Package: SurvDisc (via r-universe)

October 27, 2024

Type Package

Title Discrete Time Survival and Longitudinal Data Analysis

Version 0.1.1

Depends R(>= 2.10.0)

Imports cubature, mvtnorm, MASS, nlme, simex, survival

Maintainer John Lawrence <john.lawrence@fda.hhs.gov>

Description Various functions for discrete time survival analysis and longitudinal analysis. SIMEX method for correcting for bias for errors-in-variables in a mixed effects model. Asymptotic mean and variance of different proportional hazards test statistics using different ties methods given two survival curves and censoring distributions. Score test and Wald test for regression analysis of grouped survival data. Calculation of survival curves for events defined by the response variable in a mixed effects model crossing a threshold with or without confirmation.

License GPL-2

LazyData TRUE

NeedsCompilation no

Author John Lawrence [aut, cre], Jianjin Xu [ctb], Sue Jane Wang [ctb], Jim Hung [ctb]

Date/Publication 2018-05-04 20:25:09 UTC

Repository https://johnlawrence1.r-universe.dev

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

RemoteRef HEAD

RemoteSha 854973d82357fd595e6056eb085d60fba2b1a252

Contents

`AsympDiscSurv` *Asymptotic Estimate of Mean and Variance of Log-hazard Ratio for Discrete Time Survival*

Description

calculates the expected estimated log-hazard ratio and the estimated variance for large sample sizes when there are two groups with possibly non-proportional hazards and possible unequal randomization and censoring distributions.

Usage

```
AsympDiscSurv(h0,h1,p0,p1,method=c("efron","breslow","PrenticeGloeckler"),tol=1E-12)
```

Arguments

`h0`	vector of hazard rates in the control group
`h1`	vector of hazard rates in the treatment group
`p0`	vector of probabilities of being in the risk set and in the control group. See `Details` section below.
`p1`	vector of probabilities of being in the risk set and in the treatment group. See `Details` section below.
`method`	method for handling ties.
`tol`	a positive scalar giving the tolerance at which the maximum absolute value of the gradient is considered close enough to 0 to stop the algorithm.

Details

This calculates the asymptotic mean of the coefficient estimated by a proportional hazards regression model between two groups.

If there are `r` intervals, the vectors need only be of length r-1 since all subjects reaching the final interval will be assumed to have an event at some time in the last interval.

`p0` and `p1` are not the survival curves because they also include information about the allocation ratio between groups and the censoring distribution. The j^th element of `p0` is the probability of being assigned to the control group and being at risk at time `time[j]`. `p0+p1` is always less than

or equal to 1 and should be close to 1 at the first time point and decreasing with time. Note that subjects censored at `time[j]` are not in the risk set, only subjects who have an event at this time or later or who are censored later. This definition of censoring time is the definition used in the reference and may be different than used in other places. Add 1 to all censored times if desired to force censoring to conform with the more standard ways. With equal allocation and no censoring, then `p0[1]=p1[1]=0.5`.

Value

A list which contains:

`coefficients`	the estimated coefficient (log-hazard ratio)
`varn`	the asymptotic variance multiplied by n where n is the total sample size combined in both groups

Author(s)

John Lawrence,<john.lawrence@fda.hhs.gov>

See Also

LongToSurv

Examples

```
set.seed(1234)
nsim=1
n=250
k=50
trt=c(rep(0,n),rep(1,n))
betaef=rep(0,nsim)
varef=betaef
betapg=betaef
varpg=betaef
m1=3.05
for (i in 1:nsim){
  x=rexp(2*n,1)
  x[(n+1):(2*n)]=x[(n+1):(2*n)]/2
  x=ceiling(x*(k-1)/m1)
  x=pmin(x,k)
  cens=rbinom(2*n,1,0.9)
  pg1=PrenticeGloeckler.test(x,cens,trt,k)
  betapg[i]=pg1$coefficient
  varpg[i]=(pg1$coefficient/pg1$wald.test)^2
  efron=survival::coxph(survival::Surv(x,cens)~trt,ties="efron")
  betaef[i]=efron$coef
  varef[i]=efron$var}

h0=0.9*(exp(-c(0:(k-2))*m1/(k-1))-exp(-c(1:(k-1))*m1/(k-1)))
h0=h0/(h0+exp(-c(1:(k-1))*m1/(k-1)))
p0=exp(-c(0:(k-1))*m1/(k-1))
p0=(p0[1:(k-1)]*0.9+p0[2:k]*0.1)/2
```

```
h1=0.9*(exp(-c(0:(k-2))*2*m1/(k-1))-exp(-c(1:(k-1))*2*m1/(k-1)))
h1=h1/(h1+exp(-c(1:(k-1))*2*m1/(k-1)))
p1=exp(-2*c(0:(k-1))*m1/(k-1))
p1=(p1[1:(k-1)]*0.9+p1[2:k]*0.1)/2

fa=AsympDiscSurv(h0=h0, h1=h1,p0=p0,p1=p1)
c(fa$estimate,fa$var/(2*n))
c(mean(betaef),var(betaef),mean(varef))
```

LongToSurv *Longitudinal To Survival Function*

Description

This function calculates the survival curve for events where the events are defined by some function of a variable measured longitudinally. The events can be defined with or without confirmation (see arguments and details). The survival curve is integrated over a distribution of covariates. The term "covariates" is used loosely here and includes all terms in the mixed effects longitudinal model including random effects and error terms. This distribution is assumed to be truncated multivariate normal.

Usage

```
LongToSurv(M,V,L,U,time,p0f,p1f=NULL,method=c("simulation","asymptotic"),
  conf.type=c("scheduled","unscheduled","none"),nsim=100000)
```

Arguments

M	Mean vector for the parent multivariate normal distribution of the covariates.
V	Covariance matrix for the parent multivariate normal distribution of the covariates.
L	vector of lower limits for the covariates.
U	vector of upper limits for the covariates.
time	vector of time points.
p0f	multi-valued function that calculates the probability of crossing the threshold at each sceduled visit time point in the control group. If method="unscheduled", the probability of crossing the threshold at both the scheduled and subsequent unscheduled visit; even the last visit is assumed to allow an unscheduled confirmation visit.
p1f	Optional multi-valued function that calculates the probability of crossing the threshold in the treatment group.
method	Method used; either "simulation" or "analytic". Defaults to "simulation".

conf.type	type of confirmation. "none" meaing a single value crossing the threshold is an event, "scheduled" meaning two consecutive scheduled measurements crossing the threshold, or "unscheduled" meaning that after a qualifying event at a scheduled visit, a subsequent measurement is taken at an unscheduled visit to potentially confirm the event.
nsim	Approximate number of simulated covariate values used. Used only if method = "simulation".

Details

The discrete survival function is found given a distribution of covariates and a longitudinal model. The event is defined by the response variable crossing a threshold value either once (confirmation = "none") or twice in successive time points. The distribution of the covariates is assumed to be truncated multivariate normal. If method is "simulation", then /codensim/accept.rate values of the covariates are simulated first. The truncation conditions are tested and approximately nsim of these covariates will be accepted. The survival curve is found and averaged over the covariate values in the sample. If the method is "analytic", then the survival curve function is integrated analytically (using the adaptIntegrate function from the cubature package).

Value

A list consisting of:

times	numeric vector of time points
S0	numeric vector of survival beyond time t in the control group
S0err	numeric vector of the estimated standard error (or estimated absolute error for analytic method) of S0
S1	numeric vector of survival beyond time t in the test group.
S1err	numeric vector of the estimated standard error (or estimated absolute error for analytic method) of S1
accept.rate	estimate probability that a covariate vector from the parent multivariate normal disitrbution will lie between the truncation limits L and U.

Author(s)

John Lawrence

Examples

```
mu.AGE = 38.582
mu.lbtkv = 6.9276
mu.base.leGFR = 4.2237
var.AGE = 220.73
var.lbtkv = 0.46848
var.base.leGFR=0.19770
cov.AGE.lbtkv = 3.4075
cov.AGE.leGFR = -4.5065
cov.lbtkv.leGFR = -0.16303
sig.intercept=0.03975
```

```
sig.time=0.04505
sig.cor=0.008
res.sd=0.11470307/sqrt(2)

M=c(mu.AGE,mu.lbtkv,mu.base.leGFR,0,0,0)
V=diag(c(var.AGE,var.lbtkv,var.base.leGFR+res.sd^2,res.sd^2,sig.intercept^2,sig.time^2))
V[1,2] = V[2,1] = cov.AGE.lbtkv
V[1,3] = V[3,1] = cov.AGE.leGFR
V[2,3] = V[3,2] = cov.lbtkv.leGFR
V[3,4] = V[4,3] = V[4,4]
V[5,6] = V[6,5] = sig.cor*sig.intercept*sig.time
L=c(18,6.9,3.9,-Inf,-Inf,-Inf)
U=c(40,8,5,Inf,Inf,Inf)
time=c(1:12)/4

p0f=function(x,t) {
  fixed.time=-0.337166
  fixed.age=0.0008176
  fixed.lbtkv=-0.02409
  fixed.leGFR0=0.09591
  trt.acute=-0.047759
  trt.chronic=0.0191574
  res.sd=0.11470307/sqrt(2)
  pnorm((log(0.7)-as.vector(x[5]+outer(x[6]+fixed.age*x[1]+fixed.lbtkv*x[2]+
  fixed.leGFR0*(x[3]-x[4])+fixed.time,t)-x[4]))/res.sd)
}

p1f=function(x,t) {
  fixed.time=-0.337166
  fixed.age=0.0008176
  fixed.lbtkv=-0.02409
  fixed.leGFR0=0.09591
  trt.acute=-0.047759
  trt.chronic=0.0191574
  res.sd=0.11470307/sqrt(2)
  pnorm((log(0.7)-as.vector(x[5]+trt.acute+outer(x[6]+fixed.age*x[1]+fixed.lbtkv*x[2]+
  fixed.leGFR0*(x[3]-x[4])+fixed.time+trt.chronic,t)-x[4]))/res.sd)
}

LTS1=LongToSurv(M,V,L,U,time,p0f,p1f,nsim=100) #nsim much larger than 100 is recommended
LTS1
#LTS2=LongToSurv(M,V,L,U,time,p0f,p1f,method="analytic")
#LTS2
```

PrenticeGloeckler.test

Regression for Grouped Survival Data Function

Description

This function calculates the estimated hazard ratio for grouped survival data described in the reference below.

Usage

```
PrenticeGloeckler.test(time,event,grp,r)
```

Arguments

time	vector of times to event or censoring. The times are assumed to be integers from 1, 2, .., r corresponding to the discrete time points or the continuous time intervals A1, ..., Ar
event	vector of binary status indicator variables (0 = censored at the start of the interval, 1 = event during the interval)
grp	vector of binary group indicators (0 or 1)
r	number of time points or intervals

Details

The hazard functions and hazard ratio are estimated for grouped survival data.

Value

A list consisting of:

coefficient	The estimated coefficient (log hazard ratio) found by maximizing the likelihood.
indx	vector of time points where the hazard functions are estimated. The subset of 1,...,r-1 with at least one event.
gamma	numeric vector with the same length as indx representing the log(-log(hazard rate)) in the control group for time points in the vector indx
grad1	gradient evaluated at (gamma[indx],ceofficient)
r	number of time points or time intervals
hess1	hessian matrix evaluated at the maximum likelihood estimate.
ll0	log-likelihood evaluated at ceofficient=0. includes attributes "gradient" and "hessian"
ll1	log-likelihood at maximum likeohood estimate. includes attributes "gradient" and "hessian"
score.test	value of the score test statistic for testing coefficient=0 (see reference).
lr.test	value of the likelihood ratio test statistic, 2*(ll0-ll1)
wald.test	value of the Wald test statistic; the estimated coefficient divided by the square root of the estimated variance.

Author(s)

John Lawrence

References

Prentice, R. L. and Gloeckler, L.A. (1978). Regression analysis of grouped survival data with application to breast cancer data. *Biometrics*, 57 – 67

Examples

```
set.seed(1234)
nsim=1
n=250
tn=2*n
k=0.1*tn
betaef=rep(0,nsim)
betapg=rep(0,nsim)
cens=rep(1,2*n)
trt=c(rep(0,n),rep(1,n))

for (i in 1:nsim) {
  x=rexp(tn,1)
  x[(n+1):tn]=x[(n+1):tn]/2
  m1=max(x[(n+1):tn])
  x=ceiling(x*(k-1)/m1)
  x[(n+1):tn]=pmin(x[(n+1):tn],k-1)
  x[1:n]=pmin(x[1:n],k)
  pg1=PrenticeGloeckler.test(x,cens,trt,k)
  betapg[i]=pg1$coefficient
  betaef[i]=survival::coxph(survival::Surv(x,cens)~trt,ties="efron")$coef}
mean(betaef)
mean(betapg)
```

---

print.PrenticeGloeckler.test

Print Regression for Grouped Survival Data Function Object

Description

Print coefficient returned by PrenticeGloeckler function.

Usage

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

Arguments

x	object of class PrenticeGloeckler.test returned by call to PrenticeGloeckler function.
...	ignored

Details

Prints the coefficient, the estimated log-hazard ratio.

Author(s)

John Lawrence

See Also

PrenticeGloeckler.test

---

print.SSDS *Print Sample Size Discrete Survival Object*

Description

Print coefficient returned by SampleSizeDiscSurv function.

Usage

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

Arguments

x	object of class SSDS returned by call to SampleSizeDiscSurv function.
...	ignored

Details

Prints a few sentences describing the results of the call to the SampleSizeDiscSurv function.

Author(s)

John Lawrence

See Also

SampleSizeDiscSurv

rowMSD *Mean and Standard Deviation estimates for each row in a matrix*

Description

Calculates the sample mean and standard deviation for each row in a matrix. The mean vector is calculated first. The elements of the matrix are then centered by the mean vector before the sample standard deviation is calculated.

Usage

```
rowMSD(x)
```

Arguments

x	numeric matrix

Value

A list consisting of:

rm	vector of row means
rsd	vector of row standard deviations

Author(s)

John Lawrence

Examples

```
x=matrix(rnorm(1000),nrow=10)
rowMSD(x)
```

SampleSizeDiscSurv *Sample Size for Discrete Time Survival*

Description

Calculates the sample size needed to achieve any given power for any specifed type 1 error rate.

Usage

```
SampleSizeDiscSurv(power=0.9,alpha=0.025,alternative=c("less","greater"),beta0=0,
  h0,h1,p0,p1,ties.method=c("efron","breslow","PrenticeGloeckler"),
  method=c("asymptotic","simulation"),tol,AMV=NULL,nsim=10000,Nvec=NULL,
  test=c("Wald","Score"))
```

Arguments

power	scalar value of the desired power. Default value is 0.9.
alpha	scalar value of the one-sided type 1 error rate. Default value is 0.025.
alternative	character specifying the type of alternative
beta0	scalar value of the log-hazard ratio on the boundary of the null hypothesis. Default is 0.
h0	vector of length r-1 containing the postulated hazard rates in the control group for the times 1, ..., r-1 or corresponding time intervals. Assumed to be r intervals with the last interval being infinite.
h1	vector of postulated hazard rates in the treatment group
p0	vector of probabilities of being in the risk set and in the control group. See Details section below.
p1	vector of probabilities of being in the risk set and in the treatment group. See Details section below.
ties.method	method for handling ties.
method	character specifiying the asymptotic or simalution based method for determining the sample size.
tol	a positive scalar giving the tolerance at which the maximum absolute value of the gradient is considered close enough to 0 to stop the algorithm used if method="asymptotic".
AMV	AsympDiscSurv object from a previous call to the AsympDiscSurv function.
nsim	number of simulations used per N value in the Nvec vector. Used only if method="simulation"
Nvec	vector of sample sizes used in simulation based method. If none specifed, default is to use two N values close to the estimate from the asymptotic method (see details below).
test	character specifying the type of test statistics used. Used only for simulation based method because asymptotically, the tests are equivalent.

Details

If method="asymptotic", then the mean of the test statistic (wald or score, which are equivalent asymptotically) for a sample size divided by sqrt(N) converges to a constant. This constant is found from the parameters in the result of the call to AsympDiscSurv. If the AsympDiscSurv object has already been found, it can be passed to this function in the arguments. If not, then this function calls AsympDiscSurv to find those paraemters.

If method="simulation", then the mean of the test statistic is found for each sample size in the Nvec vector. The mean and variance of the test statistic for each N is found. Then, a linear regression is used to find the sample size that will provide the correct power. Each test statistic is asumed to have a mean that depends on sqrt(N) and the same variance. Theoretically, the variance should be close to 1, but the variance is estimated from the simulated values (not assumed equal to 1). The normality assumption is usually satisfied if the number of events is sufficiently large.

Neither the simulation nor the asymptotic method is reliable if the expected number of events is small (say, less than 20). The asymptotic method is faster. However, the simulation method has

several advantages. First, the asymptotic variance found by the AsympDiscSurv function can differ from the true variance by a few percent even for moderately large sample sizes. The simulation based method estimates the true variance by simulation. Second, for moderatley large sample sizes, the score test can be different from the Wald test. Third, asymptotically the mean of the test statistic is approximately constant times sqrt(N), i.e. a linear function of sqrt(N) with no intercept. But, for small N, the relationship may not be so simple. The simulation method models the relationship for values of N close to the target value without making this strong assumption. The simulation method still assumes that the test statistic is normally distributed, so may be inaccurate for very small sample sizes or rare events.

Iy is assumed there are `r` time intervals, the vectors defining the hazard and at-risk rates have length `r-1` since all subjects reaching the final interval must have an event at some time in the last interval.

`p0` and `p1` are not the survival curves because they also include information about the allocation ratio between groups and the censoring distribution. The j^th element of `p0` is the probability of being assigned to the control group and being at risk at time `time[j]`. `p0+p1` is always less than or equal to 1 and should be close to 1 at the first time point and decreasing with time. Note that subjects censored at `time[j]` are not in the risk set, only subjects who have an event at this time or later or who are censored later. This definition of censoring time is the definition used in the reference and may be different than used in other places. Add 1 to all censored times if desired to force censoring to conform with the more standard ways. With equal allocation and no censoring, then `p0[1]=p1[1]=0.5`.

Value

An object of class SSDS, which is a list containing:

`N`	sample size that should provide the correct power
`alternative`	character specifying the type of alternative
`beta0`	scalar value of the log-hazard ratio on the boundary of the null hypothesis. Default is 0.
`ties.method`	method for handling ties.
`method`	character specifiying the asymptotic or simalution based method for determining the sample size.
`AMV`	AsympDiscSurv object
`EZobj`	required expected value of the test statistic
`Nvec`	vector of sample sizes used in the simulation
`EZvec`	vector of mean values of the test statistic for each value of N
`VZvec`	vector of sample variances for each value of N
`int.est`	estimate of the intercept in the linear relationship between sqrt(N) and expected value of the test statistic.
`slope.est`	estimate of the slope in the linear relationship between sqrt(N) and expected value of the test statistic.
`nsim`	number of simulations used per `N` value in the `Nvec` vector. Used only if `method="simulation"`
`test`	character specifying the type of test statistics used. Used only for simulation based method because asymptotically, the tests are equivalent.

Author(s)

John Lawrence,<john.lawrence@fda.hhs.gov>

See Also

LongToSurv,AsympDiscSurv

Examples

```
set.seed(1234)
k=50
m1=3.05

h0=0.9*(exp(-c(0:(k-2))*m1/(k-1))-exp(-c(1:(k-1))*m1/(k-1)))
h0=h0/(h0+exp(-c(1:(k-1))*m1/(k-1)))
p0=exp(-c(0:(k-1))*m1/(k-1))
p0=(p0[1:(k-1)]*0.9+p0[2:k]*0.1)/2
h1=0.9*(exp(-c(0:(k-2))*2*m1/(k-1))-exp(-c(1:(k-1))*2*m1/(k-1)))
h1=h1/(h1+exp(-c(1:(k-1))*2*m1/(k-1)))
p1=exp(-2*c(0:(k-1))*m1/(k-1))
p1=(p1[1:(k-1)]*0.9+p1[2:k]*0.1)/2

fa=AsympDiscSurv(h0=h0,h1=h1,p0=p0,p1=p1)

(SSDS1=SampleSizeDiscSurv(power=0.9,alpha=0.025,alternative="greater",beta0=0,h0,h1,
  p0,p1,ties.method="efron",method="asymptotic",AMV=fa,Nvec=NULL,test="Wald"))
```

---

simexlme *SIMEX algorithm for linear mixed effects models*

Description

Implementation of the SIMEX algorithm for measurement error models according to Cook and Stefanski.

Usage

```
simexlme(model, model.model, SIMEXvariable, respvar, grpvar, corform, measurement.error,
  measurement.error.resp, lambda = c(0.5, 1, 1.5, 2), B = 100,
  fitting.method = "quadratic", jackknife.estimation = "quadratic")
```

Arguments

model	naive model
model.model	dataframe containing all variables in the model
SIMEXvariable	character name of the variable with measurement error. Assumed to be the baseline measurement.

respvar	character name of the response variable. The response is assumed to represent a change from baseline.
grpvar	character name of the grouping variable for the random effects in the model.
corform	formula for the correlation of residual errors within groups. see example
measurement.error	The known standard deviation of measurement errors for SIMEXvariable.
measurement.error.resp	The known stadard deviaiton for respvar
lambda	vector of lambdas for which the simulation step should be done
B	number of iterations for each lambda
fitting.method	fitting method for extrapolation. Only linear or quadratic are recommended.
jackknife.estimation	specifying the extrapolation method for jackknife variance estimation.

Details

See documentation for mcsimex function. This function for lme models was adapted from that function, which is designed to handle linear and generalized linear models, but not lme models. In this function, the measurement error variable must be the baseline value of some measurement and the response is the change from baseline in the same measurement. There is assumed to be one value of this baseline measurement per level of the grouping variable in the mixed effect model. The correlation between the measurement errors for two response values within a subject is assumed to be equal to be equal to the variance of baseline divided by the sum of the variance of baseline and variance of post-baseline errors. For example, for a study measuring the effect of some weight loss treatment, the grouping variable could be subject, the baseline weight is the covariate with measurement error and the response is change from baseline in weight.

Value

An object of class 'simex' which contains:

coefficients	the corrected coefficients of the SIMEX model
SIMEX.estimates	the estimates for every lambda
model	the naive model
measurement.error	the known error standard deviations for SIMEXvariable
B	the number of iterations
extrapolation	the model object of the extrapolation step
fitting.method	the fitting method used in the extrapolation step
residuals	the residuals of the main model
fitted.values	the fitted values of the main model
call	the function call
variance.jackknife	the jackknife variance estimate

`extrapolation.variance`
: the model object of the variance extrapolation

`variance.jackknife.lambda`
: the data set for the extrapolation

`variance.asymptotic`
: the asymptotic variance estimates

`theta`
: the estimates for every B and lambda

Author(s)

John Lawrence,<john.lawrence@fda.hhs.gov>, Jianjin Xu, Wolfgang Lederer, Heidi Seibold

References

Cook, J.R. and Stefanski, L.A. (1994) Simulation-extrapolation estimation in parametric measurement error models. *Journal of American Statistical Association*, **89**, 1314 – 1328

See Also

simex,lme

Examples

```
set.seed(1234)
data("simGFRdata")
simGFR=simGFR[is.element(simGFR$time,c(1:12)/4) & is.element(simGFR$PID,c(1:80)*100),]

fm2=nlme::lme.formula(fixed = cfb ~ time + x1:time + trt + trt:time + trt:x1:time + 0,
                data = simGFR, random = ~time | PID,
                correlation = nlme::corCompSymm(0.5,form = ~time | PID, fixed = TRUE),
                control=nlme::lmeControl(returnObject=TRUE))

(s1 = simexlme(model=fm2, model.model=simGFR[,c("cfb","PID","time","x1","trt")],
                SIMEXvariable="x1",respvar="cfb",grpvar="PID",corform="~time | PID",
                measurement.error=res.sd,measurement.error.resp=res.sd,
                lambda = c(0.5,2),B = 2, fitting.method = "linear",
                jackknife.estimation = FALSE))

plot(s1)

#values of fixed effects used to simulate data
c(fixed.time,fixed.trt,fixed.leGFR,fixed.trttime,fixed.leGFRtrt)

#naive estimates
fm2$coefficients$fixed

#SIMEX corrected estimates
s1$coefficients
```

`simGFRdata` *Data Set Containing Simulated Longitudinal eGFR*

Description

A data set containing simulated values of log-eGFR measured longitudinally over time as a function of baseline eGFR. The data were simulated from a mixed effects model with the following form (using the lme model structure syntax; see format section below for definition of variables):

cfb ~ time + x1:time + trt + trt:time + trt:x1:time + 0

and these coefficients:

time trt time:x1 time:trt time:x1:trt -0.6447911 -0.0478315 0.1333391 0.2186963 -0.0458998

In addition, each subject has a random slope and intercept. The baseline eGFR were simulated from a log-Normal distribution.

Usage

```
data(simGFRdata)
```

Format

Fixed effect coefficients used to simulate the data: fix.beta fixed.leGFR fixed.leGFRtrt fixed.time fixed.trt fixed.trttime

mu.base.leGFR: mean of baseline log-eGFR var.base.leGFR: variance of baseline log-eGFR

res.sd: residual error standard deviation. note this is for a single log-eGFR, so the standard deviation for the change from baseline is sqrt(2)*res.sd and the residual error for cfb within a patient have correlation 0.5.

Variance components of random effects distribution: sig.intercept: standard deviaiton of random intercept sig.time: standrd deviation of random slope sig.cor: correlation

A data frame named simGFR that consists of fourteen columns and 28800 rows. The variables are: PID: patient ID trt: the treatment group indicator x1: measured value of baseline log-eGFR time: time from baseline measured in years alphai: subject's random intercept betai: subject's random slope alpha: subject's intercept including fixed and random effects beta: subject's slope including fixed and random effects cfb0: the measurement error for the baseline log-eGFR x: the unobserved "true" baseline log-eGFR cfb: the change from baseline in measured log-eGFR

TKVsurv *Data Set Containing Fitted Model Estimates and Covariances*

Description

A data set containing the estimates fromthe fitted Cox proportional hazards model from a dataset of patients with Autosomal Dominant Polycystic Kideny Disease. See references for further details. The model has 6 parameters describing how the hazard changes for different levels of the 3 covariates. In addition, there are 3 strata correspoding to the different imaging modalities: CT, MRI, US.

Usage

```
data(TKVsurv)
```

Format

A list with the following components covariates: names of covariates mean: sample mean of covariates stand.dev.: standard deviation of covariates labels: labels for coefficients and rows and columns of covariance matrix sigma beta: estimated coefficients in the proportional hazards model sigma: estimated covariance martix for beta CT.time: time points where Survival curve changes for CT strata CT.lcumhaz: estimated log-cumulative hazard in CT strata with coefficients = 0 CT.sig17: estimated 7 elements to fill the last row (and last column) of the covariance matrix MRI.time: time points where Survival curve changes for MRI strata MRI.lcumhaz: estimated log-cumulative hazard in MRI strata with coefficients = 0 MRI.sig17: estimated 7 elements to fill the last row (and last column) of the covariance matrix US.time: time points where Survival curve changes for US strata US.lcumhaz: estimated log-cumulative hazard in US strata with coefficients = 0 US.sig17: estimated 7 elements to fill the last row (and last column) of the covariance matrix

Author(s)

John Lawrence, Jianjin Xu, Jim Hung, Sue Jane Wang

References

http://www.fda.gov/downloads/Drugs/DevelopmentApprovalProcess/DrugDevelopmentToolsQualificationProgram/UCM45

Index