i 'th subject with covariates X_{ij}^1 and X_{ij}^2 . Re-sampling is used for computing p-values for tests of timevarying effects.

Usage

```
dynreg(formula,data=sys.parent(),aalenmod,bandwidth=0.5,id=NULL,
bhat=NULL,start.time=0,max.time=NULL,n.sim=500,
meansub=1,weighted.test=0,resample=0)
```

Arguments

formula	a formula object with the response on the left of a '~' operator, and the independent terms on the right as regressors.
data	a data.frame with the variables.
start.time	start of observation period where estimates are computed.
max.time	end of observation period where estimates are computed. Estimates thus computed from [start.time, max.time]. Default is max of data.
id	For timevarying covariates the variable must associate each record with the id of a subject.

n.sim	number of simulations in resampling.
weighted.test	to compute a variance weighted version of the test-processes used for testing time-varying effects.
aalenmod	Aalen model for measurement times. Specified as a survival model (see aalen function).
bandwidth	bandwidth for local iterations. Default is 50% of the range of the considered observation period.
bhat	initial value for estimates. If NULL local linear estimate is computed.
meansub	if '1' then the mean of the responses is subtracted before the estimation is carried out.
resample	returns resample processes.

Details

The data for a subject is presented as multiple rows or 'observations', each of which applies to an interval of observation (start, stop]. For counting process data with the)start,stop] notation is used the 'id' variable is needed to identify the records for each subject. The program assumes that there are no ties, and if such are present random noise is added to break the ties.

Value

returns an object of type "dynreg". With the following arguments:

cum the cumulative regression coefficients. This is the efficient estimator based on an initial smoother obtained by local linear regression :

$$\hat{B}(t) = \int_0^t \tilde{\beta}(s) ds + \int_0^t X^{-} (Diag(z) - Diag(X^T(s)\tilde{\beta}(s))) dp(ds \times dz),$$

where $\tilde{\beta}(t)$ is an initial estimate either provided or computed by local linear regression. To plot this estimate use type="eff.smooth" in the plot() command.

var.cum	the martingale based pointwise variance estimates.
robvar.cum	robust pointwise variances estimates.
gamma	estimate of semi-parametric components of model.
var.gamma	variance for gamma.
robvar.gamma	robust variance for gamma.
cum0	simple estimate of cumulative regression coefficients that does not use use an initial smoothing based estimate

$$\hat{B}_0(t) = \int_0^t X^{-} Diag(z) dp(ds \times dz).$$

To plot this estimate use type="0.mpp" in the plot() command.

var.cum0 the martingale based pointwise variance estimates of cum0.
 cum.ms estimate of cumulative regression coefficients based on initial smoother (but robust to this estimator).

$$\hat{B}_{ms}(t) = \int_0^t X^- (Diag(z) - f(s)) dp(ds \times dz),$$

where f is chosen as the matrix

$$f(s) = Diag(X^T(s)\tilde{\beta}(s))(I - X_\alpha(s)X_\alpha^-(s)),$$

where X_α is the design for the sampling intensities.

This is also an efficient estimator when the initial estimator is consistent for $\beta(t)$ and then asymptotically equivalent to cum, but small sample properties appear inferior. Its variance is estimated by var.cum.

To plot this estimate use type="ms.mpp" in the plot() command.

cum.ly estimator where local averages are subtracted. Special case of cum.ms. To plot this estimate use type="ly.mpp" in plot.

var.cum.ly the martingale based pointwise variance estimates.

gamma0 estimate of parametric component of model.

var.gamma0 estimate of variance of parametric component of model.

gamma.ly estimate of parametric components of model.

var.gamma.ly estimate of variance of parametric component of model.

gamma.ms estimate of variance of parametric component of model.

var.gamma.ms estimate of variance of parametric component of model.

obs.testBeq0 observed absolute value of supremum of cumulative components scaled with the variance.

pval.testBeq0 p-value for covariate effects based on supremum test.

sim.testBeq0 resampled supremum values.

obs.testBeqC observed absolute value of supremum of difference between observed cumulative process and estimate under null of constant effect.

pval.testBeqC p-value based on resampling.

sim.testBeqC resampled supremum values.

obs.testBeqC.is observed integrated squared differences between observed cumulative and estimate under null of constant effect.

pval.testBeqC.is p-value based on resampling.

sim.testBeqC.is resampled supremum values.

conf.band resampling based constant to construct robust 95% uniform confidence bands.

test.procBeqC observed test-process of difference between observed cumulative process and estimate under null of constant effect.

sim.test.procBeqC list of 50 random realizations of test-processes under null based on resampling.

covariance covariances for nonparametric terms of model.

Author(s)

Thomas Scheike

References

Martinussen and Scheike, Dynamic Regression Models for Survival Data, Springer (2006).

Examples

```

library(survival)
data(csl)
indi.m<-rep(1,length(csl$lt))

# Fits time-varying regression model
out<-dynreg(prot~treat+prot.prev+sex+age,data=csl,
Surv(lt,rt,indi.m)~+1,start.time=0,max.time=2,id=csl$id,
n.sim=100,bandwidth=0.7,meansub=0)
summary(out)
par(mfrow=c(2,3))
plot(out)

# Fits time-varying semi-parametric regression model.
outs<-dynreg(prot~treat+const(prot.prev)+const(sex)+const(age),data=csl,
Surv(lt,rt,indi.m)~+1,start.time=0,max.time=2,id=csl$id,
n.sim=100,bandwidth=0.7,meansub=0)
summary(outs)

```

Gprop.odds

*Fit Generalized Semiparametric Proportional Odds Model***Description**

Fits a semiparametric proportional odds model:

$$\text{logit}(1 - S_{X,Z}(t)) = \log(X^T A(t)) + \beta^T Z$$

where $A(t)$ is increasing but otherwise unspecified. Model is fitted by maximising the modified partial likelihood. A goodness-of-fit test by considering the score functions is also computed by resampling methods.

An alternative way of writing the model :

$$S_{X,Z}(t) = \frac{\exp(-\beta^T Z)}{(X^T A(t)) + \exp(-\beta^T Z)}$$

such that β is the log-odds-ratio of dying before time t , and $A(t)$ is the odds-ratio.

The modelling formula uses the standard survival modelling given in the **survival** package.

Usage

```
Gprop.odds(formula = formula(data), data=sys.parent(), beta=0, Nit=50,
  detail=0, start.time=0, max.time=NULL, id=NULL, n.sim=500, weighted.test=0,
  sym=0, mle.start=0)
```

Arguments

formula	a formula object, with the response on the left of a '~' operator, and the terms on the right. The response must be a survival object as returned by the 'Surv' function.
data	a data.frame with the variables.
start.time	start of observation period where estimates are computed.
max.time	end of observation period where estimates are computed. Estimates thus computed from [start.time, max.time]. This is very useful to obtain stable estimates, especially for the baseline. Default is max of data.
id	For timevarying covariates the variable must associate each record with the id of a subject.
n.sim	number of simulations in resampling.
weighted.test	to compute a variance weighted version of the test-processes used for testing time-varying effects.
beta	starting value for relative risk estimates
Nit	number of iterations for Newton-Raphson algorithm.
detail	if 0 no details is printed during iterations, if 1 details are given.
sym	to use symmetrized second derivative in the case of the estimating equation approach (profile=0). This may improve the numerical performance.
mle.start	starting values for relative risk parameters.

Details

The data for a subject is presented as multiple rows or "observations", each of which applies to an interval of observation (start, stop]. The program essentially assumes no ties, and if such are present a little random noise is added to break the ties.

Value

returns an object of type 'cox.aalen'. With the following arguments:

cum	cumulative timevarying regression coefficient estimates are computed within the estimation interval.
var.cum	the martingale based pointwise variance estimates.
robvar.cum	robust pointwise variances estimates.
gamma	estimate of proportional odds parameters of model.
var.gamma	variance for gamma.
robvar.gamma	robust variance for gamma.

residuals	list with residuals. Estimated martingale increments (dM) and corresponding time vector (time).
obs.testBeq0	observed absolute value of supremum of cumulative components scaled with the variance.
pval.testBeq0	p-value for covariate effects based on supremum test.
sim.testBeq0	resampled supremum values.
obs.testBeqC	observed absolute value of supremum of difference between observed cumulative process and estimate under null of constant effect.
pval.testBeqC	p-value based on resampling.
sim.testBeqC	resampled supremum values.
obs.testBeqC.is	observed integrated squared differences between observed cumulative and estimate under null of constant effect.
pval.testBeqC.is	p-value based on resampling.
sim.testBeqC.is	resampled supremum values.
conf.band	resampling based constant to construct robust 95% uniform confidence bands.
test.procBeqC	observed test-process of difference between observed cumulative process and estimate under null of constant effect over time.
loglike	modified partial likelihood, pseudo profile likelihood for regression parameters.
D2linv	inverse of the derivative of the score function.
score	value of score for final estimates.
test.procProp	observed score process for proportional odds regression effects.
pval.Prop	p-value based on resampling.
sim.supProp	re-sampled supremum values.
sim.test.procProp	list of 50 random realizations of test-processes for constant proportional odds under the model based on resampling.

Author(s)

Thomas Scheike

References

- Scheike, A flexible semiparametric transformation model for survival data, *Lifetime Data Anal.* (to appear).
- Martinussen and Scheike, *Dynamic Regression Models for Survival Data*, Springer (2006).

Examples

```

library(survival)
data(sTRACE)
# Fits Proportional odds model with stratified baseline
age.c<-scale(sTRACE$age,scale=FALSE);
out<-Gprop.odds(Surv(time,status==9)~diabetes+prop(age.c)+prop(chf)+
prop(sex)+prop(vf),sTRACE,max.time=7,n.sim=100)
summary(out)
par(mfrow=c(2,3))
plot(out,sim.ci=2); plot(out,score=1)

# Fits Proportional odds model with baseline on additive form
# thus giving odds-ratio of dyings for vf and diabetes
out<-Gprop.odds(Surv(time,status==9)~vf+diabetes+prop(age.c)+prop(chf)+
prop(sex),sTRACE,max.time=7,n.sim=100)
summary(out)
par(mfrow=c(2,3))
plot(out,sim.ci=2); plot(out,score=1)

```

krylow.pls

*Fits Krylow based PLS for additive hazards model***Description**

Fits the PLS estimator for the additive risk model based on the least squares fitting criterion

$$L(\beta, D, d) = \beta^T D \beta - 2\beta^T d$$

where $D = \int ZHZdt$ and $d = \int ZHdN$.

Usage

```
krylow.pls(D,d,dim)
```

Arguments

D	defined above
d	defined above
dim	number of pls dimensions

Value

returns a list with the following arguments:

beta	PLS regression coefficients
------	-----------------------------

Author(s)

Thomas Scheike

References

Martinussen and Scheike, The Aalen additive hazards model with high-dimensional regressors, submitted.

Martinussen and Scheike, Dynamic Regression Models for Survival Data, Springer (2006).

Examples

```
## makes data for pbc complete case
data(mypbc)
pbc<-mypbc
pbc$time<-pbc$time+runif(418)*0.1; pbc$time<-pbc$time/365
pbc<-subset(pbc,complete.cases(pbc));
covs<-as.matrix(pbc[, -c(1:3,6)])
covs<-cbind(covs[,c(1:6,16)],log(covs[,7:15]))

## computes the matrices needed for the least squares
## criterion
out<-aalen(Surv(time,status>=1)~const(covs),pbc,robust=0,n.sim=0)
S=out$intZHZ; s=out$intZHdN;

out<-krylow.pls(S,s,2)
```

mela.pop

Melanoma data and Danish population mortality by age and sex

Description

Melanoma data with background mortality of Danish population.

Format

This data frame contains the following columns:

id a numeric vector. Gives patient id.

sex a numeric vector. Gives sex of patient.

start a numeric vector. Gives the starting time for the time-interval for which the covariate rate is representative.

stop a numeric vector. Gives the stopping time for the time-interval for which the covariate rate is representative.

status a numeric vector code. Survival status. 1: dead from melanoma, 0: alive or dead from other cause.

age a numeric vector. Gives the age of the patient at removal of tumor.

rate a numeric vector. Gives the population mortality for the given sex and age. Based on Table A.2 in Andersen et al. (1993).

Source

Andersen, P.K., Borgan O, Gill R.D., Keiding N. (1993), *Statistical Models Based on Counting Processes*, Springer-Verlag.

Examples

```
data(mela.pop)
names(mela.pop)
```

melanoma

The Melanoma Survival Data

Description

The melanoma data frame has 205 rows and 7 columns. It contains data relating to survival of patients after operation for malignant melanoma collected at Odense University Hospital by K.T. Drzewiecki.

Format

This data frame contains the following columns:

no a numeric vector. Patient code.

status a numeric vector code. Survival status. 1: dead from melanoma, 2: alive, 3: dead from other cause.

days a numeric vector. Survival time.

ulc a numeric vector code. Ulceration, 1: present, 0: absent.

thick a numeric vector. Tumour thickness (1/100 mm).

sex a numeric vector code. 0: female, 1: male.

Source

Andersen, P.K., Borgan O, Gill R.D., Keiding N. (1993), *Statistical Models Based on Counting Processes*, Springer-Verlag.

Drzewiecki, K.T., Ladefoged, C., and Christensen, H.E. (1980), Biopsy and prognosis for cutaneous malignant melanoma in clinical stage I. *Scand. J. Plast. Reconstr. Surg.* 14, 141-144.

Examples

```
data(melanoma)
names(melanoma)
```

pe.sasieni

*Fits Proportional excess hazards model with fixed offsets***Description**

Fits proportional excess hazards model. The Sasieni proportional excess risk model. The models are written using the survival modelling given in the survival package.

Usage

```
pe.sasieni(formula=formula(data), data=sys.parent(),
           id=NULL, start.time=0, max.time=NULL, offsets=0, Nit=50, detail=0, n.sim=500)
```

Arguments

formula	a formula object, with the response on the left of a '~' operator, and the terms on the right. The response must be a survival object as returned by the 'Surv' function.
data	a data.frame with the variables.
id	gives the number of individuals.
start.time	starting time for considered time-period.
max.time	stopping considered time-period if different from 0. Estimates thus computed from [0,max.time] if max.time>0. Default is max of data.
offsets	fixed offsets giving the mortality.
Nit	number of iterations.
detail	if detail is one, prints iteration details.
n.sim	number of simulations, 0 for no simulations.

Details

The program assumes that there are no ties, and if such are present random noise is added to break the ties.

Value

Returns an object of type "pe.sasieni". With the following arguments:

cum	baseline of Cox model excess risk.
var.cum	pointwise variance estimates for estimated cumulatives.
gamma	estimate of relative risk terms of model.
var.gamma	variance estimates for gamma.
Ut	score process for Cox part of model.
D2linv	The inverse of the second derivative.
score	final score
test.Prop	re-sampled absolute supremum values.
pval.Prop	p-value based on resampling.

Author(s)

Thomas Scheike

References

Martinussen and Scheike, Dynamic Regression Models for Survival Data, Springer Verlag (2006).
 Sasieni, P.D., Proportional excess hazards, Biometrika (1996), 127–41.
 Cortese, G. and Scheike, T.H., Dynamic regression hazards models for relative survival (2007), submitted.

Examples

```
data(mela.pop)
out<-pe.sasieni(Surv(start,stop,status==1)~age+sex,mela.pop,
id=1:205,Nit=10,max.time=7,offsets=mela.pop$rate,detail=0,n.sim=100)
summary(out)

ul<-out$cum[,2]+1.96*out$var.cum[,2]^0.5
ll<-out$cum[,2]-1.96*out$var.cum[,2]^0.5
plot(out$cum,type="s",ylim=range(ul,ll))
lines(out$cum[,1],ul,type="s"); lines(out$cum[,1],ll,type="s")
# see also prop.excess function
```

plot.aalen

*Plots estimates and test-processes***Description**

This function plots the non-parametric cumulative estimates for the additive risk model or the test-processes for the hypothesis of time-varying effects with re-sampled processes under the null.

Usage

```
## S3 method for class 'aalen'
plot(x,pointwise.ci=1,hw.ci=0,sim.ci=0,robust=0,
specific.comps=FALSE,level=0.05, start.time=0,stop.time=0,add.to.plot=FALSE,
mains=TRUE,xlab="Time",ylab="Cumulative coefficients",score=FALSE,...)
```

Arguments

x	the output from the "aalen" function.
pointwise.ci	if >1 pointwise confidence intervals are plotted with lty=pointwise.ci
hw.ci	if >1 Hall-Wellner confidence bands are plotted with lty=hw.ci. Only 0.95 % bands can be constructed.
sim.ci	if >1 simulation based confidence bands are plotted with lty=sim.ci. These confidence bands are robust to non-martingale behaviour.

<code>robust</code>	robust standard errors are used to estimate standard error of estimate, otherwise martingale based standard errors are used.
<code>specific.comps</code>	all components of the model is plotted by default, but a list of components may be specified, for example first and third "c(1,3)".
<code>level</code>	gives the significance level.
<code>start.time</code>	start of observation period where estimates are plotted.
<code>stop.time</code>	end of period where estimates are plotted. Estimates thus plotted from [start.time, max.time].
<code>add.to.plot</code>	to add to an already existing plot.
<code>mains</code>	add names of covariates as titles to plots.
<code>xlab</code>	label for x-axis.
<code>ylab</code>	label for y-axis.
<code>score</code>	to plot test processes for test of time-varying effects along with 50 random realization under the null-hypothesis.
<code>...</code>	unused arguments - for S3 compatibility

Author(s)

Thomas Scheike

References

Martinussen and Scheike, Dynamic Regression models for Survival Data, Springer (2006).

Examples

```
# see help(aalen)
```

`plot.cum.residuals` *Plots cumulative residuals*

Description

This function plots the output from the cumulative residuals function "cum.residuals". The cumulative residuals are compared with the performance of similar processes under the model.

Usage

```
## S3 method for class 'cum.residuals'
plot(x, pointwise.ci=1, hw.ci=0, sim.ci=0,
     robust=1, specific.comps=FALSE, level=0.05, start.time=0, stop.time=0,
     add.to.plot=FALSE, mains=TRUE, xlab="Time",
     ylab="Cumulative Residuals", ylim=NULL, score=0, conf.band=FALSE, ...)
```

Arguments

x	the output from the "cum.residuals" function.
pointwise.ci	if >1 pointwise confidence intervals are plotted with lty=pointwise.ci
hw.ci	if >1 Hall-Wellner confidence bands are plotted with lty=hw.ci. Only 95% bands can be constructed.
sim.ci	if >1 simulation based confidence bands are plotted with lty=sim.ci. These confidence bands are robust to non-martingale behaviour.
robust	if "1" robust standard errors are used to estimate standard error of estimate, otherwise martingale based estimate are used.
specific.comps	all components of the model is plotted by default, but a list of components may be specified, for example first and third "c(1,3)".
level	gives the significance level. Default is 0.05.
start.time	start of observation period where estimates are plotted. Default is 0.
stop.time	end of period where estimates are plotted. Estimates thus plotted from [start.time, max.time].
add.to.plot	to add to an already existing plot. Default is "FALSE".
mains	add names of covariates as titles to plots.
xlab	label for x-axis. Default is "Time".
ylab	label for y-axis. Default is "Cumulative Residuals".
ylim	limits for y-axis.
score	if '0' plots related to modelmatrix are specified, thus resulting in grouped residuals, if '1' plots for modelmatrix but with random realizations under model, if '2' plots residuals versus continuous covariates of model with random realizations under the model.
conf.band	makes simulation based confidence bands for the test processes under the 0 based on variance of these processes limits for y-axis. These will give additional information of whether the observed cumulative residuals are extreme or not when based on a variance weighted test.
...	unused arguments - for S3 compatibility

Author(s)

Thomas Scheike

References

Martinussen and Scheike, Dynamic Regression Models for Survival Data, Springer (2006).

Examples

```
# see cum.residuals for examples
```

plot.dynreg *Plots estimates and test-processes*

Description

This function plots the non-parametric cumulative estimates for the additive risk model or the test-processes for the hypothesis of constant effects with re-sampled processes under the null.

Usage

```
## S3 method for class 'dynreg'
plot(x, type="eff.smooth", pointwise.ci=1, hw.ci=0,
     sim.ci=0, robust=0, specific.comps=FALSE, level=0.05, start.time=0,
     stop.time=0, add.to.plot=FALSE, mains=TRUE, xlab="Time",
     ylab="Cumulative coefficients", score=FALSE, ...)
```

Arguments

x	the output from the "dynreg" function.
type	the estimator plotted. Choices "eff.smooth", "ms.mpp", "0.mpp" and "ly.mpp". See the dynreg function for more on this.
pointwise.ci	if >1 pointwise confidence intervals are plotted with lty=pointwise.ci
hw.ci	if >1 Hall-Wellner confidence bands are plotted with lty=hw.ci. Only 0.95 % bands can be constructed.
sim.ci	if >1 simulation based confidence bands are plotted with lty=sim.ci. These confidence bands are robust to non-martingale behaviour.
robust	robust standard errors are used to estimate standard error of estimate, otherwise martingale based estimate are used.
specific.comps	all components of the model is plotted by default, but a list of components may be specified, for example first and third "c(1,3)".
level	gives the significance level.
start.time	start of observation period where estimates are plotted.
stop.time	end of period where estimates are plotted. Estimates thus plotted from [start.time, max.time].
add.to.plot	to add to an already existing plot.
mains	add names of covariates as titles to plots.
xlab	label for x-axis.
ylab	label for y-axis.
score	to plot test processes for test of time-varying effects along with 50 random realization under the null-hypothesis.
...	unused arguments - for S3 compatibility

Author(s)

Thomas Scheike

References

Martinussen and Scheike, Dynamic Regression Models for Survival Data, Springer (2006).

Examples

```
library(survival)
data(csl)
indi.m<-rep(1,length(csl$lt))

# Fits time-varying regression model
out<-dynreg(prot~treat+prot.prev+sex+age,csl,
Surv(lt,rt,indi.m)~+1,start.time=0,max.time=3,id=csl$id,
n.sim=100,bandwidth=0.7,meansub=0)

par(mfrow=c(2,3))
# plots estimates
plot(out)
# plots tests-processes for time-varying effects
plot(out,score=TRUE)
```

predict.timereg

Predictions for Survival and Competing Risks Regression for timereg

Description

Make predictions based on the survival models (Aalen and Cox-Aalen) and the competing risks models for the cumulative incidence function (comp.risk). Computes confidence intervals and confidence bands based on resampling.

Usage

```
## S3 method for class 'timereg'
predict(object,newdata=NULL,X=NULL,Z=NULL,
n.sim=500,uniform=TRUE,se=TRUE,alpha=0.05,resample.iid,...)
```

Arguments

object	an object belonging to one of the following classes: comprisk, aalen or cox.aalen
newdata	specifies the data at which the predictions are wanted.
X	alternative to newdata, specifies the nonparametric components for predictions.
Z	alternative to newdata, specifies the parametric components of the model for predictions.
n.sim	number of simulations in resampling.

uniform	computes resampling based uniform confidence bands.
se	computes pointwise standard errors
alpha	specifies the significance level which cause we consider.
resample.iid	set to 1 to return iid decomposition of estimates, 3-dim matrix (predictions x times x subjects)
...	unused arguments - for S3 compatibility

Value

time	vector of time points where the predictions are computed.
unif.band	resampling based constant to construct 95% uniform confidence bands.
model	specifies what model that was fitted.
alpha	specifies the significance level for the confidence intervals. This relates directly to the constant given in unif.band.
newdata	specifies the newdata given in the call.
RR	gives relative risk terms for Cox-type models.
call	gives call for predict function.
initial.call	gives call for underlying object used for predictions.
P1	gives cumulative incidence predictions for competing risks models. Predictions given in matrix form with different subjects in different rows.
S0	gives survival predictions for survival models. Predictions given in matrix form with different subjects in different rows.
se.P1	pointwise standard errors for predictions of P1.
se.S0	pointwise standard errors for predictions of S0.

Author(s)

Thomas Scheike, Jeremy Silver

References

Scheike, Zhang and Gerds (2008), Predicting cumulative incidence probability by direct binomial regression, *Biometrika*, 95, 205-220.

Scheike and Zhang (2007), Flexible competing risks regression modelling and goodness of fit, *LIDA*, 14, 464-483 .

Martinussen and Scheike (2006), *Dynamic regression models for survival data*, Springer.

Examples

```
data(bmt);

add<-comp.risk(Surv(time,cause>0)~platelet+age+tcell,data=bmt,
bmt$cause,causeS=1,resample.iid=1)

ndata<-data.frame(platelet=c(1,0,0),age=c(0,1,0),tcell=c(0,0,1))
```

```

out<-predict(add,newdata=ndata,uniform=1,n.sim=1000)
par(mfrow=c(2,2))
plot(out,multiple=0,uniform=1,col=1:3,lty=1,se=1)
# see comp.risk for further examples.

add<-comp.risk(Surv(time,cause>0)~factor(tcell),data=bmt,
bmt$cause,causeS=1,resample.iid=1)
summary(add)
out<-predict(add,newdata=ndata,uniform=1,n.sim=1000)
plot(out,multiple=1,uniform=1,col=1:3,lty=1,se=1)

## SURVIVAL predictions aalen function
data(sTRACE)
out<-aalen(Surv(time,status==9)~sex+diabetes+chf+vf,
data=sTRACE,max.time=7,n.sim=0,resample.iid=1)

pout<-predict(out,X=rbind(c(1,0,0,0),rep(1,5)))
head(pout$S0[,1:5]); head(pout$se.S0[,1:5])
par(mfrow=c(2,2))
plot(pout,multiple=1,se=0,uniform=0,col=1:2,lty=1:2)
plot(pout,multiple=0,se=1,uniform=1,col=1:2)

out<-aalen(Surv(time,status==9)~const(age)+const(sex)+
const(diabetes)+chf+vf,
data=sTRACE,max.time=7,n.sim=0,resample.iid=1)

pout<-predict(out,X=rbind(c(1,0,0),c(1,1,0)),
Z=rbind(c(55,0,1),c(60,1,1)))
head(pout$S0[,1:5]); head(pout$se.S0[,1:5])
par(mfrow=c(2,2))
plot(pout,multiple=1,se=0,uniform=0,col=1:2,lty=1:2)
plot(pout,multiple=0,se=1,uniform=1,col=1:2)

pout<-predict(out,uniform=0,se=0,newdata=sTRACE[1:10,])
plot(pout,multiple=1,se=0,uniform=0)

out<-cox.aalen(Surv(time,status==9)~prop(age)+prop(sex)+
prop(diabetes)+chf+vf,
data=sTRACE,max.time=7,n.sim=0,resample.iid=1)

pout<-predict(out,X=rbind(c(1,0,0),c(1,1,0)),Z=rbind(c(55,0,1),c(60,1,1)))
head(pout$S0[,1:5]); head(pout$se.S0[,1:5])
par(mfrow=c(2,2))
plot(pout,multiple=1,se=0,uniform=0,col=1:2,lty=1:2)
plot(pout,multiple=0,se=1,uniform=1,col=1:2)

pout<-predict(out,uniform=0,se=0,newdata=sTRACE[1:10,])
plot(pout,multiple=1,se=0,uniform=0)

```

Description

Prints call for object. Lists nonparametric and parametric terms of model

Usage

```
## S3 method for class 'aalen'
print(x,...)
```

Arguments

x an aalen object
... unused arguments - for S3 compatibility

Author(s)

Thomas Scheike

prop	<i>Identifies the multiplicative terms in Cox-Aalen model and proportional excess risk model</i>
------	--

Description

Specifies which of the regressors that belong to the multiplicative part of the Cox-Aalen model

$$\lambda_i(t) = Y_i(t)(X_i^T(t)\alpha(t)) \exp(Z_i^T(t)\beta)$$

for this model prop specified the covariates to be included in $Z_i(t)$

Author(s)

Thomas Scheike

prop.excess	<i>Fits Proportional excess hazards model</i>
-------------	---

Description

Fits proportional excess hazards model.

The models are written using the survival modelling given in the survival package.

Usage

```
prop.excess(formula=formula(data), data=sys.parent(), excess=1,
  , tol=0.0001, max.time=NULL, n.sim=1000, alpha=1, frac=1)
```

Arguments

formula	a formula object, with the response on the left of a '~' operator, and the terms on the right. The response must be a survival object as returned by the 'Surv' function.
data	a data.frame with the variables.
excess	specifies for which of the subjects the excess term is present. Default is that the term is present for all subjects.
tol	tolerance for numerical procedure.
max.time	stopping considered time-period if different from 0. Estimates thus computed from [0,max.time] if max.time>0. Default is max of data.
n.sim	number of simulations in re-sampling.
alpha	tuning paramter in Newton-Raphson procedure. Value smaller than one may give more stable convergence.
frac	number between 0 and 1. Is used in supremum test where observed jump times t_1, \dots, t_k is replaced by t_1, \dots, t_l with $l=\text{round}(\text{frac}*k)$.

Details

The program assumes that there are no ties, and if such are present random noise is added to break the ties.

Value

Returns an object of type "prop.excess". With the following arguments:

cum	estimated cumulative regression functions. First column contains the jump times, then follows the estimated components of additive part of model and finally the excess cumulative baseline.
var.cum	robust pointwise variance estimates for estimated cumulatives.
gamma	estimate of parametric components of model.
var.gamma	robust variance estimate for gamma.
pval	p-value of Kolmogorov-Smirnov test (variance weighted) for excess baseline and Aalen terms, $H: B(t)=0$.
pval.HW	p-value of supremum test (corresponding to Hall-Wellner band) for excess baseline and Aalen terms, $H: B(t)=0$. Reported in summary.
pval.CM	p-value of Cramer von Mises test for excess baseline and Aalen terms, $H: B(t)=0$.
quant	95 percent quantile in distribution of resampled Kolmogorov-Smirnov test statistics for excess baseline and Aalen terms. Used to construct 95 percent simulation band.
quant95HW	95 percent quantile in distribution of resampled supremum test statistics corresponding to Hall-Wellner band for excess baseline and Aalen terms. Used to construct 95 percent Hall-Wellner band.
simScoreProp	observed scoreprocess and 50 resampled scoreprocesses (under model). List with 51 elements.

Author(s)

Torben Martinussen

References

Martinussen and Scheike, Dynamic Regression Models for Survival Data, Springer Verlag (2006).

Examples

```

library(survival)
data(melanoma)
lt<-log(melanoma$thick)      # log-thickness
excess<-(melanoma$thick>=210) # excess risk for thick tumors

# Fits Proportional Excess hazards model
fit<-prop.excess(Surv(days/365,status==1)~sex+ulc+cox(sex)+
                cox(ulc)+cox(lt),melanoma,excess=excess,n.sim=100)
summary(fit)
par(mfrow=c(2,3))
plot(fit)

```

prop.odds

Fit Semiparametric Proportional Odds Model

Description

Fits a semiparametric proportional odds model:

$$\text{logit}(1 - S_Z(t)) = \log(G(t)) + \beta^T Z$$

where $G(t)$ is increasing but otherwise unspecified. Model is fitted by maximising the modified partial likelihood. A goodness-of-fit test by considering the score functions is also computed by resampling methods.

The modelling formula uses the standard survival modelling given in the **survival** package.

Usage

```

prop.odds(formula,data=sys.parent(),beta=0,Nit=10,
          detail=0,start.time=0,max.time=NULL,id=NULL,n.sim=500,weighted.test=0,
          profile=1,sym=0)

```

Arguments

formula	a formula object, with the response on the left of a '~' operator, and the terms on the right. The response must be a survival object as returned by the 'Surv' function.
data	a data.frame with the variables.

start.time	start of observation period where estimates are computed.
max.time	end of observation period where estimates are computed. Estimates thus computed from [start.time, max.time]. This is very useful to obtain stable estimates, especially for the baseline. Default is max of data.
id	For timevarying covariates the variable must associate each record with the id of a subject.
n.sim	number of simulations in resampling.
weighted.test	to compute a variance weighted version of the test-processes used for testing time-varying effects.
beta	starting value for relative risk estimates
Nit	number of iterations for Newton-Raphson algorithm.
detail	if 0 no details is printed during iterations, if 1 details are given.
profile	if profile is 1 then modified partial likelihood is used, profile=0 fits by simple estimating equation. The modified partial likelihood is recommended.
sym	to use symmetrized second derivative in the case of the estimating equation approach (profile=0). This may improve the numerical performance.

Details

The data for a subject is presented as multiple rows or "observations", each of which applies to an interval of observation (start, stop]. The program essentially assumes no ties, and if such are present a little random noise is added to break the ties.

Value

returns an object of type 'cox.aalen'. With the following arguments:

cum	cumulative timevarying regression coefficient estimates are computed within the estimation interval.
var.cum	the martingale based pointwise variance estimates.
robvar.cum	robust pointwise variances estimates.
gamma	estimate of proportional odds parameters of model.
var.gamma	variance for gamma.
robvar.gamma	robust variance for gamma.
residuals	list with residuals. Estimated martingale increments (dM) and corresponding time vector (time).
obs.testBeq0	observed absolute value of supremum of cumulative components scaled with the variance.
pval.testBeq0	p-value for covariate effects based on supremum test.
sim.testBeq0	resampled supremum values.
obs.testBeqC	observed absolute value of supremum of difference between observed cumulative process and estimate under null of constant effect.
pval.testBeqC	p-value based on resampling.

`sim.testBeqC` resampled supremum values.
`obs.testBeqC.is` observed integrated squared differences between observed cumulative and estimate under null of constant effect.
`pval.testBeqC.is` p-value based on resampling.
`sim.testBeqC.is` resampled supremum values.
`conf.band` resampling based constant to construct robust 95% uniform confidence bands.
`test.procBeqC` observed test-process of difference between observed cumulative process and estimate under null of constant effect over time.
`loglike` modified partial likelihood, pseudo profile likelihood for regression parameters.
`D2linv` inverse of the derivative of the score function.
`score` value of score for final estimates.
`test.procProp` observed score process for proportional odds regression effects.
`pval.Prop` p-value based on resampling.
`sim.supProp` re-sampled supremum values.
`sim.test.procProp` list of 50 random realizations of test-processes for constant proportional odds under the model based on resampling.

Author(s)

Thomas Scheike

References

Martinussen and Scheike, Dynamic Regression Models for Survival Data, Springer (2006).

Examples

```

library(survival)
data(sTRACE)
# Fits Proportional odds model
out<-prop.odds(Surv(time,status==9)~age+diabetes+chf+vf+sex,
sTRACE,max.time=7,n.sim=100)

summary(out)

par(mfrow=c(2,3))
plot(out,sim.ci=2)
plot(out,score=1)

```

pval	<i>For internal use</i>
------	-------------------------

Description

for internal use

Author(s)

Thomas Scheike

summary.aalen	<i>Prints summary statistics</i>
---------------	----------------------------------

Description

Computes p-values for test of significance for nonparametric terms of model, p-values for test of constant effects based on both supremum and integrated squared difference.

Returns parameter estimates and their standard errors.

Usage

```
## S3 method for class 'aalen'  
summary(object,digits=3,...)
```

Arguments

object	an aalen object.
digits	number of digits in printouts.
...	unused arguments - for S3 compatibility

Author(s)

Thomas Scheike

References

Martinussen and Scheike,

Examples

```
### see help(aalen)
```

`summary.cum.residuals` *Prints summary statistics for goodness-of-fit tests based on cumulative residuals*

Description

Computes p-values for extreme behaviour relative to the model of various cumulative residual processes.

Usage

```
## S3 method for class 'cum.residuals'
summary(object,digits=3,...)
```

Arguments

<code>object</code>	output from the <code>cum.residuals()</code> function.
<code>digits</code>	number of digits in printouts.
<code>...</code>	unused arguments - for S3 compatibility

Author(s)

Thomas Scheike

Examples

```
# see cum.residuals for examples
```

`timecox` *Fit Cox model with partly timevarying effects.*

Description

Fits proportional hazards model with some effects time-varying and some effects constant. Time dependent variables and counting process data (multiple events per subject) are possible.

Resampling is used for computing p-values for tests of timevarying effects.

The modelling formula uses the standard survival modelling given in the **survival** package.

Usage

```
timecox(formula=formula(data),data=sys.parent(),
start.time=0,max.time=NULL,id=NULL,clusters=NULL,n.sim=1000,
residuals=0,robust=1,Nit=20,bandwidth=0.5,method="basic",
weighted.test=0,degree=1,covariance=0)
```

Arguments

formula	a formula object with the response on the left of a '~' operator, and the independent terms on the right as regressors. The response must be a survival object as returned by the 'Surv' function. Time-invariant regressors are specified by the wrapper const(), and cluster variables (for computing robust variances) by the wrapper cluster().
data	a data.frame with the variables.
start.time	start of observation period where estimates are computed.
max.time	end of observation period where estimates are computed. Estimates thus computed from [start.time, max.time]. Default is max of data.
robust	to compute robust variances and construct processes for resampling. May be set to 0 to save memory.
id	For timevarying covariates the variable must associate each record with the id of a subject.
clusters	cluster variable for computation of robust variances.
n.sim	number of simulations in resampling.
weighted.test	to compute a variance weighted version of the test-processes used for testing time-varying effects.
residuals	to returns residuals that can be used for model validation in the function cum.residuals
covariance	to compute covariance estimates for nonparametric terms rather than just the variances.
Nit	number of iterations for score equations.
bandwidth	bandwidth for local iterations. Default is 50 % of the range of the considered observation period.
method	Method for estimation. This refers to different parametrisations of the baseline of the model. Options are "basic" where the baseline is written as $\lambda_0(t) = \exp(\alpha_0(t))$ or the "breslow" version where the baseline is parametrised as $\lambda_0(t)$.
degree	gives the degree of the local linear smoothing, that is local smoothing. Possible values are 1 or 2.

Details

The data for a subject is presented as multiple rows or 'observations', each of which applies to an interval of observation (start, stop]. When counting process data with the)start,stop] notation is used the 'id' variable is needed to identify the records for each subject. The program assumes that there are no ties, and if such are present random noise is added to break the ties.

Value

Returns an object of type "timecox". With the following arguments:

cum	cumulative timevarying regression coefficient estimates are computed within the estimation interval.
var.cum	the martingale based pointwise variance estimates.

<code>robvar.cum</code>	robust pointwise variances estimates.
<code>gamma</code>	estimate of parametric components of model.
<code>var.gamma</code>	variance for gamma.
<code>robvar.gamma</code>	robust variance for gamma.
<code>residuals</code>	list with residuals. Estimated martingale increments (dM) and corresponding time vector (time).
<code>obs.testBeq0</code>	observed absolute value of supremum of cumulative components scaled with the variance.
<code>pval.testBeq0</code>	p-value for covariate effects based on supremum test.
<code>sim.testBeq0</code>	resampled supremum values.
<code>obs.testBeqC</code>	observed absolute value of supremum of difference between observed cumulative process and estimate under null of constant effect.
<code>pval.testBeqC</code>	p-value based on resampling.
<code>sim.testBeqC</code>	resampled supremum values.
<code>obs.testBeqC.is</code>	observed integrated squared differences between observed cumulative and estimate under null of constant effect.
<code>pval.testBeqC.is</code>	p-value based on resampling.
<code>sim.testBeqC.is</code>	resampled supremum values.
<code>conf.band</code>	resampling based constant to construct robust 95% uniform confidence bands.
<code>test.procBeqC</code>	observed test-process of difference between observed cumulative process and estimate under null of constant effect over time.
<code>sim.test.procBeqC</code>	list of 50 random realizations of test-processes under null based on resampling.
<code>schoenfeld.residuals</code>	Schoenfeld residuals are returned for "breslow" parametrisation.

Author(s)

Thomas Scheike

References

Martinussen and Scheike, Dynamic Regression Models for Survival Data, Springer (2006).

Examples

```
library(survival)
data(sTRACE)
# Fits time-varying Cox model
out<-timecox(Surv(time/365,status==9)~age+sex+diabetes+chf+vf,
data=sTRACE,max.time=7,n.sim=100)
```

```
summary(out)
par(mfrow=c(2,3))
plot(out)
par(mfrow=c(2,3))
plot(out,score=TRUE)

# Fits semi-parametric time-varying Cox model
out<-timecox(Surv(time/365,status==9)~const(age)+const(sex)+
const(diabetes)+chf+vf,data=sTRACE,max.time=7,n.sim=100)

summary(out)
par(mfrow=c(2,3))
plot(out)
```

TRACE

*The TRACE study group of myocardial infarction***Description**

The TRACE data frame contains 1877 patients and is a subset of a data set consisting of approximately 6000 patients. It contains data relating survival of patients after myocardial infarction to various risk factors.

sTRACE is a subsample consisting of 300 patients.

tTRACE is a subsample consisting of 1000 patients.

Format

This data frame contains the following columns:

id a numeric vector. Patient code.

status a numeric vector code. Survival status. 9: dead from myocardial infarction, 0: alive, 7: dead from other causes.

time a numeric vector. Survival time in years.

chf a numeric vector code. Clinical heart pump failure, 1: present, 0: absent.

diabetes a numeric vector code. Diabetes, 1: present, 0: absent.

vf a numeric vector code. Ventricular fibrillation, 1: present, 0: absent.

wmi a numeric vector. Measure of heart pumping effect based on ultrasound measurements where 2 is normal and 0 is worst.

sex a numeric vector code. 1: female, 0: male.

age a numeric vector code. Age of patient.

Source

The TRACE study group.

Jensen, G.V., Torp-Pedersen, C., Hildebrandt, P., Kober, L., F. E. Nielsen, Melchior, T., Joen, T. and P. K. Andersen (1997), Does in-hospital ventricular fibrillation affect prognosis after myocardial infarction?, *European Heart Journal* 18, 919–924.

Examples

```
data(TRACE)
names(TRACE)
```

```
two.stage
```

```
Fit Clayton-Oakes-Glidden Two-Stage model
```

Description

Fit Clayton-Oakes-Glidden Two-Stage model with Cox-Aalen marginals and regression on the variance parameters.

The model specification allows a regression structure on the variance of the random effects, such it is allowed to depend on covariates fixed within clusters

$$\theta_k = Q_k^T \nu$$

. This is particularly useful to model jointly different groups and to compare their variances.

Fits an Cox-Aalen survival model. Time dependent variables and counting process data (multiple events per subject) are not possible !

The marginal baselines are on the Cox-Aalen form

$$\lambda_{ki}(t) = Y_{ki}(t)(X_{ki}^T(t)\alpha(t)) \exp(Z_{ki}^T\beta)$$

The model thus contains the Cox's regression model and the additive hazards model as special cases. (see `cox.aalen` function for more on this).

The modelling formula uses the standard survival modelling given in the **survival** package. Only for right censored survival data.

Usage

```
two.stage(margsurv,data=sys.parent(),Nit=60, detail=0,
start.time=0,max.time=NULL,id=NULL,clusters=NULL,robust=1,
theta=NULL,theta.des=NULL,var.link=0,step=1,
notaylor=0)
```

Arguments

<code>margsurv</code>	fit of marginal survival cox.aalen model with residuals=2, and resample.iid=1 to get fully correct standard errors. See <code>notaylor</code> below.
<code>data</code>	a data.frame with the variables.
<code>start.time</code>	start of observation period where estimates are computed.
<code>max.time</code>	end of observation period where estimates are computed. Estimates thus computed from [start.time, max.time]. Default is max of data.
<code>id</code>	For timevarying covariates the variable must associate each record with the id of a subject.

clusters	cluster variable for computation of robust variances.
robust	if 0 then totally omits computation of standard errors.
Nit	number of iterations for Newton-Raphson algorithm.
detail	if 0 no details is printed during iterations, if 1 details are given.
theta	starting values for the frailty variance (default=0.1).
theta.des	design for regression for variances. The default is NULL that is equivalent to just one theta and the design with only a baseline.
var.link	default "0" is that the regression design on the variances is without a link, and "1" uses the link function exp.
step	step size for Newton-Raphson.
notaylor	if 1 then ignores variation due to survival model, this is quicker and then resample.iid=0 and residuals=0 is ok for marginal survival model that then is much quicker.

Details

The data for a subject is presented as multiple rows or 'observations', each of which applies to an interval of observation (start, stop]. For counting process data with the)start,stop] notation is used the 'id' variable is needed to identify the records for each subject. Only one record per subject is allowed in the current implementation for the estimation of theta. The program assumes that there are no ties, and if such are present random noise is added to break the ties.

Value

returns an object of type "two.stage". With the following arguments:

cum	cumulative timevarying regression coefficient estimates are computed within the estimation interval.
var.cum	the martingale based pointwise variance estimates.
robvar.cum	robust pointwise variances estimates.
gamma	estimate of parametric components of model.
var.gamma	variance for gamma.
robvar.gamma	robust variance for gamma.
D2linv	inverse of the derivative of the score function.
score	value of score for final estimates.
theta	estimate of Gamma variance for frailty.
var.theta	estimate of variance of theta.
S.theta	estimate of derivative of score of theta.
theta.score	score for theta parameters.

Author(s)

Thomas Scheike

References

Glidden (2000), A Two-Stage estimator of the dependence parameter for the Clayton Oakes model.
 Martinussen and Scheike, Dynamic Regression Models for Survival Data, Springer (2006).

Examples

```
library(survival)
data(diabetes)
# Marginal Cox model with treat as covariate, residuals=2, resample.iid=1 !
marg <- cox.aalen(Surv(time,status) ~ prop(treat) + cluster(id),
  data=diabetes,residuals=2,resample.iid=1,n.sim=0)

fit<-two.stage(marg,data=diabetes,theta=1.0,detail=0,Nit=40)
summary(fit)

# Stratification after adult
theta.des<-model.matrix(~-1+factor(adult),diabetes);
des.t<-model.matrix(~-1+factor(treat),diabetes);
design.treat<-cbind(des.t[,-1]*(diabetes$adult==1),
  des.t[,-1]*(diabetes$adult==2))

# test for common baselines included here
marg1<-cox.aalen(Surv(time,status)~-1+factor(adult)+prop(design.treat)+cluster(id),
  data=diabetes,residuals=2,resample.iid=1)

fit.s<-two.stage(marg1,data=diabetes,Nit=40,theta=1,theta.des=theta.des)
summary(fit.s)

# with common baselines and common treatment effect (although test reject this)
fit.s2<-two.stage(marg,data=diabetes,Nit=40,theta=1,theta.des=theta.des)
summary(fit.s2)

# test for same variance among the two strata
theta.des<-model.matrix(~factor(adult),diabetes);
fit.s3<-two.stage(marg,data=diabetes,Nit=40,theta=1,theta.des=theta.des)
summary(fit.s3)

# to fit model without covariates, use beta.fixed=1 and prop or aalen function
marg <- aalen(Surv(time,status) ~+1+ cluster(id),
  data=diabetes,residuals=2,resample.iid=1,n.sim=0)
fit<-two.stage(marg,data=diabetes,theta=0.95,detail=0)
summary(fit)
```

Index

*Topic **datasets**

- bmt, 7
- cd4, 8
- csl, 18
- diabetes, 20
- mela.pop, 28
- melanoma, 29
- TRACE, 47

*Topic **survival**

- aalen, 2
- additive.pls, 5
- comp.risk, 9
- const, 14
- cox, 14
- cox.aalen, 14
- cum.residuals, 19
- dynreg, 21
- Gprop.odds, 24
- krylow.pls, 27
- pe.sasieni, 30
- plot.aalen, 31
- plot.cum.residuals, 32
- plot.dynreg, 34
- predict.timereg, 35
- print.aalen, 37
- prop, 38
- prop.excess, 38
- prop.odds, 40
- pval, 43
- summary.aalen, 43
- summary.cum.residuals, 44
- timecox, 44
- two.stage, 48

- aalen, 2
- aalen.des (pval), 43
- aalenBase (pval), 43
- aalenBaseC (pval), 43
- additive.compSs (pval), 43
- additive.pls, 5

- additive.plsR (additive.pls), 5

- bmt, 7

- cd4, 8
- check.missing (pval), 43
- coef.aalen (pval), 43
- coef.comprisk (pval), 43
- coef.cox.aalen (pval), 43
- coef.dynreg (pval), 43
- coef.timecox (pval), 43
- coef.two.stage (pval), 43
- coefBase (pval), 43
- comp.risk, 9
- const, 14
- cox, 14
- cox.aalen, 14
- cox.aalenBase (pval), 43
- Cpred (pval), 43
- csl, 18
- Csmooth2B (pval), 43
- CsmoothB (pval), 43
- cum.residuals, 19
- cv.folds (pval), 43

- des.aalen (pval), 43
- diabetes, 20
- dynreg, 21
- dynregBase (pval), 43

- Gprop.odds, 24

- is.diag (pval), 43

- kernel (pval), 43
- krylow.pls, 27

- localTimeReg (pval), 43

- mela.pop, 28
- melanoma, 29

mypbc (pval), 43
 nameestimate (pval), 43
 namematrix (pval), 43
 pava (pval), 43
 pe.sasieni, 30
 persen (pval), 43
 plot.aalen, 31
 plot.comprisk (pval), 43
 plot.cox.aalen (plot.aalen), 31
 plot.cox.aalen2 (pval), 43
 plot.cum.residuals, 32
 plot.cums (pval), 43
 plot.dynreg, 34
 plot.predict (pval), 43
 plot.prop.excess (plot.aalen), 31
 plot.prop.excess2 (pval), 43
 plot.timecox (plot.aalen), 31
 plot.timecox2 (pval), 43
 plot.two.stage (pval), 43
 plotScore (pval), 43
 pls.surv.cv (additive.pls), 5
 pred.cum (pval), 43
 pred.des (pval), 43
 predict.aalen (predict.timereg), 35
 predict.comprisk (predict.timereg), 35
 predict.cox.aalen (predict.timereg), 35
 predict.pls (pval), 43
 predict.timereg, 35
 predict.two.stage (pval), 43
 print.aalen, 37
 print.comprisk (print.aalen), 37
 print.cox.aalen (print.aalen), 37
 print.cum.residuals (print.aalen), 37
 print.dynreg (print.aalen), 37
 print.pe.sasieni (pval), 43
 print.pls (pval), 43
 print.predict.timereg (pval), 43
 print.prop.excess (print.aalen), 37
 print.timecox (print.aalen), 37
 print.two.stage (pval), 43
 prop, 38
 prop.excess, 38
 prop.excessBase (pval), 43
 prop.odds, 40
 pval, 43
 read-design (pval), 43
 read.design (pval), 43
 read.surv (pval), 43
 rm.missing (pval), 43
 semiaalen (pval), 43
 semicox (pval), 43
 semiregBase (pval), 43
 slaaop (pval), 43
 sTRACE (TRACE), 47
 summary.aalen, 43
 summary.comprisk (pval), 43
 summary.cox.aalen (summary.aalen), 43
 summary.cum.residuals, 44
 summary.dynreg (summary.aalen), 43
 summary.pe.sasieni (pe.sasieni), 30
 summary.pe.sasieni (pval), 43
 summary.pls (pval), 43
 summary.predict.timereg (pval), 43
 summary.prop.excess (summary.aalen), 43
 summary.timecox (summary.aalen), 43
 summary.two.stage (pval), 43
 timecox, 44
 timecoxBase (pval), 43
 timetest (pval), 43
 TRACE, 47
 tTRACE (TRACE), 47
 two.stage, 48
 wald.test (pval), 43