Package 'pseval'

April 8, 2025

Type Package

Title Methods for Evaluating Principal Surrogates of Treatment Response

Version 1.3.3

Date 2025-04-08

Maintainer Michael C Sachs <sachsmc@gmail.com>

Description Contains the core methods for the evaluation of principal surrogates in a single clinical trial. Provides a flexible interface for defining models for the risk given treatment and the surrogate, the models for integration over the missing counterfactual surrogate responses, and the estimation methods. Estimated maximum likelihood and pseudo-score can be used for estimation, and the bootstrap for inference. A variety of post-estimation summary methods are provided, including print, summary, plot, and testing.

License MIT + file LICENSE

URL https://sachsmc.github.io/pseval/

BugReports https://github.com/sachsmc/pseval/issues/

Suggests ggplot2, testthat, knitr, printr, rmarkdown

Imports survival

VignetteBuilder knitr

RoxygenNote 7.3.2

Encoding UTF-8

NeedsCompilation no

Author Michael C Sachs [aut, cre], Erin E Gabriel [aut]

Repository CRAN

Date/Publication 2025-04-08 10:20:02 UTC

Contents

+.ps *Modify a psdesign object by adding on new components.*

Description

This operator allows you to add objects to a psdesign object, such as integration models and risk models

Usage

```
## S3 method for class 'ps'
p1 + p2
```

Arguments

p1	An object of class psdesign
p2	Another object to be added to p1, see list below for possible options If the first object is an object of class psdesign, you can add the following types of objects, and it will return a modified psdesign object. Users will generally add them in the order that they appear. • integration: Add or replace integration model • riskmodel: Add or replace risk model • estimate: Estimate parameters • bootstrap: Bootstrap estimates

add_bootstrap *Bootstrap resampling parameters*

Description

Bootstrap resampling parameters

Usage

```
add_bootstrap(psdesign, bootstrap)
```

Arguments

psdesign	A psdesign object, it must have risk model and integration model components
bootstrap	A bootstrap object created by ps_bootstrap

Examples

```
## Not run:
test <- psdesign(generate_example_data(n = 100), Z = Z, Y = Y.obs, S = S.obs, BIP = BIP)
est1 <- test + integrate_parametric(S.1 ~ BIP) + risk_binary() + ps_estimate(method = "BFGS")
est1 + ps_bootstrap(method = "BFGS", start = est1$estimates$par, n.boots = 50)

## End(Not run)
```

---

add_estimate *Estimate parameters*

Description

Estimate parameters

Usage

```
add_estimate(psdesign, estimate)
```

Arguments

psdesign	A psdesign object, it must have risk model and integration model components
estimate	An estimate object created by ps_estimate

Examples

```
test <- psdesign(generate_example_data(n = 100), Z = Z, Y = Y.obs, S = S.obs, BIP = BIP)
test + integrate_parametric(S.1 ~ BIP) + risk_binary(D = 50) + ps_estimate(method = "BFGS")
```

---

add_integration *Integration models*

Description

Add integration model to a psdesign object

Usage

```
add_integration(psdesign, integration)
```

Arguments

psdesign	A psdesign object
integration	An integration object

Details

This is a list of the available integration models. The fundamental problem in surrogate evaluation is that there are unobserved values of the counterfactual surrogate responses S(1). In the estimated maximum likelihood framework, for subjects missing the S(1) values, we use an auxiliary pre-treatment variable or set of variables W that is observed for every subject to estimate the distribution of S(1) | W. Typically, this W is a BIP. Then for each missing S(1), we integrate likelihood contributions over each non-missing S(1) given their value of W, and average over the contributions.

- integrate_parametric This is a parametric integration model that fits a linear model for the mean of S(1) | W and assumes a Gaussian distribution.
- integrate_bivnorm This is another parametric integration model that assumes that S(1) and W are jointly normally distributed. The user must specify their mean, variances and correlation.
- integrate_nonparametric This is a non-parametric integration model that is only valid for categorical S(1) and W. It uses the observed proportions to estimate the joint distribution of S(1), W.
- integrate_semiparametric This is a semi-parametric model that uses the semi-parametric location scale model of Heagerty and Pepe (1999). Models are specified for the location of S(1) | W and the scale of S(1) | W. Then integrations are drawn from the empirical distribution of the residuals from that model, which are then transformed to the appropriate location and scale.

Examples

```
test <- psdesign(generate_example_data(n = 100), Z = Z, Y = Y.obs, S = S.obs, BIP = BIP)
add_integration(test, integrate_parametric(S.1 ~ BIP))
test + integrate_parametric(S.1 ~ BIP)  # same as above
```

---

add_riskmodel *Add risk model to a psdesign object*

Description

Add risk model to a psdesign object

Usage

```
add_riskmodel(psdesign, riskmodel)
```

Arguments

psdesign	A psdesign object
riskmodel	A risk model object, from the list above

Details

The risk model component specifies the likelihood for the data. This involves specifying the distribution of the outcome variable, whether it is binary or time-to-event, and specifying how the surrogate S(1) and the treatment Z interact and affect the outcome. We use the formula notation to be consistent with other regression type models in R. Below is a list of available risk models.

- risk_binary This is a generic risk model for binary outcomes. The user can specify the formula, and link function using either risk.logit for the logistic link, or risk.probit for the probit link. Custom link functions may also be specified, which take a single numeric vector argument, and returns a vector of corresponding probabilities.
- risk_weibull This is a parameterization of the Weibull model for time-to-event outcomes that is consistent with that of rweibull. The user specifies the formula for the linear predictor of the scale parameter.
- risk_exponential This is a simple exponential model for a time-to-event outcome.
- risk_poisson This is a Poisson model for count outcomes. It allows for offsets in the formula.
- risk_continuous This is a Gaussian model for continuous outcomes. It assumes that larger values of the outcome are harmful (e.g. blood pressure)

Examples

```
test <- psdesign(generate_example_data(n = 100), Z = Z, Y = Y.obs, S = S.obs, BIP = BIP) +
     integrate_parametric(S.1 ~ BIP)
add_riskmodel(test, risk_binary())
test + risk_binary() # same as above
```

`calc_risk` *Calculate the risk and functions of the risk*

Description

Computes the treatment efficacy (TE) and other functions of the risk in each treatment arm over the range of surrogate values observed in the data. TE(s) is defined as 1 - risk(s, z = 1)/risk(s, z = 0), where z is the treatment indicator. If any other variables are present in the risk model, then the risk is computed at their median value.

Usage

```
calc_risk(
  psdesign,
  contrast = "TE",
  t,
  sig.level = 0.05,
  CI.type = "band",
  n.samps = 5000,
  bootstraps = TRUE,
  newdata = NULL
)
```

Arguments

psdesign	A psdesign object. It must contain a risk model, an integration model, and estimated parameters. Bootstrapped parameters are optional
contrast	The contrast function, or the name of the contrast function. See details.
t	For time to event outcomes, a fixed time t may be provided to compute the cumulative distribution function. If not, the restricted mean survival time is used. Omit for binary outcomes.
sig.level	Significance level for bootstrap confidence intervals
CI.type	Character string, "pointwise" for pointwise confidence intervals, and "band" for simultaneous confidence band.
n.samps	The number of samples to take over the range of S.1 at which the contrast is calculated
bootstraps	If true, and bootstrapped estimates are present, will calculate bootstrap standard errors and confidence bands.
newdata	Vector of S values. If present, will calculate the contrast function at values of newdata instead of the observed S.1

Details

The contrast function is a function that takes 2 inputs, the risk_0 and risk_1, and returns some one dimensional function of those two inputs. It must be vectorized. Some built-in functions are `"TE"` for treatment efficacy = 1 - risk_1(s)/risk_0(s), `"RR"` for relative risk = risk_1(s)/risk_0(s), `"logRR"` for log of the relative risk, and `"RD"` for the risk difference = risk_1(s) - risk_0(s).

Value

A data frame containing columns for the S values, the computed contrast function at S, R0, and R1 at those S values, and optionally standard errors and confidence intervals computed using bootstrapped estimates.

Examples

```
## Not run:
# same result passing function name or function
calc_risk(binary.boot, contrast = "TE", n.samps = 20)
calc_risk(binary.boot, contrast = function(R0, R1) 1 - R1/R0, n.samps = 20)

## End(Not run)
```

`calc_STG` *Calculate the Standardized total gain*

Description

Computes the standardized total gain for the risk difference. Optionally produces bootstrap standard errors and confidence intervals. The standardized total gain is the area between the risk difference curve and the horizontal line at the marginal risk difference. If the outcome is time to event then the STG is time-dependent, and a time point for evaluation is needed. If one is not provided then the restricted mean survival is estimated from the data and used.

Usage

```
calc_STG(
  psdesign,
  t,
  sig.level = 0.05,
  n.samps = 5000,
  bootstraps = TRUE,
  permute = TRUE,
  permute.times = 2000,
  progress.bar = TRUE
)
```

Arguments

`psdesign`	A psdesign object. It must contain a risk model, an integration model, and estimated parameters. Bootstrapped parameters are optional
`t`	For time to event outcomes, a fixed time `t` may be provided to compute the cumulative distribution function. If not, the restricted mean survival time is used. Omit for binary outcomes.
`sig.level`	Significance level for bootstrap confidence intervals
`n.samps`	The number of samples to take over the range of S.1 at which the VE is calculated
`bootstraps`	If true, and bootstrapped estimates are present, will calculate bootstrap standard errors and confidence interval.
`permute`	Not used, included for backwards compatibility
`permute.times`	Not used, included for backwards compatibility
`progress.bar`	Not used, included for backwards compatibility

`empirical_TE` *Compute the empirical Treatment Efficacy*

Description

Compute the empirical Treatment Efficacy

Usage

```
empirical_TE(psdesign, t)
```

Arguments

`psdesign`	An object of class psdesign
`t`	Fixed time for time to event outcomes to compute TE. If missing, uses restricted mean survival.

`empirical_VE` *Compute the empirical Treatment Efficacy*

Description

Included for backwards compatibility

Usage

```
empirical_VE(psdesign, t)
```

Arguments

`psdesign`	An object of class psdesign
`t`	Fixed time for time to event outcomes to compute TE. If missing, uses restricted mean survival.

expand_augdata *Expand augmented data using the integration function*

Description

Expand augmented data using the integration function

Usage

```
expand_augdata(model, psdesign, D = 500)
```

Arguments

model	Formula defining the risk model
psdesign	An object of class psdesign, that contains at least 1 integration model
D	The number of samples to take for the simulated annealing

generate_example_data *Generate sample data used for testing*

Description

Generate sample data used for testing

Usage

```
generate_example_data(n)
```

Arguments

n	Integer, the sample size

integrate_bivnorm *Bivariate normal integration models for the missing S(1)*

Description

This model assumes that the pair [S(1), W] is bivariate normal, where W is the BIP. The means, standard deviations, and correlation are estimated or fixed before calling this function. Then the conditional normal formula is applied in order to get the distribution of S(1) | W. That distribution is used to integrate over the missing S(1) values. This method requires a BIP in the design.

Usage

```
integrate_bivnorm(x = "S.1", mu = c(0, 0), sd = c(1, 1), rho = 0.2)
```

Arguments

x	expression identifying the variable to be integrated. Typically this is S.1 or S.0
mu	means of the pair of surrogates, missing one first
sd	standard deviations of the pair, missing one first
rho	the correlation between X1 and X2

integrate_nonparametric *Nonparametric integration model for the missing S(1)*

Description

Both S(1) and the BIP or set of BIPs must be categorical. This model integrates over the estimated distribution of S(1) | BIP

Usage

```
integrate_nonparametric(formula, ...)
```

Arguments

formula	Formula specifying the integration model for the surrogate under treatment. Generally the candidate surrogate will be on the left side in the formula, and the BIP or BIPs will be on the right side. In this case the BIP and the S(1) must be categorical.
...	Not currently used

integrate_parametric *Parametric integration model for the missing S(1)*

Description

Parametric integration model for the missing S(1)

Usage

```
integrate_parametric(formula, family = gaussian, ...)
```

Arguments

formula	Formula specifying the integration model for the surrogate under treatment. Generally the candidate surrogate will be on the left side in the formula, and the BIP or BIPs will be on the right side
family	Assumed distribution for the integration model. Must be compatible with the family argument of glm. Currently only Gaussian models are supported
...	Arguments passed to glm

integrate_semiparametric *Semiparametric integration model using the location-scale model*

Description

Semiparametric integration model using the location-scale model

Usage

```
integrate_semiparametric(formula.location, formula.scale, ...)
```

Arguments

formula.location	Formula specifying the integration model for the location component of the surrogate under treatment. Generally the candidate surrogate will be on the left side in the formula, and the BIP or BIPs will be on the right side
formula.scale	Formula specifying the integration model for the scale component of the surrogate under treatment. Generally the candidate surrogate will be on the left side in the formula, and the BIP or BIPs will be on the right side
...	Other parameters passed to sp_locscale

plot.psdesign *Plot summary statistics for a psdesign object*

Description

Plot the treatment efficacy or another contrast of risk versus S.1 for an estimated psdesign object

Usage

```
## S3 method for class 'psdesign'
plot(
  x,
  t,
  contrast = "TE",
  sig.level = 0.05,
  CI.type = "band",
  n.samps = 500,
  xlab = "S.1",
  ylab = contrast,
  col = 1,
  lty = 1,
  lwd = 1,
  ...
)
```

Arguments

x	A psdesign object that contains a risk model, integration model, and valid estimates
t	For time to event outcomes, a fixed time t may be provided to compute the cumulative distribution function. If not, the restricted mean survival time is used. Omit for binary outcomes.
contrast	Name of contrast function to plot. "TE" or "VE" for treatment (vaccine) efficacy = 1 - risk_1(s)/risk_0(s), "RR" for relative risk = risk_1(s)/risk_0(s), "logRR" for log of the relative risk, "risk" for the risk in each treatment arm, and "RD" for the risk difference = risk_1(s) - risk_0(s). You can also pass a custom function directly as long as it takes 2 vectors as input (risk0 and risk1) and returns 1 vector of the same length.
sig.level	Significance level used for confidence bands on the contrast curve. This is only used if bootstrapped estimates are available.
CI.type	Character string, "pointwise" for pointwise confidence intervals, and "band" for simultaneous confidence band.
n.samps	Number of samples to use over the range of S.1 for plotting the curve
xlab	X-axis label
ylab	Y-axis label

col	Vector of integers specifying colors for each curve.
lty	Vector of integers specifying linetypes for each curve.
lwd	Vector of numeric values for line widths.
...	Other arguments passed to plot

`print.psdesign` *Concisely print information about a psdesign object*

Description

Concisely print information about a psdesign object

Usage

```
## S3 method for class 'psdesign'
print(x, digits = 3, sig.level = 0.05, ...)
```

Arguments

x	An object of class psdesign
digits	Number of significant digits to display
sig.level	Significance level to use for computing bootstrapped confidence intervals
...	Currently unused

`psdesign` *Specify a design for a principal surrogate evaluation*

Description

Generate mappings that describe how variables in the data are mapped to components of the principal surrogate analysis. Other than `data`, this is a list of key-value pairs describing the common elements of a ps analysis. The required keys are Z, Y, and S. Optional keys are BIP, CPV, BSM, and weights. These elements are described in details below. Additional keys-value pairs can be included in `....` This function generates an augmented dataset and additional information for subsequent steps in the analysis. In the subsequent steps, refer to the variables by the keys. See add_integration and add_riskmodel for information on how to proceed in the analysis.

Usage

```
psdesign(
  data,
  Z,
  Y,
  S,
  BIP = NULL,
  CPV = NULL,
  BSM = NULL,
  weights = NULL,
  tau,
  ...
)
```

Arguments

data	Data frame containing data to be analyzed
Z	Expression defining the treatment variable which has 2 levels
Y	Expression defining the outcome variable. For binary events this should be coded as 0/1 or a factor with 2 levels. For censored time-to-event outcomes this can be a call to Surv
S	Expression defining the candidate surrogate
BIP	Optional expression defining the baseline irrelevant predictor
CPV	Optional expression defining the closeout placebo vaccination measurement
BSM	Optional expression defining the baseline surrogate measurement
weights	optional expression defining weights to accommodate nonrandom subsampling, such as case control or two phase
tau	numeric, When the outcome Y is a survival time, it is possible that the surrogate was measured at some time tau after enrollment. Use the argument tau to specify the time when the surrogate was measured, in study time. Not required for binary Y.
...	Other key-value pairs that will be included in the augmented data, e.g. additional candidate surrogates, covariates for adjustment, variables used for integration

---

`pseudo_score` *Estimate parameters from a specified model using pseudo-score*

Description

Estimate parameters from a specified model using pseudo-score

Usage

```
pseudo_score(psdesign, start = NULL, epsilon = 1e-05, maxit = 50)
```

Arguments

psdesign	An object of class psdesign
start	Vector of starting values, if NULL, will come up with starting values
epsilon	Convergence criteria
maxit	Maximum number of iterations

`ps_bootstrap` *Estimate parameters from a specified model using bootstrap resampling and estimated maximum likelihood*

Description

Estimate parameters from a specified model using bootstrap resampling and estimated maximum likelihood

Usage

```
ps_bootstrap(
  n.boots = 200,
  progress.bar = TRUE,
  start = NULL,
  method = "BFGS",
  control = list(),
  ...
)
```

Arguments

n.boots	Number of bootstrap replicates
progress.bar	Logical, if true will display a progress bar in the console
start	Vector of starting values, if NULL, will come up with starting values
method	Method to use for optimization, can be "pseudo-score" for categorical S with nonparametric integration, or any of the methods available in optim. Defaults to "BFGS"
control	List of control parameters for passed to optim
...	Arguments passed to optim

ps_estimate — *Estimate parameters from a specified model using estimated maximum likelihood*

Description

Estimate parameters from a specified model using estimated maximum likelihood

Usage

```
ps_estimate(start = NULL, method = "BFGS", control = list(), ...)
```

Arguments

start	Vector of starting values, if NULL, will come up with starting values
method	Method to use for optimization, can be "pseudo-score" for categorical BIP, or any of the methods available in optim. Defaults to "BFGS"
control	List of control parameters for passed to optim
...	Arguments passed to optim or pseudo_score.

risk.logit — *Logit link function*

Description

Logit link function

Usage

```
risk.logit(x)
```

Arguments

x	A vector of linear predictors

Value

A vector of probabilities

risk.probit *Probit link function*

Description

Probit link function

Usage

```
risk.probit(x)
```

Arguments

x	A vector of linear predictors

Value

A vector of probabilities

riskcalc *Calculate risks with handlers for survival data*

Description

Calculate risks with handlers for survival data

Usage

```
riskcalc(risk.function, Y, par, t, dat0, dat1)
```

Arguments

risk.function	Function taking three arguments, a data.frame, parameters, and time. It should return a vector the same number of rows as the data frame
Y	The outcome variable
par	the vector of parameter values
t	Time for a survival outcome, may be missing
dat0	Data frame containing S and Z = 1
dat1	Data frame containing S and Z = 0

risk_binary *Risk model for binary outcome*

Description

Risk model for binary outcome

Usage

```
risk_binary(model = Y ~ S.1 * Z, D = 5000, risk = risk.logit)
```

Arguments

model	Formula specifying the risk model
D	number of samples for the simulated annealing integration
risk	Function for transforming a linear predictor into a probability. E.g., risk.logit for the logistic model, risk.probit for the probit model

risk_continuous *Risk model for continuous outcome*

Description

This model assumes that the outcome Y is normally distributed conditional on S.1 and Z, with mean determined by the model formula. It also assumes that larger values of Y are more indicative of poor outcomes, e.g., blood pressure.

Usage

```
risk_continuous(model = Y ~ S.1 * Z, D = 5000)
```

Arguments

model	Formula specifying the risk model for the mean
D	number of samples for the simulated annealing integration

risk_exponential *Exponential risk model for time to event outcome*

Description

Exponential risk model for time to event outcome

Usage

```
risk_exponential(model = Y ~ S.1 * Z, D = 5000)
```

Arguments

model	Formula specifying the risk model. The outcome should be a Surv object specifying right censoring
D	number of samples for simulated annealing

risk_poisson *Poisson risk model for count outcomes*

Description

Poisson risk model for count outcomes

Usage

```
risk_poisson(model = Y ~ S.1 * Z, D = 5000)
```

Arguments

model	Formula specifying the risk model. The outcome should be an integer of counts. This right side of the formula may contain an offset term.
D	number of samples for simulated annealing

risk_weibull *Weibull risk model for time to event outcome*

Description

Weibull risk model for time to event outcome

Usage

```
risk_weibull(model = Y ~ S.1 * Z, D = 5000)
```

Arguments

model	Formula specifying the risk model. The outcome should be a Surv object specifying right censoring
D	number of samples for simulated annealing

sp_locscale *Fit the semi-parametric location-scale model*

Description

This estimates the location-scale model as described in Heagerty and Pepe (1999) using the Newton-Raphson method. The location and scale formulas must have the same outcome, but they may have different predictors.

Usage

```
sp_locscale(
  formula.location,
  formula.scale,
  data,
  weights,
  tol = 1e-06,
  maxit = 100
)
```

Arguments

formula.location	Formula specifying the model for the location
formula.scale	Formula specifying the model for the scale
data	Data used to estimate the model
weights	Weights applied to the estimating equations
tol	Convergence tolerance
maxit	Maximum number of iterations

Value

A list containing the parameter estimates, the convergence indicator, and residuals

---

`stg` *Compute the standardized total gain*

Description

Compute the standardized total gain

Usage

```
stg(R1, R0, stand = TRUE)
```

Arguments

`R1`	Risk in the treatment group
`R0`	Risk in the control group
`stand`	Standardize?

`summarize_bs` *Summarize bootstrap samples*

Description

Summarize bootstrap samples

Usage

```
summarize_bs(bootdf, estdf = NULL, sig.level = 0.05, CI.type = "band")
```

Arguments

`bootdf`	Data frame containing bootstrapped estimates, with a column containing a convergence indicator
`estdf`	Data frame containing full sample estimate
`sig.level`	Significance level to use for confidence intervals
`CI.type`	Character string, "pointwise" for pointwise confidence intervals, and "band" for simultaneous confidence band.

summary.psdesign *Summary method for psdesign objects*

Description

Summary method for psdesign objects

Usage

```
## S3 method for class 'psdesign'
summary(object, digits = 3, sig.level = 0.05, ...)
```

Arguments

object	An object of class psdesign
digits	Number of significant digits to display
sig.level	Significance level to use for computing bootstrapped confidence intervals
...	Currently unused

Value

Invisibly returns the printed table, along with the three estimates of vaccine efficacy. The empirical TE is 1 minus the relative risk comparing the treatment arm to the control arm. The risk is estimated as the proportion in the binary outcome case, or with the Kaplan-Meier estimate at the restricted mean survival in the time-to-event case. The marginal TE estimate is the TE estimate under the specified parametric risk model, ignoring the effect of S.1. The model based average TE is the TE estimate from the specified risk model, averaged over the distribution of S.1. The point of displaying these three is to assess the validity of the parametric model, and to assess the validity of the model estimation. Wild differences among these estimates may indicate problems with the model or convergence.

TE *Treatment efficacy contrast functions*

Description

Treatment efficacy contrast functions

Usage

```
TE(R0, R1)
```

Arguments

R0	A vector of risks in the control arm
R1	A vector of risks in the treatment arm

Details

These functions take the risk in the two treatment arms, and computes a one-dimensional summary of those risks. Built-in choices are `"TE"` for treatment efficacy = 1 - risk_1(s)/risk_0(s), `"RR"` for relative risk = risk_0(s)/risk_1(s), `"logRR"` for log of the relative risk, and `"RD"` for the risk difference = risk_0(s) - risk_1(s).

Value

A vector the same length as R0 and R1.

---

verify_trt *Check that a variable is suitable for using as binary treatment indicator*

Description

Checks for two classes and gives a warning message indicating which level is assumed to be 0/1

Usage

```
verify_trt(D)
```

Arguments

`D`	Vector that will be checked for 2-class labels

---

wem_test *Test for wide effect modification*

Description

This runs a multivariate Wald test on the interaction terms of the model, using the bootstrap covariance

Usage

```
wem_test(x)
```

Arguments

`x`	An object of class psdesign with bootstrap replicates

Index