

Package ‘anoint’

July 22, 2025

Type Package

Title Analysis of Interactions

Version 1.5

Date 2024-09-04

Maintainer Ravi Varadhan <ravi.varadhan@jhu.edu>

Description The tools in this package are intended to help researchers assess multiple treatment-covariate interactions with data from a parallel-group randomized controlled clinical trial. The methods implemented in the package were proposed in Kovalchik, Varadhan and Weiss (2013) <[doi:10.1002/sim.5881](https://doi.org/10.1002/sim.5881)>.

License GPL (>= 2)

LazyLoad yes

Depends R (>= 2.13.0), stats, survival, MASS, graphics, grid

Imports methods, glmnet

NeedsCompilation no

Repository CRAN

Author Ravi Varadhan [aut, cre],
Stephanie Kovalchik [aut],
Leon Wang [ctb]

Date/Publication 2024-09-09 10:00:09 UTC

Contents

anoint-package	2
anoint	3
anoint.fit	5
anoint.formula	7
anoint.subgroups	8
coef	9
confint	10
data.anoint	10
fits	12

forest	12
forest.subsets	13
formula.anoint-class	15
obo	16
pim	17
pim-class	19
pim.fit	20
pim.subsets	21
plot	23
predict	24
print	25
prognostic.score	25
show	25
show-anoint.fit	26
show.anoint	26
show.formula.anoint	26
simsolvd	27
summary	28
uim	28
vcov	29
Index	30

anoint-package	<i>Analysis of interactions for generalized linear models (GLM) or Cox proportional hazards regression models.</i>
----------------	--

Description

Testing procedures and models for investigating the heterogeneity of treatment effect in a clinical trial with multiple baseline covariates.

Details

Package: anoint
Type: Package
Version: 1.4
Date: 2015-7-10
License: GPL

Author(s)

Ravi Varadhan <rvaradhan@jhmi.edu> and Stephanie Kovalchik (maintainer) <s.a.kovalchik@gmail.com>

References

Follmann DA, Proschan MA. A multivariate test of interaction for use in clinical trials. *Biometrics* 1999; 55(4):1151-1155

anoimt	Create analysis of interactions object
--------	--

Description

Prepares formula and data to be used in methods with the analysis of interactions class.

Usage

```
anoimt(formula,data,family="binomial",select=NULL,nfolds=10,
       type.measure="deviance",keep.vars=NULL,na.action=na.omit,...)
```

Arguments

formula	analysis of interaction formula for glm or coxph, see details
data	data.frame containing the variables of formula
family	character specifying family of glm or character "coxph" if coxph model is fit
select	character for type of selection to perform, either "stepAIC" or "glmnet".
nfolds	number of folds used in cross-validation to find lasso penalty parameter when select is set to TRUE. Used only when select is glmnet. See cv.glmnet
type.measure	loss to use for cross-validation. Used only when select is glmnet. See cv.glmnet
keep.vars	vector of names of variables to retain if selection procedure is used. Used only when select is glmnet.
na.action	function, na.action to perform for handling observations with missing variables among variables in formula. Default is na.omit
...	additional arguments passed to cv.glmnet when select is glmnet or stepAIC when select is stepAIC.

Details

To test proportional multiple interactions between treatment variable indicator z (binary, 0 or 1) and variables a , b , with response y of a GLM model, formula must be $y \sim (a+b)*z$. If a Cox model with event time $time$ and event indicator $event$, formula is $Surv(time,event) \sim (a+b)*z$.

Factors should not be included as a or b because this could change how the reference group is represented in the model. Separate 0/1 dummy variables must be supplied by the user.

When select is glmnet a Lasso method ([cv.glmnet](#)) is used to select prognostic factors using 10-fold cross-validation with the control data only. If select is set to [stepAIC](#) a stepwise selection procedure is used with specifications based on arguments passed to ...

Value

Returns instance of `anoint` class.

Author(s)

Stephanie Kovalchik <s.a.kovalchik@gmail.com>

References

Follmann DA, Proschan MA. A multivariate test of interaction for use in clinical trials. *Biometrics* 1999; 55(4):1151-1155

See Also

[data.anoint, cv.glmnet](#)

Examples

```
# NO INTERACTION CONDITION, LOGISTIC MODEL

null.interaction <- data.anoint(
  alpha = c(log(.5), log(.5*.75)),
  beta = log(c(1.5, 2)),
  gamma = rep(1, 2),
  mean = c(0, 0),
  vcov = diag(2),
  type="survival", n = 500
)

head(null.interaction)

object <- anoint(Surv(y, event)~(V1+V2)*trt, data=null.interaction, family="coxph")

object

summary(object)

# NO INTERACTION CONDITION, WITH PROGNOSTIC SELECTION

null.interaction <- data.anoint(
  alpha = c(log(.2/.8), log(.2*.75/(1-.2*.75))),
  beta = c(1.5, 2, 0, 0),
  gamma = rep(1, 4),
  mean = rep(0, 4),
  vcov = diag(4),
  type="binomial", n = 500
)

head(null.interaction)

object <- anoint(y~(V1+V2+V3+V4)*trt, data=null.interaction, select="glmnet")
```

```
summary(object)

# FORCE V1, V2 INTO THE MODEL; INTERCEPT IS ALWAYS THE FIRST TERM OF MODEL
object <- anoimt(y~(V1+V2+V3+V4)*trt,data=null.interaction,
select="glmnet",keep.vars=c("V1","V2"))

summary(object)

# SELECTION WITH STEPWISE SELECTION AND AIC CRITERION
object <- anoimt(y~(V1+V2+V3+V4)*trt,data=null.interaction,
select="stepAIC")

summary(object)
```

anoimt.fit

*Fits and global tests of analysis of interaction models***Description**

Fits one-by-one (OBO), unrestricted (UIM), and proportional interaction (PIM) regression models to investigate multiple treatment response factors in a parallel-group clinical trial.

Arguments

object	object of anoimt class
level	significance level for global interaction tests
interval	interval of possible values for responsiveness parameter of PIM

Details

The global tests for the presence of treatment response factors (treatment-covariate interaction) are one-stage or two-stage likelihood ratio tests.

The fitted multiple interaction models include: one-by-one univariate interaction models (OBO), a full unrestricted model with all pairwise treatment-covariate interactions (UIM), and a proportional interactions model (PIM) fit with an exact or asymptotic approximate estimate for the likelihood ratio test and responsiveness parameter, theta.

Objects from the Class

Objects can be created by calls of the form `anoimt.fit(object, level = .05, interval=c(.5,3))`

Slots

K number of prognostic factors
 responsiveness list with exact and approximate estimates of PIM responsiveness parameter
 tests list of global interaction test results

`pvalues` list of pvalues on which test rejections are based

`fits` list of fitted models for each `anoimt` method

Components of tests are the results of the global tests of interaction:

`obo.reject` Result of unadjusted one-by-one global test of interaction. Null is no effect modification for K subgroups, the alternative is at least one K is an effect modifier.

`obo.adjust` Same as `obo.reject` but with Bonferroni-correction for K comparisons

`uim.reject` Result of UIM global test of interaction. Null is no effect modification for K subgroups, the alternative is at least one K is an effect modifier.

`pim.exact.reject` Result of PIM exact global test of interaction. Null is no *proportional* effect modification (theta responsiveness parameter = 1) against the alternative that the treatment responsiveness parameter theta is not equal to 1.

`pim.approx.reject` Same as `pim.exact.reject` but using approximate method.

`pim.obo` Two-stage global test. First stage tests PIM using an exact method at level/2 significance. If not rejected, the second stage is a test of adjusted OBO with a second-stage global level/2 significance.

`pim.uim` Same as `pim.obo` but with UIM at the second stage.

Components of `pvalues` on which the global tests are based:

`obo.p` p-value for the maximum LRT of the one-by-one testing

`uim.p` p-value for the global LRT of any interaction base on UIM

`pim.exact.p` p-value for the test of proportional interaction using the PIM exact method

`pim.approx.p` p-value for the test of proportional interaction using the PIM approximate method

Components of `fits` are the models underlying the global interaction tests:

`obo` Univariate interaction regression models of each subgroup.

`uim` Full regression model with all pairwise treatment-covariate interactionns

`pim.exact` Proportional interactions model with exact fit

`pim.approx` Proportional interactions model with asymptotic approximate estimation

Methods

show `signature(object = "anoimt.fit")`: Display table of results of global test of interaction.

print `signature(x = "anoimt.fit", ...)`: Display table of results of global test of interaction.

summary `signature(object = "anoimt.fit", ...)`: Display results of global test of interaction and p-values. Returns list with tests and pvalues.

fits `signature(object = "anoimt.fit", type=c("obo", "uim", "pim.exact", "pim.approx"))`: Extracts the specified fitted object from a `anoimt.fit`.

Author(s)

S. Kovalchik <s.a.kovalchik@gmail.com>

See Also

[anoint](#), [anoint-class](#), [obo](#), [uim](#), [pim](#)

Examples

```
# NO INTERACTION CONDITION, LOGISTIC MODEL

null.interaction <- data.anoint(
  alpha = c(log(.5), log(.5*.75)),
  beta = log(c(1.5, 2)),
  gamma = rep(1, 2),
  mean = c(0, 0),
  vcov = diag(2),
  type="survival", n = 500
)

object <- anoint(Surv(y, event)~(V1+V2)*trt, data=null.interaction, family="coxph")

fit <- anoint.fit(object)

summary(fit)

fits(fit, type="obo")
```

anoint.formula	Create a formula.anoint object
----------------	--------------------------------

Description

Create formula for use with anoint.

Usage

```
anoint.formula(formula=y~(a+b)*trt, family="binomial")
```

Arguments

formula	formula specifying analysis of interaction model
family	string indicating model family, should be one of glm families or coxph

Details

The formula should have all subgroups enclosed in parantheses and the treatment variable as an interaction, as shown in default.

Value

Returns instance of formula.anoint class.

Author(s)

Stephanie Kovalchik <s.a.kovalchik@gmail.com>

See Also

[formula.anoimt-class,anoimt](#)

Examples

```
anoimt.formula()
```

<code>anoimt.subgroups</code>	<i>Perform one-by-one subgroup analyses</i>
-------------------------------	---

Description

Computes all interaction effects one variable at a time.

Usage

```
anoimt.subgroups(formula,trt,data,family="binomial",na.action=na.omit,fwer=0.05,...)
```

Arguments

<code>formula</code>	formula for covariate model as given in <code>glm</code> or <code>coxph</code> , i.e. $y \sim x_1 + x_2$
<code>trt</code>	character name of treatment assignment indicator
<code>data</code>	<code>data.frame</code> containing the variables of <code>formula</code> and <code>trt</code>
<code>family</code>	character specifying family of <code>glm</code> or character "coxph" if <code>coxph</code> model is fit
<code>na.action</code>	function, na.action to perform for handling observations with missing variables among variables in <code>formula</code> . Default is <code>na.omit</code>
<code>fwer</code>	numeric value for the desired familywise error rate, should be between 0 and 1.
<code>...</code>	additional arguments passed to <code>glm</code> or <code>coxph</code>

Value

Returns a list with

subset indicator of the covariates included in the fitted model

interaction value of the of treatment-covariate interaction effect (using model with treatment-covariate product term)

LRT value of likelihood ratio test of treatment-covariate interaction

lower lower endpoints of 95 percent confidence interval for interaction parameter

upper upper endpoints of 95 percent confidence interval for interaction parameter

pvalue pvalue for 1-df chi-squared test

include.exclude.matrix matrix of same rows as covariates and columns as covariates with logical entries indicating which covariates (columns) were include in the fitted model (row)

covariates vector of covariate names as in formula

reject indicator of rejected hypotheses using a Bonferroni multiple testing correction such that familywise error is controlled at level fwer.

Author(s)

Stephanie Kovalchik <s.a.kovalchik@gmail.com>

Examples

```
set.seed(11903)

# NO INTERACTION CONDITION, LOGISTIC MODEL

null.interaction <- data.anoint(
  alpha = c(log(.5),log(.5*.75)),
  beta = log(c(1.5,2)),
  gamma = rep(1,2),
  mean = c(0,0),
  vcov = diag(2),
  type="survival", n = 500
)

head(null.interaction)

anoint.subgroups(Surv(y, event)~V1+V2,trt="trt",data=null.interaction,family="coxph")

# PROPORTIONAL INTERACTION WITH THREE COVARIATES AND BINARY OUTCOME

pim.interaction <- data.anoint(
  n = 5000,
  alpha = c(log(.2/.8),log(.2*.75/(1-.2*.75))),
  beta = rep(log(.8),3),
  gamma = rep(1.5,3),
  mean = c(0,0,0),
  vcov = diag(3),
  type="binomial"
)

anoint.subgroups(y~V1+V2+V3,trt="trt",data=pim.interaction,family="binomial")
```

coef

Get coefs from pim object.

Description

Extract vector of coefs of the fit of a pim model.

Methods

coef signature(object = "pim"): Extractor for MLEs returned as a matrix with one column.

Author(s)

S. Kovalchik <s.a.kovalchik@gmail.com>

confint

Compute confidence intervals of pim model terms.

Description

Computes Wald-based confidence intervals for the terms of a PIM model.

Methods

confint signature(object = "pim", parm, level = 0.95, ...): Returns estimate and confidence intervals for specified parm. Default is to return all params of the model.

Author(s)

S. Kovalchik <s.a.kovalchik@gmail.com>

data.anoint

Generate a clinical trial data set under a specified multiple interaction model

Description

Returns a simulated equal-allocation, parallel-group clinical trial with possibly multiple interactions among prognostic factors.

Usage

```
data.anoint(alpha, beta, gamma, mean, vcov, n=100, event=.8,
  type = c("binomial", "survival"))
```

Arguments

alpha	vector specifying control and treatment group intercept
beta	vector specifying prognostic factor main effects
gamma	vector specifying modification of prognostic effects in the presence of treatment
mean	vector of covariate mean to be supplied to mvnrm of the MASS package.
vcov	matrix of variance-covariance matrix of prognostic covariates to be supplied to mvnrm
n	number of subjects in each treatment arm
event	proportion of observed events when type is "survival"
type	string indicating type of response variable

Details

When type is "survival", the parameters specify the log-rate for an exponentially distributed random variable. Censored times are non-informatively right-censored. When type is "binomial", the parameters specify a log-odds model.

Value

Data frame with y, trt, and V1,...,VK prognostic factors. The data frame also has event with a time-to-event response, which is an indicator for an observed event.

Author(s)

S. Kovalchik <s.a.kovalchik@gmail.com>

Examples

```
null.interaction <- data.anoint(
  alpha = c(log(.2/.8),log(.2*.75/(1-.2*.75))),
  beta = c(1.5,2),
  gamma = rep(1,2),
  mean = c(0,0),
  vcov = diag(2),
  type="binomial"
)
```

```
fit <- glm(y~(.)*trt,data=null.interaction,family="binomial")
```

```
summary(fit)
```

```
null.interaction <- data.anoint(
  alpha = c(log(.5),log(.5*.75)),
  beta = c(1.5,2),
  gamma = rep(1,2),
  mean = c(0,0),
  vcov = diag(2),
```

```

        type="survival",
        event = .7
    )

fit <- coxph(Surv(y, event)~(.)*trt,data=null.interaction)

summary(fit)

```

fits	<i>Extract fits from anoint.fit object</i>
------	--

Description

Extract the specified anoint model from a `anoint.fit` object.

Methods

fits `signature(object = "anoint.fit", type=c("obo", "uim", "pim.exact", "pim.approx")):`
 Extracts the specified fitted object from a `anoint.fit`.

Author(s)

S. Kovalchik <s.a.kovalchik@gmail.com>

forest	<i>Construct a forestplot from a anoint object</i>
--------	--

Description

Constructs a forestplot displaying the treatment effect within subgroups for each set of specified categorical variables. Provides a visual comparison of subgroup treatment effect to overall treatment effect, without adjustment for confounding factors or multiplicity.

Includes unadjusted p-values for each subgroup's treatment-covariate interaction test for glm or Cox regression models.

Usage

```
forest(object, terms=NULL, x.axis=NULL, labels=NULL, fun=exp, ...)
```

Arguments

<code>object</code>	object of <code>anoint</code> class
<code>terms</code>	numeric index indicating which terms in prognostic model to include in plot
<code>x.axis</code>	Points for tick marks of the bottom axis
<code>labels</code>	Matrix of labels of the same rows as <code>onebyone</code> , columns printed from left to right
<code>fun</code>	functional transformation applied to treatment effects and confidence intervals
<code>...</code>	additional arguments passed to <code>foresplot</code> (see details)

Details

Additional arguments that can be specified include:

pch.size one number or a vector of two elements indicating the (min, max) for the relative plotting symbols. These reflect the relative precision of the symbol estimates

x.axis Points for tick marks of the bottom axis

header Vector of names for the columns of labels

main String of the title of plot

If no values are specified for the `x.axis`, eight values over the range of the CIs is used.

If no values are specified for `labels`, the row names are used.

Value

A labeled forestplot of subgroup treatment effects.

Author(s)

Stephanie Kovalchik <s.a.kovalchik@gmail.com>

Examples

```
data(simsolvd)

obj <- anoint(Surv(ttodthorchfhosp, censor==0)~(nyha+cardratio+current)*trt,data=simsolvd,
             family="coxph")

forest(obj)

grid.newpage()

forest(obj,x.axis=seq(0.7,1.5,by=.2))
```

forest.subsets

Subsets forest plot for proportional interactions models

Description

Plot of interaction effects for all possible proportional interactions models.

Usage

```
forest.subsets(object, index = 1:(min(length(object$interaction),
  30)), labels = NULL, exclude.fill = "white", include.fill = "grey30",
  signif.fill = "red", percent.inner = 0.9, xlimits = NULL,
  legend = TRUE, subgroup.text = NULL, subgroup.axis = NULL,
  subgroup.title = "Included Covariates",
  effects.text = NULL, effects.axis = NULL, confint = TRUE,
  segments.gpar = NULL, subgroup = FALSE)
```

Arguments

<code>object</code>	result of <code>pim.subsets</code>
<code>index</code>	vector indicating which subset models to include in plot, maximum of 30 of the best subsets if not specified.
<code>labels</code>	vector of names for subgroups. If <code>NULL</code> , covariates of <code>pim.subsets</code> is used.
<code>exclude.fill</code>	color for grid squares of excluded covariates
<code>include.fill</code>	color for grid squares of included covariates
<code>signif.fill</code>	color for plot circles indicating multiplicity-corrected significance
<code>percent.inner</code>	percentage of graphic device window for plot region
<code>xlimits</code>	vector of two elements indicating minimum and maximum value for effects plot. Values and confidence intervals outside <code>xlimits</code> will be clipped.
<code>legend</code>	logical value indicating whether legend for significant values should be included
<code>subgroup.text</code>	gpar list for modifying title of subgroup grid
<code>subgroup.axis</code>	gpar list for modifying text of subgroup grid labels
<code>subgroup.title</code>	character for title over inclusion/exclusion grid
<code>effects.text</code>	gpar list for modifying title of effects plot
<code>effects.axis</code>	gpar list for modifying text of effects plot axis
<code>confint</code>	logical indicating whether to include 95 percent confidence intervals on effects plot
<code>segments.gpar</code>	gpar list for rendering of confidence interval segments
<code>subgroup</code>	logical indicator of whether fitted object is the result of <code>anooint.subgroups</code>

Details

The significance level is the multiplicity corrected criterion with `fwer` control as specified by `pim.subsets`.

Value

Returns a plot of the results of all subsets of proportional interactions models. On the lefthand side we plot a grid describing the subsets models. This is a grid showing the included and exclude covariates of each proportional interactions model. Each row corresponds to a particular model. Colored squares in each row indicate the covariates given a proportional interaction effect, while unfilled (`exclude.fill`) indicate covariates left out of the model. The righthand side shows the interaction effect estimates (`effects`) for the corresponding subset model.

Author(s)

Stephanie Kovalchik <s.a.kovalchik@gmail.com>

See Also

[pim.subsets](#)

Examples

```
set.seed(11903)

# NO INTERACTION CONDITION, LOGISTIC MODEL
# SUPPOSE 5 HYPOTHEZED EFFECT MODIFIERS

null.interaction <- data.anoint(
  alpha = c(log(.5), log(.5*.75)),
  beta = log(rep(1.5, 5)),
  gamma = rep(1, 5),
  mean = rep(0, 5),
  vcov = diag(5),
  type="survival", n = 500
)

head(null.interaction)

fit <- pim.subsets(Surv(y, event)~V1+V2+V3+V4+V5, trt="trt",
  data=null.interaction, family="coxph")

forest.subsets(fit)
```

formula.anoint-class *Class "formula.anoint"*

Description

Formula class for anoint objects

Objects from the Class

Objects can be created by calls of the form `new("formula.anoint", ...)`.

Slots

formula: formula supplied to anoint
uni: list of all univariate interaction model formulas
prognostic: formula of subgroup main effects
prognostic.trt: formula of subgroup and treatment main effects model (no interactions)
trt: character string of treatment variable
family: character string describing model family

Methods

show signature(object = "formula.anoint"): Display main formula of formula.anoint object.

print signature(x = "formula.anoint",...): Display main formula of formula.anoint object.

update signature(object = "formula.anoint",...): Updates formula with same syntax as supplied to anoint function.

Author(s)

Stephanie Kovalchik <s.a.kovalchik@gmail.com>

See Also

[anoint](#)

obo	<i>Perform one-by-one (OBO) estimates of treatment-covariate interaction</i>
-----	--

Description

Fits GLM of Cox regression models with a treatment-covariate interaction separately for each covariate specified in anoint object.

Usage

```
obo(anoint,...)
```

Arguments

anoint	object of class anoint
...	other arguments passed to glm or coxph

Value

A list of model fits (fit), interaction likelihood-ratio statistics (LRT), and pvalues per interaction test (pvalue).

Author(s)

Stephanie Kovalchik <s.a.kovalchik@gmail.com>

See Also

[anoint](#)

Examples

```
data(simsolvd)

obj <- anoint(Surv(ttodthorchfhosp,censor==0)~(age+beat+lvef)*trt,
data = simsolvd, family = "coxph")

fits <- obo(obj)

fits

lapply(fits$fit, summary)
```

pim

*Fit proportional interaction model***Description**

Fits a proportional interactions model from parallel-group clinical trial.

Usage

```
pim(object,exact=TRUE,interval=c(-3,3),n.boot=NULL,...)
```

Arguments

object	object of class anoint
exact	logical indicator whether exact or approximate fit is performed
interval	interval passed to optimize if exact is TRUE
n.boot	number of bootstrap resamples for variance calculations
...	other arguments passed to optimize if exact is TRUE

Details

When exact is FALSE the method of Follmann and Proschan (1999) is used to estimate the PIM coefficients and perform a likelihood-ratio test on the responsiveness parameter theta.

If exact method is specified, then optimize is used to maximize the profile-likelihood alternating between fixing theta and fixing all other PIM parameters. The arguments interval and additional arguments to ... control the optimization with respect to theta.

When n.boot is NULL no bootstrap resamples are taken. In this case, when using the exact method the variance-covariance for the main effects is based on the model likelihood treating the responsiveness parameter as fixed. To include uncertainty measures for the responsiveness parameter, bootstrap resampling can be used. For the approximate method, only the bootstrap resampling variance is provided for the vcov and confint methods, which is invoked by specifying a positive integer number of samples or n.boot.

Value

Returns instance of `pim` class.

Author(s)

Stephanie Kovalchik <s.a.kovalchik@gmail.com>

References

Follmann DA, Proschan MA. A multivariate test of interaction for use in clinical trials. *Biometrics* 1999; 55(4):1151-1155

See Also

[pim-class](#)

Examples

```
set.seed(1115)

pim.interaction <- data.anoint(
  alpha = c(log(.2/.8), log(.2*.75/(1-.2*.75))),
  beta = log(c(1.25, 1.5)),
  gamma = rep(1.2, 2),
  mean = c(0, 0),
  vcov = diag(2),
  type="binomial", n = 500
)

object <- anoint(y~(V1+V2)*trt, data=pim.interaction, family="binomial")

object

fit <- pim(object)

fit

summary(fit)

# EXAMPLE WITH BOOTSTRAP
fit <- pim(object, n=50)
summary(fit)
confint(fit)
```

pim-class

Class "pim"

Description

Class for proportional interactions model (PIM).

Objects from the Class

Objects can be created by calls of the form `new("pim", ...)`.

Slots

formula: object of anoint class

coef: list of baseline (alpha), control prognostic effects (beta), and responsiveness parameter (theta)

exact: logical indicator whether exact fit used

LRT: likelihood ratio test statistic of responsiveness parameter

interval: numeric vector supplied to optimize if exact fit is used

boot.pim: matrix of bootstrap estimates of PIM coefficients

vcov: matrix of bootstrap variance-covariance

Methods

show signature(object = "pim"): Display point estimates of pim object.

print signature(x = "pim", ...): Display point estimates of pim object.

summary signature(object = "pim", ...): List of estimates and convergence information.

coef signature(object = "pim"): Extractor for MLEs.

confint signature(object = "pim", parm, level=0.95, ...): Computes confidence interval for specified parameters and level. Intervals are based on bootstrap resamples of PIM model.

predict signature(object = "pim"): Returns linear predictor for each subject.

vcov signature(object = "pim"): Extractor for variance-covariance matrix which is estimated with bootstrap resampling.

Author(s)

Stephanie Kovalchik <s.a.kovalchik@gmail.com>

See Also

[pim](#)

pim.fit *Fit proportional interactions model*

Description

Fits a single proportional interactions model for generalized linear or Cox regression model.

Usage

```
pim.fit(formula, trt, data, family="binomial", na.action=na.omit, ...)
```

Arguments

formula	formula for covariate model as given in glm or coxph, i.e. $y \sim x_1 + x_2$
trt	character name of treatment assignment indicator
data	data.frame containing the variables of formula and trt
family	character specifying family of glm or character "coxph" if coxph model is fit
na.action	function, na.action to perform for handling observations with missing variables among variables in formula. Default is na.omit
...	additional arguments passed to glm or coxph

Details

Under the proportional interaction model the coef of the main covariate effects in the control arm are multiplied by the interaction effect to get the covariate effects for the treatment group.

Value

Returns a list with

interaction value of the interaction effect of the proportional interaction model, see details

LRT value of likelihood ratio test of proportional interaction

lower lower endpoint of 95 percent confidence interval for interaction parameter

upper upper endpoint of 95 percent confidence interval for interaction parameter

pvalue pvalue for 1-df chi-squared test

model0 model object for control group

model1 model object for treatment group

Author(s)

Stephanie Kovalchik <s.a.kovalchik@gmail.com>

References

Follmann DA, Proschan MA. A multivariate test of interaction for use in clinical trials. *Biometrics* 1999; 55(4):1151-1155

See Also[pim.subsets](#)**Examples**

```

set.seed(11903)

# NO INTERACTION CONDITION, LOGISTIC MODEL

null.interaction <- data.anoint(
  alpha = c(log(.5), log(.5*.75)),
  beta = log(c(1.5, 2)),
  gamma = rep(1, 2),
  mean = c(0, 0),
  vcov = diag(2),
  type="survival", n = 500
)

head(null.interaction)

pim.fit(Surv(y, event)~V1+V2, trt="trt", data=null.interaction, family="coxph")

# PROPORTIONAL INTERACTION WITH THREE COVARIATES AND BINARY OUTCOME

pim.interaction <- data.anoint(
  n = 5000,
  alpha = c(log(.2/.8), log(.2*.75/(1-.2*.75))),
  beta = rep(log(.8), 3),
  gamma = rep(1.5, 3),
  mean = c(0, 0, 0),
  vcov = diag(3),
  type="binomial"
)

pim.fit(y~V1+V2+V3, trt="trt", data=pim.interaction, family="binomial")

```

pim.subsets

*Perform all subsets proportional interactions modeling***Description**

Computes all possible proportional interactions model among p covariates.

Usage

```
pim.subsets(formula, trt, data, family="binomial", na.action=na.omit, fwer=0.05, ...)
```

Arguments

<code>formula</code>	formula for covariate model as given in <code>glm</code> or <code>coxph</code> , i.e. $y \sim x_1 + x_2$
<code>trt</code>	character name of treatment assignment indicator
<code>data</code>	<code>data.frame</code> containing the variables of <code>formula</code> and <code>trt</code>
<code>family</code>	character specifying family of <code>glm</code> or character "coxph" if coxph model is fit
<code>na.action</code>	function, na.action to perform for handling observations with missing variables among variables in <code>formula</code> . Default is <code>na.omit</code>
<code>fw</code>	numeric value for the desired familywise error rate, should be between 0 and 1.
<code>...</code>	additional arguments passed to <code>glm</code> or <code>coxph</code>

Details

Under the proportional interaction model the coef of the main covariate effects in the control arm are multiplied by the interaction effect to get the covariate effects for the treatment group.

Value

Returns a list with

subset indicator of the covariates included in the fitted model

interaction value of the interaction effect of the proportional interaction model, see details

LRT value of likelihood ratio test of proportional interaction

lower lower endpoints of 95 percent confidence interval for interaction parameter

upper upper endpoints of 95 percent confidence interval for interaction parameter

pvalue pvalue for 1-df chi-squared test

include.exclude.matrix matrix of same rows as subsets and columns as covariates with logical entries indicating which covariates (columns) were include in which subset model (row)

covariates vector of covariate names as in `formula`

reject indicator of rejected hypotheses using a multiple testing correction such that familywise error is controlled at level `fw`.

Author(s)

Stephanie Kovalchik <s.a.kovalchik@gmail.com>

References

Follmann DA, Proschan MA. A multivariate test of interaction for use in clinical trials. *Biometrics* 1999; 55(4):1151-1155

Examples

```
set.seed(11903)

# NO INTERACTION CONDITION, LOGISTIC MODEL

null.interaction <- data.anoint(
  alpha = c(log(.5), log(.5*.75)),
  beta = log(c(1.5, 2)),
  gamma = rep(1, 2),
  mean = c(0, 0),
  vcov = diag(2),
  type="survival", n = 500
)

head(null.interaction)

pim.subsets(Surv(y, event)~V1+V2, trt="trt", data=null.interaction, family="coxph")

# PROPORTIONAL INTERACTION WITH THREE COVARIATES AND BINARY OUTCOME

pim.interaction <- data.anoint(
  n = 5000,
  alpha = c(log(.2/.8), log(.2*.75/(1-.2*.75))),
  beta = rep(log(.8), 3),
  gamma = rep(1.5, 3),
  mean = c(0, 0, 0),
  vcov = diag(3),
  type="binomial"
)

pim.subsets(y~V1+V2+V3, trt="trt", data=pim.interaction, family="binomial")
```

plot

Prognostic response plot (PR-plot) for anoint class.

Description

Computes the prognostic score (baseline risk) based on the covariates of anoint or a supplied set of predictions. Risk scores are binned into ten groups based on risk deciles and a treatment effect (and confidence interval) is estimated in each group. This is compared to the overall treatment effect which is indicated by the shaded region.

Details

Additional arguments are passed to glm or coxph.

Methods

plot signature(object = "anoint", predict=NULL, fun=exp, ...): Prognostic response plot.

Author(s)

S. Kovalchik <s.a.kovalchik@gmail.com>

Examples

```
set.seed(11903)

# BINOMIAL EVENT DATA WITH 4 NORMAL PROGNOSTIC FACTORS
pim.interaction <- data.anoint(
  alpha = c(log(.2/.8),log(.2*.75/(1-.2*.75))),
  beta = log(c(1.5,1.1,2,1.3)),
  gamma = rep(1.5,4),
  mean = rep(0,4),
  vcov = diag(4),
  type="binomial", n = 500
)

object <- anoint(y~(V1+V2+V3+V4)*trt,data=pim.interaction)

plot(object,bty="n",las=1)

# PLOT TREATMENT EFFECT ON LINEAR PREDICTOR SCALE
plot(object,fun=function(x)x,bty="n",las=1,ylab="treatment effect (linear predictor)")
```

predict

Get risk predictions for pim object.

Description

Computes linear predictors for the dataset used to fit the model.

Methods

predict signature(object = "pim"): Linear predictors for dataset used to fit pim model.

Author(s)

S. Kovalchik <s.a.kovalchik@gmail.com>

print	<i>Print coefficients of pim</i>
-------	----------------------------------

Description

Prints the regression coefficients of the fit of a pim.

Methods

print signature(x = "pim", digits=4, ...): Prints call, coefficients, and responsiveness parameter for pim object.

Author(s)

S. Kovalchik <s.a.kovalchik@gmail.com>

prognostic.score	<i>Prognostic scores for pim object.</i>
------------------	--

Description

Computes prognostic score from pim model, which is the linear predictor for the main effects of the prognostic factors for untreated subjects.

Methods

prognostic.score signature(object = "pim"): Prognostic score at baseline.

Author(s)

S. Kovalchik <s.a.kovalchik@gmail.com>

show	<i>Show coefficients of pim</i>
------	---------------------------------

Description

Shows the regression coefficients of the fit of a pim.

Methods

show signature(object = "pim"): Shows call, coefficients, and multiple interaction parameter for pim object.

Author(s)

S. Kovalchik <s.a.kovalchik@gmail.com>

show-anoaint.fit	Show table of LRT global test results for anoaint.fit object
------------------	--

Description

Prints a matrix with logical indicators of whether the given anoaint test rejected the null hypothesis for the global test of interaction.

Methods

show signature(object = "anoaint.fit"): Matrix of logical indicators for global LRT test rejection.

Author(s)

S. Kovalchik <s.a.kovalchik@gmail.com>

show.anoaint	Show anoaint object
--------------	---------------------

Description

Shows the formula specification of the anoaint object.

Methods

show signature(object = "anoaint"): Shows formula call for anoaint object.

Author(s)

S. Kovalchik <s.a.kovalchik@gmail.com>

show.formula.anoaint	Show formula.anoaint object
----------------------	-----------------------------

Description

Returns the formula of the formula.anoaint object.

Methods

show signature(object = "formula.anoaint"): Shows formula of the formula.anoaint object.

Author(s)

S. Kovalchik <s.a.kovalchik@gmail.com>

simsolvd*Simulated SOLVD-Trial data set*

Description

A simulated clinical trial based on the design of the Studies of Left Ventricular Dysfunction Trial (SOLVD-T), a placebo-controlled trial of the angiotensin-converting-enzyme inhibitor enalapril for patients with congestive heart failure.

Usage

```
data(simsolvd)
```

Format

A data frame with 2569 observations on the following 12 variables.

trt indicator for enalapril group
age age at baseline (centered and scaled)
beat pulse at baseline (centered and scaled)
lymphocyte lymphocyte count at baseline (centered and scaled)
lvef left ventricular ejection fraction at baseline (centered and scaled)
noise simulated vector of random uniform variables
nyha indicator whether New York Heart Association score greater than 2
cardratio indicator whether cardiothoracic ratio is greater than 0.5
creatinine creatinine at baseline (centered and scaled)
sodium sodium at baseline (centered and scaled)
ttodthorchfhosp time to death or hospitalization in days
censor indicator whether censored (1) or an event (0)
current indicator whether current smoker

Source

Simulated data set based on the clinical study reported by: Yusuf, S. et al. (1991). Effect of Enalapril on Survival in Patients with Reduced Left-Ventricular Ejection Fractions and Congestive-Heart-Failure. *NEJM* 325:293-302.

Examples

```
data(simsolvd)

obj <- anoint(Surv(ttodthorchfhosp,censor==0)~(age+beat+lvef)*trt,
data = simsolvd, family = "coxph")

uim(obj)$fit
```

summary	<i>Summary of anoint model fit.</i>
---------	-------------------------------------

Description

A list of estimates and convergence status of a anoint model fit. When exact is TRUE the test statistic is the likelihood ratio test, otherwise the Follmann approximate likelihood ratio test statistic is used.

Methods

summary signature(x = "anoint", ...): Calls summary method of underlying model fit.

Author(s)

S. Kovalchik <s.a.kovalchik@gmail.com>

uim	<i>Perform unrestricted multiple treatment-covariate interaction regression</i>
-----	---

Description

Fits GLM of Cox regression analysis for model with all pairwise interactions between treatment and covariates specified in anoint object.

Usage

```
uim(object, ...)
```

Arguments

object	object of class anoint
...	other arguments passed to glm or coxph

Value

List of model fit of glm or coxph class (fit), likelihood-ratio test of global test of treatment-covariate interaction (LRT), and LRT p-value for global test of interaction (pvalue).

Author(s)

Stephanie Kovalchik <s.a.kovalchik@gmail.com>

See Also

[glm](#), [coxph](#), [anoint](#)

Examples

```
data(simsolvd)

obj <- anoint(Surv(ttodthorchfhosp,censor==0)~(age+beat+lvef)*trt,
data = simsolvd, family = "coxph")

fit <- uim(obj)

fit

summary(fit$fit)
```

vcov*Get variance-covariance from pim object.*

Description

Returns Hessian-based variance-covariance matrix of the fit of a pim model.

Methods

vcov signature(object = "pim"): Extractor for variance-covariance of MLEs.

Author(s)

S. Kovalchik <s.a.kovalchik@gmail.com>

Index

- * **classes**
 - formula.anooint-class, 15
 - pim-class, 19
- * **datasets**
 - simsolvd, 27
- * **methods**
 - coef, 9
 - confint, 10
 - fits, 12
 - plot, 23
 - predict, 24
 - print, 25
 - prognostic.score, 25
 - show, 25
 - show-anooint.fit, 26
 - show.anooint, 26
 - show.formula.anooint, 26
 - summary, 28
 - vcov, 29
- * **package**
 - anooint-package, 2

anooint, 3, 7, 8, 16, 28
anooint-class, 7
anooint-class(anooint), 3
anooint-package, 2
anooint.fit, 5
anooint.fit-class(anooint.fit), 5
anooint.formula, 7
anooint.subgroups, 8

coef, 9
coef,pim-method(coef), 9
confint, 10
confint,pim-method(confint), 10
coxph, 28
cv.glmnet, 3, 4

data.anooint, 4, 10

fits, 12

fits,anooint.fit-method(fits), 12
forest, 12
forest.subsets, 13
formula.anooint-class, 8, 15

glm, 28

na.action, 3, 8, 20, 22

obo, 7, 16

pim, 7, 17, 19
pim-class, 18, 19
pim.fit, 20
pim.subsets, 15, 21, 21
plot, 23
plot,anooint-method(plot), 23
predict, 24
predict,pim-method(predict), 24
print, 25
print,pim-method(print), 25
prognostic.score, 25
prognostic.score,pim-method
(prognostic.score), 25

show, 25
show,anooint-method(show.anooint), 26
show,anooint.fit-method
(show-anooint.fit), 26
show,formula.anooint-method
(show.formula.anooint), 26
show,pim-method(show), 25
show-anooint.fit, 26
show.anooint, 26
show.formula.anooint, 26
simsolvd, 27
stepAIC, 3
summary, 28
summary,anooint-method(summary), 28

uim, 7, 28

`vcov`, [29](#)

`vcov,pim-method (vcov)`, [29](#)