Package ‘anoint’

July 19, 2015

Type Package

Title Analysis of Interactions

Version 1.4

Date 2015-07-10

Author Ravi Varadhan <r.varadhan@jhmi.edu> and Stephanie Kovalchik

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

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.

License GPL (>= 2)

LazyLoad yes

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

Imports methods, glmnet

NeedsCompilation no

Repository CRAN

Date/Publication 2015-07-19 19:44:25

R topics documented:

anoint-package ................................................................. 2
anoint ................................................................. 3
anoint.fit ................................................................. 5
anoint.formula ............................................................. 7
anoint.subgroups ........................................................... 8
coef ................................................................. 10
confint ................................................................. 10
data.anoint ................................................................. 10
fits ................................................................. 12
forest ................................................................. 12
forest.subsets ............................................................. 14
formula.anoint ........................................................... 16
obo ................................................................. 17
anoint-package

Analysis of interactions for generalized linear models (GLM) or Cox proportional hazards regression models.

Description

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

Details

<table>
<thead>
<tr>
<th>Package:</th>
<th>anoint</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type:</td>
<td>Package</td>
</tr>
<tr>
<td>Version:</td>
<td>1.4</td>
</tr>
<tr>
<td>Date:</td>
<td>2015-7-10</td>
</tr>
<tr>
<td>License:</td>
<td>GPL</td>
</tr>
</tbody>
</table>

Author(s)

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

References

anoint | Create analysis of interactions object

Description

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

Usage

anoint(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~(a+b)*z. If a Cox model with event time time and event indicator event, formula is Surv(time, event)~(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

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
anoint.fit

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 anoint 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 `anoint.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 anoint 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 = "anoint.fit"): Display table of results of global test of interaction.
print  signature(x = "anoint.fit",...): Display table of results of global test of interaction.
summary  signature(object = "anoint.fit",...): Display results of global test of interaction and p-values. Returns list with tests and pvalues.
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>
Create a formula for use with anoint.

Usage

```r
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 parentheses 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.anoint-class, anoint

Examples

```
anoint.formula()
```

---

**anoint.subgroups**  
*Perform one-by-one subgroup analyses*

---

Description
Computes all interaction effects one variable at a time.

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

Arguments
- `formula`  
  formula for covariate model as given in `glm` or `coxph`, i.e. `y~x1+x2`
- `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`
- `fwer`  
  numeric value for the desired familywise error rate, should be between 0 and 1.
- `...`  
  additional arguments passed to `glm` or `coxph`

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
anoint.subgroups

pvalue p-value 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")


coeff

Get coefs from pim object.

Description

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

Methods

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

Author(s)

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

confint

Compute confindence 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 `mvrnorm` of the MASS package.
- **vcov**: matrix of variance-covariance matrix of prognostic covariates to be supplied to `mvrnorm`
- **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

```r
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)
```

```r
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

*Extract fits from anoint.fit object*

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

*Construct a forestplot from a anoint object*

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

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

Details

Additional arguments that can be specified include:

- **pch.size**: one number of 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(ttodthorchf hosp, 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

```r
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

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

```r
set.seed(11903)

# NO INTERACTION CONDITION, LOGISTIC MODEL
# SUPPOSE 5 HYPOTHESIZED 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`
Perform one-by-one (OBO) estimates of treatment-covariate interaction

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(simsolv)
obj <- anoint(Surv(ttdothorchf hosp, censor==0)-(age+beat+lvef)*trt,
data = simsolvd, family = "coxph")
fits <- obo(obj)
fits
lapply(fits$fit, summary)
Fit proportional interaction model

Description

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

Usage

\[
\text{pim}(\text{object}, \text{exact}=\text{TRUE}, \text{interval}=c(-3, 3), \text{n.boot}=\text{NULL}, \ldots)
\]

Arguments

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

Details

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

When \text{n.boot} is \text{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 \text{vcov} and \text{confint} methods, which is invoked by specifying a positive integer number of samples or \text{n.boot}.

Value

Returns instance of \text{pim} class.

Author(s)

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

References

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(8,0),  
  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 codeanoint 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~x1+x2
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


See Also

- pim.subsets

Examples

```r
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

```r
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

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

Arguments

- **formula**: formula for covariate model as given in glm or coxph, i.e. y~x1+x2
- **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`
- **fwer**: numeric value for the desired familywise error rate, should be between 0 and 1.
- **...**: 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

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 included 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 fwer.

Author(s)

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

References


Examples

```r
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(0,8),
  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(
```
\begin{verbatim}
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")
\end{verbatim}

**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**

\begin{verbatim}
plot signature(object = "anoint",predict=NULL,fun=exp,...): Prognostic response plot.
\end{verbatim}

**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)
```
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
Print coefficients of pim

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  Prognostic scores for pim object.

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  Show coefficients of pim

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-anoint.fit  

Show table of LRT global test results for anoint.fit object

Description

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

Methods

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

Author(s)

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

---

show.anoint  

Show anoint object

Description

Shows the formula specification of the anoint object.

Methods

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

Author(s)

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

---

show.formula.anoint  

Show formula.anoint object

Description

Returns the formula of the formula.anoint object.

Methods

`show signature(object = "formula.anoint")`: Shows formula of the formula.anoint 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)
- ttodthorfhosp  time to death or hospitalization in days
- censor  indicator whether censored (1) or an event (0)
- current  indicator whether current smoker

Source

Examples
data(simsolvd)

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

uim(obj)$fit
**summary**

**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**

**Perform unrestricted multiple treatment-covariate interaction regression**

**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

```r
data(simsolvd)

obj <- anoint(Surv(ttodthorcfhosp,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

*Topic **classes**
  - formula.anoint-class, 16
  - pim-class, 19
*Topic **datasets**
  - simsolvd, 28
*Topic **methods**
  - coef, 10
  - confint, 10
  - fits, 12
  - plot, 24
  - predict, 25
  - print, 25
  - prognostic.score, 26
  - show, 26
  - show.anoint.fit, 27
  - show.anoint, 27
  - show.formula.anoint, 27
  - summary, 29
  - vcov, 30
*Topic **package**
  - anoint-package, 2
  - anoint, 3, 7, 8, 16, 17, 29
  - anoint-class, 7
  - anoint-class(anoint), 3
  - anoint-package, 2
  - anoint.fit, 5
  - anoint.fit-class(anoint.fit), 5
  - anoint.formula, 7
  - anoint.subgroups, 8
  - coef, 10
  - coef.pim-method(coef), 10
  - confint, 10
  - confint.pim-method(confint), 10
  - coxph, 29
  - cv.glmnet, 3, 4
  - data.anoint, 4, 10
  - fits, 12
  - fits.anoint.fit-method(fits), 12
  - forest, 12
  - forest.subsets, 14
  - formula.anoint-class, 8, 16
  - glm, 29
  - na.action, 3, 8, 20, 22
  - obo, 7, 17
  - pim, 7, 18, 20
  - pim-class, 19, 19
  - pim.fit, 20
  - pim.subsets, 15, 21, 22
  - plot, 24
  - plot.anoint-method(plot), 24
  - predict, 25
  - predict.pim-method(predict), 25
  - print, 25
  - print.pim-method(print), 25
  - prognostic.score, 26
  - prognostic.score.pim-method(prognostic.score), 26
  - show, 26
  - show.anoint-method(show.anoint), 27
  - show.anoint.fit-method(show.anoint.fit), 27
  - show.formula.anoint-method(show.formula.anoint), 27
  - show.pim-method(show), 26
  - show.anoint.fit, 27
  - show.anoint, 27
  - show.formula.anoint, 27
  - simsolvd, 28
  - stepAIC, 3
  - summary, 29
  - summary.anoint-method(summary), 29
  - uim, 7, 29
vcov, 30
vcov, pim-method (vcov), 30