

Package ‘LongCART’

May 31, 2021

Type Package

Title Recursive Partitioning for Longitudinal Data and Right Censored Data Using Baseline Covariates

Version 3.1

Date 2021-05-31

Author Madan G Kundu

Maintainer Madan G Kundu <madan_g.kundu@yahoo.com>

Depends R (>= 3.4.0), nlme, rpart, survival, magic, survminer

Imports Formula

Description Constructs tree for continuous longitudinal data and survival data using baseline covariates as partitioning variables according to the 'LongCART' and 'SurvCART' algorithm, respectively. Later also included functions to calculate conditional power and predictive power of success based on interim results and probability of success for a prospective trial.

License GPL (>= 2)

URL <https://www.r-project.org>

BugReports <https://github.com/madanstat/LongCART/issues>

NeedsCompilation no

Repository CRAN

Date/Publication 2021-05-31 07:10:36 UTC

R topics documented:

ACTG175	2
GBSG2	3
KMPlot.SurvCART	4
LongCART	5
plot	9
PoS	10
ProfilePlot.LongCART	13
StabCat	15

StabCat.surv	16
StabCont	18
StabCont.surv	19
succ_ia	21
succ_ia_betabinom_one	26
succ_ia_betabinom_two	27
SurvCART	29
text	32

Index	35
--------------	-----------

ACTG175	<i>Converted AIDS Clinical Trials Group Study 175 (source: speff2trial package)</i>
---------	---

Description

ACTG 175 was a randomized clinical trial to compare monotherapy with zidovudine or didanosine with combination therapy with zidovudine and didanosine or zidovudine and zalcitabine in adults infected with the human immunodeficiency virus type I whose CD4 T cell counts were between 200 and 500 per cubic millimeter.

Usage

```
data(ACTG175)
```

Format

A data frame with 6417 observations from 2139 patients on the following 24 variables.

pidnum patient ID number
age age in years at baseline
wtkg weight in kg at baseline
hemo hemophilia (0=no, 1=yes)
homo homosexual activity (0=no, 1=yes)
drugs history of intravenous drug use (0=no, 1=yes)
karnof Karnofsky score (on a scale of 0-100)
oprior non-zidovudine antiretroviral therapy prior to initiation of study treatment (0=no, 1=yes)
z30 zidovudine use in the 30 days prior to treatment initiation (0=no, 1=yes)
zprior zidovudine use prior to treatment initiation (0=no, 1=yes)
preanti number of days of previously received antiretroviral therapy
race race (0=white, 1=non-white)
gender gender (0=female, 1=male)
str2 antiretroviral history (0=naive, 1=experienced)

strat antiretroviral history stratification (1:antiretroviral naive, 2:greater than 1 but less than 52 weeks of prior antiretroviral therapy, 3: greater than 52 weeks)

symptom symptomatic indicator (0=asymptomatic, 1=symptomatic)

treat treatment indicator (0=zidovudine only, 1=other therapies)

offtrt indicator of off-treatment before 96 weeks (0=no,1=yes)

r missing CD4 T cell count at 96 weeks (0=missing, 1=observed)

cens indicator of observing the event in days

days number of days until the first occurrence of: (i) a decline in CD4 T cell count of at least 50 (ii) an event indicating progression to AIDS, or (iii) death.

arms treatment arm (0=zidovudine, 1=zidovudine and didanosine, 2=zidovudine and zalcitabine, 3=didanosine)

time time in weeks

cd4 CD4 T cell count

References

Hammer, S.M., et al. (1996), A trial comparing nucleoside monotherapy with combination therapy in HIV-infected adults with CD4 cell counts from 200 to 500 per cubic millimeter. *New England Journal of Medicine*, 335:1081-1090.

GBSG2

German Breast Cancer Study Group 2 (source: TH.data package)

Description

A data frame containing the observations from the GBSG2 study.

Usage

```
data(GBSG2)
```

Format

A data frame with 686 observations on the following 10 variables.

horTh hormonal therapy, a factor with levels no yes

age age in years

menostat menopausal status, a factor with levels Pre Post

tsize tumor size (in mm)

tgrade an ordered factor with levels I < II < III

pnodes number of positive nodes

progrec progesterone receptor (in fmol).

estrec estrogen receptor (in fmol).

time recurrence free survival time (in days).

cens censoring indicator (0- censored, 1- event).

References

Schumacher M, Bastert G, Bojar H, Huebner K, Olschewski M, Sauerbrei W, Schmoor C, Beyerle C, Neumann RL, Rauschecker HF. Randomized 2 x 2 trial evaluating hormonal treatment and the duration of chemotherapy in node-positive breast cancer patients. German Breast Cancer Study Group. *Journal of Clinical Oncology*. 1994 Oct;12(10):2086-93.

Examples

```
data(GBSG2)
```

KMPlot.SurvCART	<i>KM plot for SurvCART object</i>
-----------------	------------------------------------

Description

Generates KM plot for sub-groups (i.e., terminal nodes) associated with survival tree generated by SurvCART()

Usage

```
KMPlot.SurvCART(x, scale.time = 1, type = 1, ...)
```

Arguments

x	a fitted object of class "SurvCART", containing a survival tree.
scale.time	Divides the time variable by the factor of scale.time. For example, if times are collected in days, then to plot time in years, specify 365.25.
type	1 for KM plot of survival probabilities, 2 for KM plot of censoring probabilities
...	arguments to be passed to or from other methods.

Author(s)

Madan Gopal Kundu <madan_g.kundu@yahoo.com>

References

Kundu, M. G.(2020). Survival trees based on score based parameter instability test.

See Also

[text.SurvCART](#), [plot.SurvCART](#), [SurvCART](#), [StabCat.surv](#), [StabCont.surv](#)

Examples

```

#--- Get the data
data(GBSG2)

#numeric coding of character variables
GBSG2$horTh1<- as.numeric(GBSG2$horTh)
GBSG2$tgrade1<- as.numeric(GBSG2$tgrade)
GBSG2$menostat1<- as.numeric(GBSG2$menostat)

#Add subject id
GBSG2$subjid<- 1:nrow(GBSG2)

#--- Run SurvCART() with time-to-event distribution: exponential, censoring distribution: None
out<- SurvCART(data=GBSG2, patid="subjid", censorvar="cens", timevar="time",
  gvars=c('horTh1', 'age', 'menostat1', 'tsize', 'tgrade1', 'pnodes', 'progrec', 'estrec'),
  tgvars=c(0,1,0,1,0,1, 1,1),
  event.ind=1, alpha=0.05, minsplit=80, minbucket=40, print=TRUE)

#--- Plot tree
par(xpd = TRUE)
plot(out, compress = TRUE)
text(out, use.n = TRUE)

#Plot KM plot of survival probabilities for sub-groups identified by tree
KMPlot.SurvCART(out, scale.time=365.25, type=1)

#Plot KM plot of censoring probabilities for sub-groups identified by tree
KMPlot.SurvCART(out, scale.time=365.25, type=1)

```

LongCART

Longitudinal CART with continuous response via binary partitioning

Description

Recursive partitioning for linear mixed effects model with continuous univariate response variables per LonCART algorithm based on baseline partitioning variables (Kundu and Harezlak, 2019).

Usage

```

LongCART(data, patid, fixed, gvars, tgvars, minsplit=40,
  minbucket=20, alpha=0.05, coef.digits=2, print.lme=FALSE)

```

Arguments

data	name of the dataset. It must contain variable specified for patid (indicating subject id), all the variables specified in the formula and the baseline partitioning variables.
patid	name of the subject id variable.

<code>fixed</code>	a two-sided linear formula object describing the fixed-effects part of the model, with the response on the left of a <code>~</code> operator and the terms, separated by <code>+</code> operators, on the right. Model with <code>-1</code> to the end of right side indicates no intercept. For model with no fixed effect beyond intercept, please specify only <code>1</code> right to the <code>~</code> operator.
<code>gvars</code>	list of partitioning variables of interest. Value of these variables should not change over time. Regarding categorical variables, only numerically coded categorical variables should be specified. For nominal categorical variables or factors, please first create corresponding dummy variable(s) and then pass through <code>gvars</code> .
<code>tgvars</code>	types (categorical or continuous) of partitioning variables specified in <code>gvar</code> . For each of continuous partitioning variables, specify <code>1</code> and for each of the categorical partitioning variables, specify <code>0</code> . Length of <code>tgvars</code> should match to the length of <code>gvars</code>
<code>minsplit</code>	the minimum number of observations that must exist in a node in order for a split to be attempted.
<code>minbucket</code>	the minimum number of observations in any terminal node.
<code>alpha</code>	alpha (i.e., nominal type I error) level for parameter instability test
<code>coef.digits</code>	decimal points for displaying coefficients in the tree structure.
<code>print.lme</code>	if TRUE, then summary of fitted model from <code>lme()</code> will be printed for each node.

Details

Construct regression tree based on heterogeneity in linear mixed effects models of following type: $Y_i(t) = W_i(t)\theta + b_i + \epsilon_{it}$ where $W_i(t)$ is the design matrix, θ is the parameter associated with $W_i(t)$ and b_i is the random intercept. Also, $\epsilon_{it} \sim N(0, \sigma^2)$ and $b_i \sim N(0, \sigma_u^2)$.

Value

<code>Treeout</code>	contains summary information of tree fitting for each terminal nodes and non-terminal nodes. Columns of <code>Treeout</code> include "ID", the (unique) node numbers that follow a binary ordering indexed by node depth, <code>n</code> , the number of observations reaching the node, <code>yval</code> , the fitted model of the response at the node, <code>var</code> , a factor giving the names of the variables used in the split at each, <code>index</code> , the cut-off value of splitting variable for binary partitioning, <code>p</code> (Instability), the p-value for parameter instability test for the splitting variable, <code>loglik</code> , the log-likelihood of the node, <code>improve</code> , the improvement in deviance given by this split, and <code>Terminal</code> , indicator (True or False) of terminal node.
<code>p</code>	number of fixed parameters
<code>AIC.tree</code>	AIC of the tree-structured model
<code>AIC.root</code>	AIC at the root node (i.e., without tree structure)
<code>improve.AIC</code>	improvement in AIC due to tree structure (<code>AIC.tree - AIC.root</code>)
<code>logLik.tree</code>	log-likelihood of the tree-structured model
<code>logLik.root</code>	log-likelihood at the root node (i.e., without tree structure)

Deviance	$2 * (\log\text{Lik.tree} - \log\text{Lik.root})$
LRT.df	degrees of freedom for likelihood ratio test comparing tree-structured model with the model at root node.
LRT.p	p-value for likelihood ratio test comparing tree-structured model with the model at root node.
subj.class	Assigned node for each individual subjects per fitted longitudinal tree
frame	rpart compatible object
splits	rpart compatible object
cptable	rpart compatible object
functions	rpart compatible object

Author(s)

Madan Gopal Kundu <madan_g.kundu@yahoo.com>

References

Kundu, M. G., and Harezlak, J. (2019). Regression trees for longitudinal data with baseline covariates. *Biostatistics & Epidemiology*, 3(1):1-22.

See Also

[plot.LongCART](#), [text.LongCART](#), [ProfilePlot.LongCART](#), [StabCat](#), [StabCont](#)

Examples

```
#--- Get the data
data(ACTG175)

#-----#
#  model: cd4~ time + subject(random)          #
#-----#

#--- Run LongCART()
gvars=c("gender", "wtkg", "hemo", "homo", "drugs",
        "karnof", "oprior", "z30", "zprior", "race",
        "str2", "symptom", "treat", "offtrt")
tgvars=c(0, 1, 0, 0, 0,
         1, 0, 0, 0, 0,
         0, 0, 0, 0)

out1<- LongCART(data=ACTG175, patid="pidnum", fixed=cd4~time,
               gvars=gvars, tgvars=tgvars, alpha=0.05,
               minsplit=100, minbucket=50, coef.digits=2)

#--- Plot tree
par(mfrow=c(1,1))
```

```

par(xpd = TRUE)
plot(out1, compress = TRUE)
text(out1, use.n = TRUE)

#--- Plot longitudinal profiles of subgroups
par(mfrow=c(3,3))
ProfilePlot.LongCART(x=out1, timevar="time")

#-----#
#  model: cd4~ time+ time^2 + subject(random) #
#-----#

ACTG175$time2<- ACTG175$time^2

out2<- LongCART(data=ACTG175, patid="pidnum", fixed=cd4~time + time2,
               gvars=gvars, tgvars=tgvars, alpha=0.05,
               minsplit=100, minbucket=50, coef.digits=2)

par(mfrow=c(1,1))
par(xpd = TRUE)
plot(out2, compress = TRUE)
text(out2, use.n = TRUE)

par(mfrow=c(3,3))
ProfilePlot.LongCART(x=out2, timevar="time", timevar.power=c(1,2))

#-----#
#  model: cd4~ time+ time^2 + subject(random) + karnof #
#-----#

out3<- LongCART(data=ACTG175, patid="pidnum", fixed=cd4~time + time2 + karnof,
               gvars=gvars, tgvars=tgvars, alpha=0.05,
               minsplit=100, minbucket=50, coef.digits=2)

par(mfrow=c(1,1))
par(xpd = TRUE)
plot(out3, compress = TRUE)
text(out3, use.n = TRUE)

#the value of the covariate karnof is set at median by default
par(mfrow=c(3,3))
ProfilePlot.LongCART(x=out3, timevar="time", timevar.power=c(1,2, NA))

#the value of the covariate karnof is set at 120
par(mfrow=c(3,3))
ProfilePlot.LongCART(x=out3, timevar="time", timevar.power=c(1,2, NA),
                    covariate.val=c(NA, NA, 120))

```

plot

Plot an SurvCART or LongCART Object

Description

Plots an SurvCART or LongCART object on the current graphics device.

Usage

```
## S3 method for class 'SurvCART'
plot(x, uniform = FALSE, branch = 1, compress = FALSE,
      nspace = branch, margin = 0, minbranch = 0.3, ...)
## S3 method for class 'LongCART'
plot(x, uniform = FALSE, branch = 1, compress = FALSE,
      nspace = branch, margin = 0, minbranch = 0.3, ...)
```

Arguments

x	a fitted object of class "SurvCART", containing a survival tree or "LongCART", containing a longitudinal tree.
uniform	similar to <code>plot.rpart</code> ; if TRUE, uniform vertical spacing of the nodes is used; this may be less cluttered when fitting a large plot onto a page. The default is to use a non-uniform spacing proportional to the error in the fit.
branch	similar to <code>plot.rpart</code> ; controls the shape of the branches from parent to child node. Any number from 0 to 1 is allowed. A value of 1 gives square shouldered branches, a value of 0 give V shaped branches, with other values being intermediate.
compress	similar to <code>plot.rpart</code> ; if FALSE, the leaf nodes will be at the horizontal plot coordinates of 1:nleaves. If TRUE, the routine attempts a more compact arrangement of the tree.
nspace	similar to <code>plot.rpart</code> ; the amount of extra space between a node with children and a leaf, as compared to the minimal space between leaves. Applies to compressed trees only. The default is the value of <code>branch</code> .
margin	similar to <code>plot.rpart</code> ; an extra fraction of white space to leave around the borders of the tree. (Long labels sometimes get cut off by the default computation).
minbranch	similar to <code>plot.rpart</code> ; set the minimum length for a branch to <code>minbranch</code> times the average branch length. This parameter is ignored if <code>uniform=TRUE</code> . Sometimes a split will give very little improvement, or even (in the classification case) no improvement at all. A tree with branch lengths strictly proportional to improvement leaves no room to squeeze in node labels.
...	arguments to be passed to or from other methods.

Details

This function is a method for the generic function `plot`, for objects of class `SurvCART`. The y-coordinate of the top node of the tree will always be 1.

Value

The coordinates of the nodes are returned as a list, with components x and y.

Author(s)

Madan Gopal Kundu <madan_g.kundu@yahoo.com>

References

Kundu, M. G.(2020). Survival trees based on score based parameter instability test.

See Also

[text](#), [SurvCART](#), [LongCART](#)

Examples

```
#--- Get the data
data(GBSG2)

#numeric coding of character variables
GBSG2$horTh1<- as.numeric(GBSG2$horTh)
GBSG2$tgrade1<- as.numeric(GBSG2$tgrade)
GBSG2$menostat1<- as.numeric(GBSG2$menostat)

#Add subject id
GBSG2$subjid<- 1:nrow(GBSG2)

#--- Run SurvCART()
out<- SurvCART(data=GBSG2, patid="subjid", censorvar="cens", timevar="time", event.ind=1,
  gvars=c('horTh1', 'age', 'menostat1', 'tsize', 'tgrade1', 'pnodes', 'progrec', 'estrec'),
  tgvars=c(0,1,0,1,0,1, 1,1),
  alpha=0.05, minsplit=80,
  minbucket=40, print=TRUE)

#--- Plot tree
par(xpd = TRUE)
plot(out, compress = TRUE)
text(out, use.n = TRUE)
```

PoS

Probability of trial and clinical success for a prospective trial using normal-normal approximation

Description

This function can be used to determine probability of trial success and clinical success based on the prior distribution for each of continuous, binary and time-to-event endpoints. The calculation is carried out assuming normal distribution for estimated parameter and normal prior distribution.

Usage

```
PoS(type, nsamples, null.value = NULL, alternative = "greater",
     N = NULL, D = NULL, a = 1,
     succ.crit = "trial", Z.crit.final = 1.96, alpha.final = 0.025,
     clin.succ.threshold = NULL, se.exp = NULL,
     meandiff.prior = NULL, mean.prior = NULL, sd.prior = NULL,
     proppdiff.prior = NULL, prop.prior = NULL, hr.prior = NULL, D.prior = NULL)
```

Arguments

<code>type</code>	Type of the endpoint. It could be <code>cont</code> for continuous, <code>bin</code> for binary and <code>surv</code> for survival endpoint.
<code>nsamples</code>	Number of samples. For continuous and binary case, it can be 1 or 2. For survival endpoint, it can be only 2.
<code>null.value</code>	The specified value under null hypothesis. Default is 0 for continuous and binomial case and 1 for survival case.
<code>alternative</code>	Direction of alternate hypothesis. Can be "greater" or "less". Default is "less" for test of HR and "greater" otherwise.
<code>N</code>	Total sample size at final analysis. Cannot be missing for continuous and binary endpoint.
<code>D</code>	Total number of events at final analysis. Cannot be missing for survival endpoint.
<code>a</code>	Allocation ratio in two sample case.
<code>succ.crit</code>	Specify "trial" for trial success (i.e., null hypothesis is rejected at final analysis) or "clinical" for clinical success (i.e., estimated value at the final analysis is greater than clinically meaningful value as specified under <code>clin.succ.threshold</code> .)
<code>Z.crit.final</code>	The rejection boundary at final analysis in Z-value scale. Either <code>alpha.final</code> or <code>Z.crit.final</code> must be specified when determining trial success.
<code>alpha.final</code>	The rejection boundary at final analysis in alpha (1-sided) scale (e.g., 0.025). Either <code>alpha.final</code> or <code>Z.crit.final</code> must be specified when determining trial success.
<code>clin.succ.threshold</code>	Clinically meaningful value. Required when <code>succ.crit="clinical"</code> .
<code>se.exp</code>	Expected standard error to be observed in the study. Must be specified in continuous case and two-sample binary case.
<code>meandiff.prior</code>	Mean value of prior distribution for mean difference. Relevant for two-sample continuous case.
<code>mean.prior</code>	Mean value of prior distribution for mean. Relevant for one-sample continuous case.
<code>sd.prior</code>	Standard deviation of prior distribution for mean difference (2-sample continuous case) or mean (1-sample continuous case) or prop (2-sample binary case) or difference of proportion (1-sample binary case) or log(HR) (2 sample survival case).

<code>proppdiff.prior</code>	Mean value of prior distribution for difference in proportion. Relevant for two-sample binomial case.
<code>prop.prior</code>	Mean value of prior distribution for proportion. Relevant for one-sample binomial case.
<code>hr.prior</code>	Mean value of prior distribution for hazards ratio (HR). Relevant for two-sample survival case.
<code>D.prior</code>	Ignored if <code>sd.prior</code> is specified. If <code>sd.prior</code> is not specified then <code>sd.prior</code> is determined as $2/D.prior$. Relevant for two-sample survival case.

Details

This function can be used to determine probability of success (PoS) for a prospective trial for each of continuous (one-sample or two-samples), binary (one-sample or two-samples) and time-to-event endpoints (two-samples). The PoS is calculated based on the prior distribution and expected standard error in the estimate in trial. The calculation PoS is carried out assuming normal distribution for estimated parameter and normal prior distribution. This function can be used to determine clinical success (`succ.crit="clinical"`) and trial success (`succ.crit="trial"`). For clinical success, `clin.succ.threshold` must be specified. For trial success, `Z.crit.final` or `alpha.final` must be specified.

Author(s)

Madan Gopal Kundu <madan_g.kundu@yahoo.com>

See Also

[succ_ia_betabinom_one](#), [succ_ia_betabinom_two](#), [succ_ia](#)

Examples

```
#--- Example 1
PoS(type="cont", nsamples=2, null.value=-0.05, alternative="greater",
     N=1552, a=1,
     succ.crit="trial", Z.crit.final=1.97,
     se.exp=0.12*sqrt(1/776 + 1/776),
     meandiff.prior=0, sd.prior=0.02)
```

```
#--- Example 2
PoS(type="bin", nsamples=2, null.value=0, alternative="greater",
     N=210, a=2,
     succ.crit="trial", Z.crit.final=2.012,
     se.exp=0.5*sqrt(1/140 + 1/70),
     proppdiff.prior=0.20, sd.prior=sqrt(0.06))
```

```
PoS(type="bin", nsamples=2, null.value=0, alternative="greater",
     N=210, a=2,
     succ.crit="clinical", clin.succ.threshold =0.15,
     se.exp=0.5*sqrt(1/140 + 1/70),
     proppdiff.prior=0.20, sd.prior=sqrt(0.06))
```

```

#--- Example 4
PoS(type="surv", nsamples=2, null.value=1, alternative="less",
     D=441,
     succ.crit="trial", Z.crit.final=1.96,
     hr.prior=0.71, D.prior=133)

PoS(type="surv", nsamples=2, null.value=1, alternative="less",
     D=441,
     succ.crit="clinical", clin.succ.threshold =0.8,
     hr.prior=0.71, D.prior=133)

```

ProfilePlot.LongCART *Population level longitudinal profile plot for sub-groups*

Description

Generates population level longitudinal profile plot for each of sub-groups (i.e., terminal nodes) associated with longitudinal tree generated by LongCART()

Usage

```
ProfilePlot.LongCART(x, timevar, timevar.power=NULL, covariate.val=NULL,
                    xlab=NULL, ylab=NULL, sg.title=3, ...)
```

Arguments

x	a fitted object of class "LongCART", containing a longitudinal tree.
timevar	Specify the variable name containing time information in the dataset that was used to fit LongCART object
timevar.power	Mandatory when the fixed part of the fitted model contains term as time with power not equal to 1. For example, if fixed part of the model is $t + \sqrt{t} + cov1$, then specify $c(1, 0.5, NA)$. If fixed part of the model is $t + t^2 + cov1$, then specify $c(1, 2, NA)$.
covariate.val	Specify the covariate values for generation of longitudinal profile plot. In the longitudinal profile plot, only time can vary and therefore, and therefore the value for the other covariates are fixed at constant value. This is not needed if the longitudinal model does not contain additional covariate(s). By default, the covariates values are specified at median value over all the datapoint (not at the subject level). For example, if the fixed part of the model is $t + cov1$, then $c(NA, 100)$ sets the value of $cov1$ at 100. Similarly, If fixed part of the model is $t + t^2 + cov1$, then $c(NA, NA, 100)$ would be acceptable.
xlab	Optional label for X-axis
ylab	Optional label for Y-axis
sg.title	1 for sub-groups' title as Sub-group=x, 2 for sub-groups' title as Node=x, and 3 for sub-groups' title as Sub-group=x (Node=x).
...	Graphical parameters other than x, y, type, xlab, ylab.

Author(s)

Madan Gopal Kundu <madan_g.kundu@yahoo.com>

References

Kundu, M. G., and Harezlak, J. (2019). Regression trees for longitudinal data with baseline covariates. *Biostatistics & Epidemiology*, 3(1):1-22.

See Also

[text.LongCART](#), [plot.LongCART](#), [LongCART](#)

Examples

```
#--- Get the data
data(ACTG175)

#-----#
#  model: cd4~ time + subject(random)      #
#-----#

#--- Run LongCART()
gvars=c("gender", "wtkg", "hemo", "homo", "drugs",
        "karnof", "oprior", "z30", "zprior", "race",
        "str2", "symptom", "treat", "offtrt")
tgvars=c(0, 1, 0, 0, 0,
        1, 0, 0, 0, 0,
        0, 0, 0, 0)

out1<- LongCART(data=ACTG175, patid="pidnum", fixed=cd4~time,
               gvars=gvars, tgvars=tgvars, alpha=0.05,
               minsplit=100, minbucket=50, coef.digits=2)

#--- Plot longitudinal profiles of subgroups
par(mfrow=c(3,3))
ProfilePlot.LongCART(x=out1, timevar="time")

#-----#
#  model: cd4~ time+ time^2 + subject(random) #
#-----#

ACTG175$time2<- ACTG175$time^2

out2<- LongCART(data=ACTG175, patid="pidnum", fixed=cd4~time + time2,
               gvars=gvars, tgvars=tgvars, alpha=0.05,
               minsplit=100, minbucket=50, coef.digits=2)

par(mfrow=c(3,3))
ProfilePlot.LongCART(x=out2, timevar="time", timevar.power=c(1,2))
```

```

#-----#
# model: cd4~ time+ time^2 + subject(random) + karnof #
#-----#

out3<- LongCART(data=ACTG175, patid="pidnum", fixed=cd4~time + time2 + karnof,
               gvars=gvars, tgvars=tgvars, alpha=0.05,
               minsplit=100, minbucket=50, coef.digits=2)

#the value of the covariate karnof is set at median by default
par(mfrow=c(3,3))
ProfilePlot.LongCART(x=out3, timevar="time", timevar.power=c(1,2, NA))

#the value of the covariate karnof is set at 120
par(mfrow=c(3,3))
ProfilePlot.LongCART(x=out3, timevar="time", timevar.power=c(1,2, NA),
                    covariate.val=c(NA, NA, 120))

```

StabCat

*parameter stability test for categorical partitioning variable***Description**

Performs parameter stability test (Kundu and Harezlak, 2019) with categorical partitioning variable to determine whether the parameters of linear mixed effects model remains same across all distinct values of given categorical partitioning variable.

Usage

```
StabCat(data, patid, fixed, splitvar)
```

Arguments

data	name of the dataset. It must contain variable specified for patid (indicating subject id) and all the variables specified in the formula and the categorical partitioning variable of interest specified in splitvar. Note that, only numerically coded categorical variable should be specified.
patid	name of the subject id variable.
fixed	a two-sided linear formula object describing the fixed-effects part of the model, with the response on the left of a ~ operator and the terms, separated by + operators, on the right. Model with -1 to the end of right side indicates no intercept. For model with no fixed effect beyond intercept, please specify only 1 right to the ~ operator.
splitvar	the categorical partitioning variable of interest. It's value should not change over time.

Details

The categorical partitioning variable of interest. It's value should not change over time.

$$Y_i(t) = W_i(t) \theta + b_i + \epsilon_{it}$$

where $W_i(t)$ is the design matrix, θ is the parameter associated with $W_i(t)$ and b_i is the random intercept. Also, $\epsilon_{it} \sim N(0, \sigma^2)$ and $b_i \sim N(0, \sigma_u^2)$. Let X be the baseline categorical partitioning variable of interest. `StabCat()` performs the following omnibus test

$$H_0: \theta_{(g)} = \theta_0 \text{ vs. } H_1: \theta_{(g)} \neq \theta_0, \text{ for all } g$$

where, $\theta_{(g)}$ is the true value of θ for subjects with $X=C_g$ where C_g is the any value realized by X .

Value

`p` It returns the p-value for parameter instability test

Author(s)

Madan Gopal Kundu <madan_g.kundu@yahoo.com>

References

Kundu, M. G., and Harezlak, J. (2019). Regression trees for longitudinal data with baseline covariates. *Biostatistics & Epidemiology*, 3(1):1-22.

See Also

[StabCont](#), [LongCART](#), [plot.LongCART](#), [text.LongCART](#)

Examples

```
#--- Get the data
data(ACTG175)

#--- Run StabCat()
out<- StabCat(data=ACTG175, patid="pidnum", fixed=cd4~time, splitvar="gender")
out$pval
```

StabCat.surv

parameter stability test for categorical partitioning variable

Description

Performs parameter stability test (Kundu, 2020) with categorical partitioning variable to determine whether the parameters of exponential time-to-event distribution and exponential censoring distribution remain same across all distinct values of given categorical partitioning variable.

Usage

```
StabCat.surv(data, timevar, censorvar, splitvar,
             time.dist="exponential", cens.dist="NA", event.ind=1, print=FALSE)
```

Arguments

<code>data</code>	name of the dataset. It must contain variable specified for <code>timevar</code> (indicating follow-up times), <code>censorvar</code> (indicating censoring status) and the categorical partitioning variable of interest specified in <code>splitvar</code> . Note that, only numerically coded categorical variable should be specified.
<code>timevar</code>	name of the variable with follow-up times.
<code>censorvar</code>	name of the variable with censoring status.
<code>time.dist</code>	name of time-to-event distribution. It can be one of the following distributions: "exponential", "weibull", "lognormal" or "normal".
<code>cens.dist</code>	name of censoring distribution. It can be one of the following distributions: "exponential", "weibull", "lognormal", "normal" or "NA". If specified "NA", then parameter instability test corresponding to censoring distribution will not be performed.
<code>event.ind</code>	value of the censoring variable indicating event.
<code>splitvar</code>	the categorical partitioning variable of interest. It's value should not change over time.
<code>print</code>	if TRUE, then additional information including estimated parameters, score function and its variance will be printed.

Details

`StabCat.surv()` performs the following omnibus test

$H_0: \lambda_{\{g\}} = \lambda_0$ vs. $H_1: \lambda_{\{g\}} \neq \lambda_0$, for all g

where, $\theta_{\{g\}}$ is the true value of θ for subjects with $X=C_g$. θ includes all the parameters of time to event distribution and also parameters of censoring distribution, if specified. C_g is the any value realized by categorical partitioning variable X .

Exponential distribution: $f(t) = \lambda \exp(-\lambda t)$

Weibull distribution: $f(t) = \alpha \lambda t^{\alpha-1} \exp(-\lambda t^\alpha)$

Lognormal distribution: $f(t) = (1/t) * (1/\sqrt{2*\pi*\sigma^2}) * \exp[-(1/2)*(log(t)-\mu)/\sigma^2]$

Normal distribution: $f(t) = (1/\sqrt{2*\pi*\sigma^2}) * \exp[-(1/2)*(t-\mu)/\sigma^2]$

Value

<code>pval</code>	p-value for parameter instability test
<code>type</code>	1, if event times are more heterogeneous; 2, if censoring times are more heterogeneous.

Author(s)

Madan Gopal Kundu <madan_g.kundu@yahoo.com>

References

Kundu, M. G.(2020). Survival trees based on score based parameter instability test.

See Also

[StabCont.surv](#), [SurvCART](#), [plot.SurvCART](#), [text.SurvCART](#)

Examples

```
#--- time-to-event distribution: exponential, censoring distribution: None
out1<- StabCat.surv(data=lung, timevar="time", censorvar="status", splitvar="sex", event.ind=2)
out1$pval

#--- time-to-event distribution: weibull, censoring distribution: None
StabCat.surv(data=lung, timevar="time", censorvar="status", splitvar="sex",
             time.dist="weibull", event.ind=2)

#--- time-to-event distribution: weibull, censoring distribution: exponential
StabCat.surv(data=lung, timevar="time", censorvar="status", splitvar="sex",
             time.dist="weibull", cens.dist="exponential", event.ind=2)
```

StabCont

parameter stability test for continuous partitioning variable

Description

Performs parameter stability test (Kundu and Harezlak, 2019) with continuous partitioning variable to determine whether the parameters of linear mixed effects model remains same across all distinct values of given continuous partitioning variable.

Usage

```
StabCont(data, patid, fixed, splitvar)
```

Arguments

<code>data</code>	name of the dataset. It must contain variable specified for <code>patid</code> (indicating subject id) and all the variables specified in the formula and the <code>StabCont(data, fixed, splitvar)</code> partitioning variable of interest specified in <code>splitvar</code> .
<code>patid</code>	name of the subject id variable.
<code>fixed</code>	a two-sided linear formula object describing the fixed-effects part of the model, with the response on the left of a <code>~</code> operator and the terms, separated by <code>+</code> operators, on the right. Model with <code>-1</code> to the end of right side indicates no intercept. For model with no fixed effect beyond intercept, please specify only <code>1</code> right to the <code>~</code> operator.
<code>splitvar</code>	the continuous partitioning variable of interest. It's value should not change over time.

Details

The continuous partitioning variable of interest. It's value should not change over time.

$$Y_i(t) = W_i(t)\theta + b_i + \epsilon_{it}$$

where $W_i(t)$ is the design matrix, θ is the parameter associated with $W_i(t)$ and b_i is the random intercept. Also, $\epsilon_{it} \sim N(0, \sigma^2)$ and $b_i \sim N(0, \sigma_u^2)$. Let X be the baseline continuous partitioning variable of interest. `StabCont()` performs the following omnibus test

$H_0 : \theta_{(g)} = \theta_0$ vs. $H_1 : \theta_{(g)} \neq \theta_0$, for all g

where, $\theta_{(g)}$ is the true value of θ for subjects with $X = C_g$ where C_g is the any value realized by X .

Value

`p` It returns the p-value for parameter instability test

Author(s)

Madan Gopal Kundu <madan_g.kundu@yahoo.com>

References

Kundu, M. G., and Harezlak, J. (2019). Regression trees for longitudinal data with baseline covariates. *Biostatistics & Epidemiology*, 3(1):1-22.

See Also

[StabCont](#), [LongCART](#), [plot.LongCART](#), [text.LongCART](#)

Examples

```
#--- Get the data
data(ACTG175)

#--- Run StabCont()
out<- StabCont(data=ACTG175, patid="pidnum", fixed=cd4~time, splitvar="age")
out$pval
```

StabCont.surv

parameter stability test for continuous partitioning variable

Description

Performs parameter stability test (Kundu, 2020) with continuous partitioning variable to determine whether the parameters of exponential time-to-event distribution and exponential censoring distribution remain same across all distinct values of given continuous partitioning variable.

Usage

```
StabCont.surv(data, timevar, censorvar, splitvar,
              time.dist="exponential", cens.dist="NA", event.ind=1, print=FALSE)
```

Arguments

<code>data</code>	name of the dataset. It must contain variable specified for <code>timevar</code> (indicating follow-up times), <code>censorvar</code> (indicating censoring status) and the categorical partitioning variable of interest specified in <code>splitvar</code> . Note that, only numerically coded categorical variable should be specified.
<code>timevar</code>	name of the variable with follow-up times.
<code>censorvar</code>	name of the variable with censoring status.
<code>time.dist</code>	name of time-to-event distribution. It can be one of the following distributions: "exponential", "weibull", "lognormal" or "normal".
<code>cens.dist</code>	name of censoring distribution. It can be one of the following distributions: "exponential", "weibull", "lognormal", "normal" or "NA". If specified "NA", then parameter instability test corresponding to censoring distribution will not be performed.
<code>event.ind</code>	value of the censoring variable indicating event.
<code>splitvar</code>	the continuous partitioning variable of interest.
<code>print</code>	if TRUE, then additional information including estimated parameters, score function and its variance will be printed.

Details

`StabCont.surv()` performs the following omnibus test

$H_0: \theta_{\{g\}} = \theta_0$ vs. $H_1: \theta_{\{g\}} \neq \theta_0$, for all g

where, $\theta_{\{g\}}$ is the true value of θ for subjects with $X=C_g$. θ includes all the parameters of time to event distribution and also parameters of censoring distribution, if specified. C_g is the any value realized by continuous partitioning variable X .

Exponential distribution: $f(t) = \lambda \exp(-\lambda t)$

Weibull distribution: $f(t) = \alpha \lambda t^{\alpha-1} \exp(-\lambda t^\alpha)$

Lognormal distribution: $f(t) = (1/t) * (1/\sqrt{2*\pi*\sigma^2}) * \exp[-(1/2)*(log(t)-\mu)/\sigma^2]$

Normal distribution: $f(t) = (1/\sqrt{2*\pi*\sigma^2}) * \exp[-(1/2)*(t-\mu)/\sigma^2]$

Value

<code>pval</code>	p-value for parameter instability test
<code>type</code>	1, if event times are more heterogeneous; 2, if censoring times are more heterogeneous.

Author(s)

Madan Gopal Kundu <madan_g.kundu@yahoo.com>

References

Kundu, M. G.(2020). Survival trees based on score based parameter instability test.

See Also

[StabCont.surv](#), [SurvCART](#), [plot.SurvCART](#), [text.SurvCART](#)

Examples

```
#--- time-to-event distribution: exponential, censoring distribution: None
out1<- StabCont.surv(data=lung, timevar="time", censorvar="status", splitvar="age", event.ind=2)
out1$pval

#--- time-to-event distribution: weibull, censoring distribution: None
StabCont.surv(data=lung, timevar="time", censorvar="status", splitvar="age",
              time.dist="weibull", event.ind=2)

#--- time-to-event distribution: weibull, censoring distribution: exponential
StabCont.surv(data=lung, timevar="time", censorvar="status", splitvar="age",
              time.dist="weibull", cens.dist="exponential", event.ind=2)
```

succ_ia

Conditional power and predictive power of success based on interim results using normal-normal approximation

Description

This function can be used to determine conditional power and predictive power for trial success and clinical success based on the interim results and prior distribution for each of continuous, binary and time-to-event endpoints. The calculation is carried out assuming normal distribution for estimated parameter and normal prior distribution.

Usage

```
succ_ia(type, nsamples, null.value = NULL, alternative = NULL,
        N = NULL, n = NULL, D = NULL, d = NULL, a = 1,
        meandiff.ia = NULL, mean.ia = NULL,
        propdiff.ia = NULL, prop.ia = NULL, hr.ia = NULL,
        stderr.ia = NULL, sd.ia = NULL,
        succ.crit = "trial", Z.crit.final = 1.96,
        alpha.final = 0.025, clin.succ.threshold = NULL,
        meandiff.exp = NULL, mean.exp = NULL,
        propdiff.exp = NULL, prop.exp = NULL, hr.exp = NULL,
        meandiff.prior = NULL, mean.prior = NULL, sd.prior = NULL,
        propdiff.prior = NULL, prop.prior = NULL, hr.prior = NULL, D.prior = NULL)
```

Arguments

<code>type</code>	Type of the endpoint. It could be <code>cont</code> for continuous, <code>bin</code> for binary and <code>surv</code> for survival endpoint.
<code>nsamples</code>	Number of samples. For continuous and binary case, it can be 1 or 2. For survival endpoint, it can be only 2.
<code>null.value</code>	The specified value under null hypothesis. Default is 0 for continuous and binomial case and 1 for survival case.
<code>alternative</code>	Direction of alternate hypothesis. Can be "greater" or "less". Default is "less" for test of HR and "greater" otherwise.
<code>N</code>	Total sample size at final analysis. Cannot be missing for continuous and binary endpoint.
<code>n</code>	Total sample size at interim analysis. Cannot be missing for continuous and binary endpoint.
<code>D</code>	Total number of events at final analysis. Cannot be missing for survival endpoint.
<code>d</code>	Total number of events at interim analysis. Cannot be missing for survival endpoint.
<code>a</code>	Allocation ratio in two sample case.
<code>meandiff.ia</code>	Estimated mean difference at interim analysis. Mandatory for continuous two sample case.
<code>mean.ia</code>	Estimated mean value at interim analysis. Mandatory for continuous single sample case
<code>propdiff.ia</code>	Estimated difference in proportion at interim analysis. Mandatory for binary two sample case
<code>prop.ia</code>	Estimated proportion at interim analysis. Mandatory for binary single sample case
<code>hr.ia</code>	Estimate hazards ratio (HR) at interim analysis. Mandatory for continuous single sample case
<code>stderr.ia</code>	Standard error (SE) of estimated mean difference (in one-sample continuous case) or estimated mean (in two-sample continuous case) or estimated difference in proportion (in two-sample binary case) at interim analysis. For continuous case, if not specified, then the function attempts to estimate SE from <code>sd.ia</code> . Mandatory for two-sample binary case.
<code>sd.ia</code>	Standard deviation of estimated mean difference (in one-sample continuous case) or estimated mean (in two-sample continuous case) at interim analysis. If <code>stderr.ia</code> is specified, then the value of <code>sd.ia</code> is ignored. If <code>codestderr.ia</code> is not specified, then mandatory for continuous case.
<code>succ.crit</code>	Specify "trial" for trial success (i.e., null hypothesis is rejected at final analysis) or "clinical" for clinical success (i.e., estimated value at the final analysis is greater than clinically meaningful value as specified under <code>clin.succ.threshold</code> .)
<code>Z.crit.final</code>	The rejection boundary at final analysis in Z-value scale. Either <code>alpha.final</code> or <code>Z.crit.final</code> must be specified when determining trial success.

<code>alpha.final</code>	The rejection boundary at final analysis in alpha (1-sided) scale (e.g., 0.025). Either <code>alpha.final</code> or <code>Z.crit.final</code> must be specified when determining trial success.
<code>clin.succ.threshold</code>	Clinically meaningful value. Required when <code>succ.crit="clinical"</code> .
<code>meandiff.exp</code>	Expected mean difference in post interim data. Relevant for two-sample continuous case.
<code>mean.exp</code>	Expected mean in post interim data. Relevant for one-sample continuous case.
<code>propdiff.exp</code>	Expected difference in proportion in post interim data. Relevant for two-sample binary case.
<code>prop.exp</code>	Expected proportion in post interim data. Relevant for one-sample binary case.
<code>hr.exp</code>	Expected hazards ratio (HR) in post interim data. Relevant for two-sample survival case.
<code>meandiff.prior</code>	Mean value of prior distribution for mean difference. Relevant for two-sample continuous case.
<code>mean.prior</code>	Mean value of prior distribution for mean. Relevant for one-sample continuous case.
<code>sd.prior</code>	Standard deviation of prior distribution for mean difference (2-sample continuous case) or mean (1-sample continuous case) or prop (2-sample binary case) or difference of proportion (1-sample binary case) or log(HR) (2 sample survival case).
<code>propdiff.prior</code>	Mean value of prior distribution for difference in proportion. Relevant for two-sample binomial case.
<code>prop.prior</code>	Mean value of prior distribution for proportion. Relevant for one-sample binomial case.
<code>hr.prior</code>	Mean value of prior distribution for hazards ratio (HR). Relevant for two-sample survival case.
<code>D.prior</code>	Ignored if <code>sd.prior</code> is specified. If <code>sd.prior</code> is not specified then <code>sd.prior</code> is determined as $2/D.prior$. Relevant for two-sample survival case.

Details

This function can be used to determine Conditional power (CP) and Predictive power or predictive probability of success (PPoS) based on the interim results for each of continuous (one-sample or two-samples), binary (one-sample or two-samples) and time-to-event endpoints (two-samples). The PPoS can be based on interim results only or using both prior information and interim results. The calculation of CP and PPoS are carried out assuming normal distribution for estimated parameter and normal prior distribution. This function can be used to determine clinical success (`succ.crit="clinical"`) and trial success (`succ.crit="trial"`). For clinical success, `clin.succ.threshold` must be specified. For trial success, `Z.crit.final` or `alpha.final` must be specified.

In order to calculate CP and PPoS, `succ.ia()` should be invoked in the following form:

Continuous-two sample case (trial success):

```
succ.ia(type="cont", nsamples=2, null.value=, alternative=, N=, n=, a, meandiff.ia, stderr.ia=, succ.crit="t
```

Continuous-two sample case (clinical success):

```
succ.ia(type="cont", nsamples=2, null.value=, alternative=, N=, n=, a, meandiff.ia, stderr.ia=, succ.crit="c
```

Continuous-one sample case (trial success):

```
succ.ia(type="cont", nsamples=1, null.value=, alternative=, N=, n=, mean.ia, stderr.ia=, succ.crit="trial",
```

Continuous-one sample case (clinical success):

```
succ.ia(type="cont", nsamples=1, null.value=, alternative=, N=, n=, mean.ia, stderr.ia=, succ.crit="clini
```

Binary-two sample case (trial success):

```
succ.ia(type="bin", nsamples=2, null.value=, alternative=, N=, n=, a, proppdiff.ia, stderr.ia=, succ.crit="tr
```

Binary-two sample case (clinical success):

```
succ.ia(type="bin", nsamples=2, null.value=, alternative=, N=, n=, a, proppdiff.ia, stderr.ia=, succ.crit="cl
```

Binary-one sample case (trial success):

```
succ.ia(type="bin", nsamples=1, null.value=, alternative=, N=, n=, prop.ia, succ.crit="trial", Z.crit.fina
```

Binary-one sample case (clinical success):

```
succ.ia(type="bin", nsamples=1, null.value=, alternative=, N=, n=, prop.ia, succ.crit="clinical", clin.succ
```

Survival-two sample case (trial success):

```
succ.ia(type="surv", nsamples=2, null.value=, alternative=, D=, d=, a, hr.ia, succ.crit="trial", Z.crit.fina
```

Survival-two sample case (clinical success):

```
succ.ia(type="surv", nsamples=2, null.value=, alternative=, D=, d=, a, hr.ia, succ.crit="clinical", clin.suc
```

The conditional power is calculated assuming interim trend for post-interim data. If `meandiff.exp` (for continuous 2-samples case), `mean.exp` (for continuous 1-sample case), `proppdiff.exp` (for binomial 2-samples case), `prop.exp` (for binomial 1-sample case), or `hr.exp` (for survival 2-samples case) is specified, then conditional power would be calculated using these specified value as well.

The Predictive power or Predictive probability of success (PPoS) is calculated based interim results. On top of this, it can also incorporate prior information. The prior information can be specified as follows: If `meandiff.prior`, `sd.prior` for continuous 2-samples case, `mean.prior`, `sd.prior` for continuous 1-sample case, `proppdiff.prior`, `sd.prior` for binomial 2-samples case, `prop.prior`, `sd.prior` for binomial 1-sample case, and `hr.exp`, `sd.prior` (or, `hr.exp`, `D.prior`) for survival 2-samples case.

Author(s)

Madan Gopal Kundu <madan_g.kundu@yahoo.com>

See Also

[succ_ia_betabinom_one](#), [succ_ia_betabinom_two](#), [PoS](#)

Examples

```
### Lan et al. (2009), see #6. Example, outcome: Matching
succ_ia(type="cont", nsamples=1, null.value=0, alternative="greater",
        N=225, n=45,
        mean.ia=0, stderr.ia=1,
```



```

succ.crit="trial", Z.crit.final=1.96)

#--- Dallow et al. (2011), see Figure 1. Example, outcome: Matching
succ_ia(type="cont", nsamples=1, null.value=0, alternative="greater",
        N=100, n=50,
        mean.ia=1.364, stderr.ia=1,
        succ.crit="trial", Z.crit.final=1.64)

#--- Example 1 in the paper (Continuous endpoint)
succ_ia(type="cont", nsamples=2, null.value=-0.05, alternative="greater",
        N=1552, n=776, a=1,
        meandiff.ia=-0.025, sd.ia=0.16,
        succ.crit="trial", Z.crit.final=1.97,
        meandiff.exp=-0.030,
        meandiff.prior=0, sd.prior=0.02)

#--- Example 2 in the paper (Binary endpoint)
p1<- 0.379; p2<- 0.222
n1<- 105; n2<- 53

#-- Trial success
succ_ia(type="bin", nsamples=2, null.value=0, alternative="greater",
        N=210, n=158, a=2,
        proppdiff.ia=p1-p2,
        stderr.ia=sqrt(p1*(1-p1)/n1 + p2*(1-p2)/n2),
        succ.crit="trial", Z.crit.final=2.012,
        proppdiff.exp=0.20,
        proppdiff.prior=0.20, sd.prior=sqrt(0.06))

#-- Clinical success
succ_ia(type="bin", nsamples=2, null.value=0, alternative="greater",
        N=210, n=158, a=2,
        proppdiff.ia=p1-p2,
        stderr.ia=sqrt(p1*(1-p1)/n1 + p2*(1-p2)/n2),
        succ.crit="clinical", clin.succ.threshold=0.15,
        proppdiff.exp=0.20,
        proppdiff.prior=0.20, sd.prior=sqrt(0.06))

#--- Example 3 in the paper (Survival endpoint)

#--- Trial success
succ_ia(type="surv", nsamples=2, null.value=1, alternative="less",
        D=441, d=346, a=1,
        hr.ia=0.82,
        succ.crit="trial", Z.crit.final=2.012,
        hr.exp=0.75,
        hr.prior=0.71, D.prior=133)

#--- clinical success
succ_ia(type="surv", nsamples=2, null.value=1, alternative="less",
        D=441, d=346, a=1,

```

```
hr.ia=0.82,
succ.crit="clinical", clin.succ.threshold=0.80,
hr.exp=0.75,
hr.prior=0.71, D.prior=133)
```

succ_ia_betabinom_one *Determines predictive power of success based on interim results and beta prior for one-sample binary data*

Description

This function can be used to determine predictive power for trial success and clinical success based on the interim results and beta prior distribution for test of population proportion.

Usage

```
succ_ia_betabinom_one(N, n, x,
                      null.value = 0, alternative = "greater",
                      test="z", correct=TRUE,
                      succ.crit = "trial", Z.crit.final = 1.96,
                      alpha.final = 0.025, clin.succ.threshold = NULL,
                      a = 1, b = 1)
```

Arguments

N	Sample size at final analysis. Cannot be missing.
n	Sample size at interim analysis. Cannot be missing.
x	Number of observed response at interim analysis. Cannot be missing.
null.value	The specified value under null hypothesis. Default is 0.
alternative	Direction of alternate hypothesis. Can be "greater" or "less".
test	Statistical test. Default is "z" for Z test. For Exact binomial test, specify "exact".
correct	A logical indicating whether Yates' continuity correction should be applied where possible. Applies to approximate Z-test only.
succ.crit	Specify "trial" for trial success (i.e., null hypothesis is rejected at final analysis) or "clinical" for clinical success (i.e., estimated value at the final analysis is greater than clinically meaningful value as specified under clin.succ.threshold.)
Z.crit.final	The rejection boundary at final analysis in Z-value scale. Either alpha.final or Z.crit.final must be specified when determining trial success.
alpha.final	The rejection boundary at final analysis in alpha (1-sided) scale (e.g., 0.025). Either alpha.final or Z.crit.final must be specified when determining trial success.
clin.succ.threshold	Clinically meaningful value. Required when succ.crit="clinical".
a	Value of a corresponding to Beta(a, b) prior for proportion.
b	Value of b corresponding to Beta(a, b) prior for proportion.

Details

This function can be used to determine Predictive power or predictive probability of success (PPoS) based on the interim results for testing of population proportion. The calculation of PoS is carried out assuming beta prior distributions for proportion. This function can be used to determine clinical success (succ.crit="clinical") and trial success (succ.crit="trial"). For clinical success, clin.succ.threshold must be specified. For trial success, Z.crit.final or alpha.final must be specified.

Author(s)

Madan Gopal Kundu <madan_g.kundu@yahoo.com>

See Also

[succ_ia_betabinom_two](#), [succ_ia](#), [PoS](#)

Examples

```
succ_ia_betabinom_one( N=40, n=30, x=25,
  null.value=0.6, alternative="greater",
  succ.crit = "trial", Z.crit.final = 1.96,
  a = 1, b=1)
```

```
succ_ia_betabinom_one( N=40, n=30, x=25,
  null.value=0.6, alternative="greater", test="exact",
  succ.crit = "trial", Z.crit.final = 1.96,
  a = 1, b=1)
```

```
succ_ia_betabinom_one( N=40, n=30, x=15,
  null.value=0.6, alternative="greater",
  succ.crit = "clinical", clin.succ.threshold =0.5,
  a = 1, b=1)
```

succ_ia_betabinom_two *Determines predictive power of success based on interim results and beta priors for two-sample binary data*

Description

This function can be used to determine predictive power for trial success and clinical success based on the interim results and beta prior distribution for test of difference of two proportions.

Usage

```
succ_ia_betabinom_two(N.trt, N.con,
  n.trt, x.trt, n.con, x.con,
  alternative = "greater", test = "z",
  succ.crit = "trial", Z.crit.final = 1.96,
  alpha.final = 0.025, clin.succ.threshold = NULL,
  a.trt = 1, b.trt = 1, a.con = 1, b.con = 1)
```

Arguments

<code>N.trt</code>	Sample size in treatment arm at final analysis. Cannot be missing.
<code>N.con</code>	Sample size in control arm at final analysis. Cannot be missing.
<code>n.trt</code>	Sample size in treatment arm at interim analysis. Cannot be missing.
<code>x.trt</code>	Number of observed response in treatment arm at interim analysis. Cannot be missing.
<code>n.con</code>	Sample size in control arm at interim analysis. Cannot be missing.
<code>x.con</code>	Number of observed response in control arm at interim analysis. Cannot be missing.
<code>alternative</code>	Direction of alternate hypothesis. Can be "greater" or "less".
<code>test</code>	Statistical test. Default is "z" for Z test. For Fisher's exact test, specify "fisher".
<code>succ.crit</code>	Specify "trial" for trial success (i.e., null hypothesis is rejected at final analysis) or "clinical" for clinical success (i.e., estimated value at the final analysis is greater than clinically meaningful value as specified under <code>clin.succ.threshold</code> .)
<code>Z.crit.final</code>	The rejection boundary at final analysis in Z-value scale. Either <code>alpha.final</code> or <code>Z.crit.final</code> must be specified when determining trial success.
<code>alpha.final</code>	The rejection boundary at final analysis in alpha (1-sided) scale (e.g., 0.025). Either <code>alpha.final</code> or <code>Z.crit.final</code> must be specified when determining trial success.
<code>clin.succ.threshold</code>	Clinically meaningful value. Required when <code>succ.crit="clinical"</code> .
<code>a.trt</code>	Value of a corresponding to Beta(a, b) prior in treatment arm.
<code>b.trt</code>	Value of b corresponding to Beta(a, b) prior in treatment arm.
<code>a.con</code>	Value of a corresponding to Beta(a, b) prior for proportion in control arm.
<code>b.con</code>	Value of b corresponding to Beta(a, b) prior for proportion in control arm.

Details

This function can be used to determine Predictive power or predictive probability of success (PPoS) based on the interim results for comparison of two proportions. The calculation of PoS is carried out assuming beta prior distributions for proportions in both treatment and control arms. This function can be used to determine clinical success (`succ.crit="clinical"`) and trial success (`succ.crit="trial"`). For clinical success, `clin.succ.threshold` must be specified. For trial success, `Z.crit.final` or `alpha.final` must be specified.

Author(s)

Madan Gopal Kundu <madan_g.kundu@yahoo.com>

See Also

[succ_ia_betabinom_one](#), [succ_ia](#), [PoS](#)

Examples

```

succ_ia_betabinom_two( N.con=40, N.trt=40,
  n.trt=30, x.trt=20, n.con=30, x.con=15,
  alternative="greater", test="fisher",
  succ.crit = "trial", Z.crit.final = 1.96,
  a.trt = 1, b.trt=1, a.con=1, b.con=1)

succ_ia_betabinom_two( N.con=40, N.trt=40,
  n.trt=30, x.trt=20, n.con=30, x.con=15,
  alternative="greater", test="z",
  succ.crit = "trial", Z.crit.final = 1.96,
  a.trt = 1, b.trt=1, a.con=1, b.con=1)

succ_ia_betabinom_two( N.con=40, N.trt=40,
  n.trt=30, x.trt=20, n.con=30, x.con=15,
  alternative="greater", test="fisher",
  succ.crit = "clinical", clin.succ.threshold = 0.5,
  a.trt = 1, b.trt=1, a.con=1, b.con=1)

#--- Johns & Andersen, 1999, Example 1a (results matching)
succ_ia_betabinom_two( N.trt=32, N.con=32,
  n.trt=12, x.trt=8, n.con=12, x.con=8,
  alternative="greater", test="fisher",
  succ.crit = "clinical", clin.succ.threshold = 0,
  a.trt = 1, b.trt=1, a.con=1, b.con=1)

#--- Johns & Andersen, 1999, Example 1b (results matching)
succ_ia_betabinom_two( N.trt=32, N.con=32,
  n.trt=12, x.trt=8, n.con=12, x.con=11,
  alternative="greater", test="fisher",
  succ.crit = "clinical", clin.succ.threshold = 0,
  a.trt = 1, b.trt=1, a.con=1, b.con=1)

#--- Johns & Andersen, 1999, Example 2 (not matching, reported 0.586, got 0.536)
succ_ia_betabinom_two( N.trt=155+170, N.con=152+171,
  n.trt=155, x.trt=13, n.con=152, x.con=21,
  alternative="less", test="z",
  succ.crit = "trial", Z.crit.final = 1.96,
  a.trt = 1, b.trt=1, a.con=1, b.con=1)

succ_ia_betabinom_two( N.trt=155+170, N.con=152+171,
  n.trt=155, x.trt=13, n.con=152, x.con=21,
  alternative="less", test="fisher",
  succ.crit = "trial", Z.crit.final = 1.96,
  a.trt = 1, b.trt=1, a.con=1, b.con=1)

```

Description

Recursive partitioning for linear mixed effects model with survival data per SurvCART algorithm based on baseline partitioning variables (Kundu, 2020).

Usage

```
SurvCART(data, patid, timevar, censorvar, gvars, tgvars,
         time.dist="exponential", cens.dist="NA", event.ind=1,
         alpha=0.05, minsplit=40, minbucket=20, print=FALSE)
```

Arguments

<code>data</code>	name of the dataset. It must contain variable specified for <code>patid</code> (indicating subject id), all the variables specified in the formula and the baseline partitioning variables.
<code>patid</code>	name of the subject id variable.
<code>timevar</code>	name of the variable with follow-up times.
<code>censorvar</code>	name of the variable with censoring status.
<code>gvars</code>	list of partitioning variables of interest. Value of these variables should not change over time. Regarding categorical variables, only numerically coded categorical variables should be specified. For nominal categorical variables or factors, please first create corresponding dummy variable(s) and then pass through <code>gvars</code> .
<code>tgvars</code>	types (categorical or continuous) of partitioning variables specified in <code>gvar</code> . For each of continuous partitioning variables, specify 1 and for each of the categorical partitioning variables, specify 0. Length of <code>tgvars</code> should match to the length of <code>gvars</code>
<code>time.dist</code>	name of time-to-event distribution. It can be one of the following distributions: "exponential", "weibull", "lognormal" or "normal".
<code>cens.dist</code>	name of censoring distribution. It can be one of the following distributions: "exponential", "weibull", "lognormal", "normal" or "NA". If specified "NA", then parameter instability test corresponding to censoring distribution will not be performed.
<code>event.ind</code>	value of the censoring variable indicating event.
<code>alpha</code>	alpha (i.e., nominal type I error) level for parameter instability test
<code>minsplit</code>	the minimum number of observations that must exist in a node in order for a split to be attempted.
<code>minbucket</code>	the minimum number of observations in any terminal node.
<code>print</code>	if TRUE, then summary such as number of subjects at risk, number of events, median event time and median censoring time model will be printed for each node.

Details

Construct survival tree based on heterogeneity in time-to-event and censoring distributions.

Exponential distribution: $f(t)=\lambda \exp(-\lambda t)$

Weibull distribution: $f(t)=\alpha \lambda t^{(\alpha-1)} \exp(-\lambda t^\alpha)$

Lognormal distribution: $f(t)=(1/t) \cdot (1/\sqrt{2 \cdot \pi \cdot \sigma^2}) \cdot \exp[-(1/2) \cdot (\log(t)-\mu)/\sigma^2]$

Normal distribution: $f(t)=(1/\sqrt{2 \cdot \pi \cdot \sigma^2}) \cdot \exp[-(1/2) \cdot (t-\mu)/\sigma^2]$

Value

Treeout	contains summary information of tree fitting for each terminal nodes and non-terminal nodes. Columns of Treeout include "ID", the (unique) node numbers that follow a binary ordering indexed by node depth, n, the number of subjects reaching the node, D, the number of events reaching the node, median.T, the median survival time at the node, median.C, the median censoring time at the node, var, splitting variable, index, the cut-off value of splitting variable for binary partitioning, p (Instability), the p-value for parameter instability test for the splitting variable, loglik, the log-likelihood at the node, AIC, the AIC at the node, improve, the improvement in deviance given by this split, and Terminal, indicator (True or False) of terminal node.
logLik.tree	log-likelihood of the tree-structured model, based on Cox model including sub-groups as covariates
logLik.root	log-likelihood at the root node (i.e., without tree structure), based on Cox model without any covariate
AIC.tree	AIC of the tree-structured model, based on Cox model including sub-groups as covariates
AIC.root	AIC at the root node (i.e., without tree structure), based on Cox model without any covariate
subj.class	Assigned node for each individual subjects per fitted longitudinal tree
frame	rpart compatible object
splits	rpart compatible object
cptable	rpart compatible object
functions	rpart compatible object

Author(s)

Madan Gopal Kundu <madan_g.kundu@yahoo.com>

References

Kundu, M. G.(2020). Survival trees based on score based parameter instability test.

See Also

[plot.SurvCART](#), [KMPlot.SurvCART](#), [text.SurvCART](#), [StabCat.surv](#), [StabCont.surv](#)

Examples

```

#--- Get the data
data(GBSG2)

#numeric coding of character variables
GBSG2$horTh1<- as.numeric(GBSG2$horTh)
GBSG2$tgrade1<- as.numeric(GBSG2$tgrade)
GBSG2$menostat1<- as.numeric(GBSG2$menostat)

#Add subject id
GBSG2$subjid<- 1:nrow(GBSG2)

#--- Run SurvCART() with time-to-event distribution: exponential, censoring distribution: None
out<- SurvCART(data=GBSG2, patid="subjid", censorvar="cens", timevar="time",
  gvars=c('horTh1', 'age', 'menostat1', 'tsize', 'tgrade1', 'pnodes', 'progrec', 'estrec'),
  tgvars=c(0,1,0,1,0,1, 1,1),
  event.ind=1, alpha=0.05, minsplit=80, minbucket=40, print=TRUE)

#--- Plot tree
par(xpd = TRUE)
plot(out, compress = TRUE)
text(out, use.n = TRUE)

#Plot KM plot for sub-groups identified by tree
KMPlot.SurvCART(out, scale.time=365.25, type=1)

#--- Run SurvCART() with time-to-event distribution: weibull censoring distribution: None
out2<- SurvCART(data=GBSG2, patid="subjid", censorvar="cens", timevar="time",
  gvars=c('horTh1', 'age', 'menostat1', 'tsize', 'tgrade1', 'pnodes', 'progrec', 'estrec'),
  tgvars=c(0,1,0,1,0,1, 1,1),
  time.dist="weibull", event.ind=1, alpha=0.05, minsplit=80, minbucket=40, print=TRUE)

#--- Run SurvCART() with time-to-event distribution: weibull censoring distribution: exponential
out<- SurvCART(data=GBSG2, patid="subjid", censorvar="cens", timevar="time",
  gvars=c('horTh1', 'age', 'menostat1', 'tsize', 'tgrade1', 'pnodes', 'progrec', 'estrec'),
  tgvars=c(0,1,0,1,0,1, 1,1),
  time.dist="weibull", cens.dist="exponential", event.ind=1,
  alpha=0.05, minsplit=80, minbucket=40, print=TRUE)

```

text

Place text on SurvCART or LongCART tree

Description

Labels the current plot of the tree generated from SurvCART or LongCART object with text.

Usage

```
## S3 method for class 'SurvCART'
text(x, splits = TRUE, all = FALSE,
      use.n = FALSE, minlength = 1L, ...)
## S3 method for class 'LongCART'
text(x, splits = TRUE, all = FALSE,
      use.n = FALSE, minlength = 1L, ...)
```

Arguments

x	a fitted object of class "SurvCART", containing a survival tree, or class "LongCART", containing a longitudinal tree.
splits	similar to text.rpart; logical flag. If TRUE (default), then the splits in the tree are labeled with the criterion for the split.
all	similar to text.rpart; Logical. If TRUE, all nodes are labeled, otherwise just terminal nodes.
use.n	similar to text.rpart; Logical. If TRUE, adds n to label.
minlength	similar to text.rpart; the length to use for factor labels. A value of 1 causes them to be printed as 'a', 'b', Larger values use abbreviations of the label names. See the labels.rpart function for details.
...	arguments to be passed to or from other methods.

Author(s)

Madan Gopal Kundu <madan_g.kundu@yahoo.com>

References

Kundu, M. G.(2020). Survival trees based on score based parameter instability test.

See Also

[plot](#), [SurvCART](#), [LongCART](#)

Examples

```
### Get the data
data(GBSG2)

#numeric coding of character variables
GBSG2$horTh1<- as.numeric(GBSG2$horTh)
GBSG2$tgrade1<- as.numeric(GBSG2$tgrade)
GBSG2$menostat1<- as.numeric(GBSG2$menostat)

#Add subject id
GBSG2$subjid<- 1:nrow(GBSG2)

### Run SurvCART()
out<- SurvCART(data=GBSG2, patid="subjid", censorvar="cens", timevar="time", event.ind=1,
```

```
gvars=c('horTh1', 'age', 'menostat1', 'tsize', 'tgrade1', 'pnodes', 'progrec', 'estrec'),
  tgvars=c(0,1,0,1,0,1, 1,1),
  alpha=0.05, minsplit=80,
  minbucket=40, print=TRUE)

#--- Plot tree
par(xpd = TRUE)
plot(out, compress = TRUE)
text(out, use.n = TRUE)
```

Index

ACTG175, [2](#)

GBSG2, [3](#)

KMPlot.SurvCART, [4](#), [31](#)

LongCART, [5](#), [10](#), [14](#), [16](#), [19](#), [33](#)

plot, [9](#), [33](#)

plot.LongCART, [7](#), [14](#), [16](#), [19](#)

plot.SurvCART, [4](#), [18](#), [21](#), [31](#)

PoS, [10](#), [24](#), [27](#), [28](#)

ProfilePlot.LongCART, [7](#), [13](#)

StabCat, [7](#), [15](#)

StabCat (StabCat), [15](#)

StabCat.surv, [4](#), [16](#), [31](#)

StabCat.surv (StabCat.surv), [16](#)

StabCont, [7](#), [16](#), [18](#), [19](#)

StabCont (StabCont), [18](#)

StabCont.surv, [4](#), [18](#), [19](#), [21](#), [31](#)

StabCont.surv (StabCont.surv), [19](#)

succ_ia, [12](#), [21](#), [27](#), [28](#)

succ_ia_betabinom_one, [12](#), [24](#), [26](#), [28](#)

succ_ia_betabinom_two, [12](#), [24](#), [27](#), [27](#)

SurvCART, [4](#), [10](#), [18](#), [21](#), [29](#), [33](#)

text, [10](#), [32](#)

text.LongCART, [7](#), [14](#), [16](#), [19](#)

text.SurvCART, [4](#), [18](#), [21](#), [31](#)