Package: bonsaiforest (via r-universe)

October 24, 2026

Type Package

Title Shrinkage Based Forest Plots

Version 0.1.1

Date 2024-09-25

Description Subgroup analyses are routinely performed in clinical trial analyses. From a methodological perspective, two key issues of subgroup analyses are multiplicity (even if only predefined subgroups are investigated) and the low sample sizes of subgroups which lead to highly variable estimates, see e.g. Yusuf et al (1991) <doi:10.1001/jama.1991.03470010097038>. This package implements subgroup estimates based on Bayesian shrinkage priors, see Carvalho et al (2019) <https://proceedings.mlr.press/v5/carvalho09a.html>. In addition, estimates based on penalized likelihood inference are available, based on Simon et al (2011) <doi:10.18637/jss.v039.i05>. The corresponding shrinkage based forest plots address the aforementioned issues and can complement standard forest plots in practical clinical trial analyses.

License Apache License 2.0

URL https://github.com/insightsengineering/bonsaiforest/

BugReports https://github.com/insightsengineering/bonsaiforest/issues

Depends R (>= 4.1)

Imports brms (>= 2.22.0), broom, checkmate, dplyr, forcats, gbm, ggplot2, glmnet, MASS, Rcpp, splines2, stats, survival, tibble, tidyr, tidyselect, vdiffr

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

LinkingTo Rcpp, RcppArmadillo

VignetteBuilder knitr

Config/testthat/edition 3

Encoding UTF-8

Language en-US

LazyData true

LazyDataCompression xz

RoxygenNote 7.3.2.9000

NeedsCompilation yes

Author Mar Vazquez Rabunal [aut], Daniel Sabanés Bové [aut], Marcel Wolbers [aut], Isaac Gravestock [cre], F. Hoffmann-La Roche AG [cph, fnd]

Maintainer Isaac Gravestock <isaac.gravestock@roche.com>

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

Date/Publication 2024-09-27 12:30:02 UTC

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

RemoteRef HEAD

RemoteSha ca4e0b4513ecf000be42d860edd7f1d94df77459

Contents

ahr_estimation . . . 3
ahr_from_km . . . 4
compare . . . 5
design_dummy1 . . . 5
design_matrix1 . . . 6
elastic_net . . . 6
elastic_net_fit_bin . . . 7
elastic_net_fit_surv . . . 8
elastic_net_surv . . . 8
est_coef_bin1 . . . 9
example_data . . . 10
generate_stacked_data . . . 10
horseshoe . . . 11
horseshoe_fit_bin . . . 12
horseshoe_fit_surv . . . 13
lor_estimation . . . 13
naive . . . 14
naive_fit_bin . . . 15
naive_fit_surv . . . 15
naivepop . . . 16
naivepop_fit_bin . . . 16
naivepop_fit_surv . . . 17
plot.compare.data . . . 17
plot.summary.elastic_net . . . 18
plot.summary.horseshoe . . . 19
plot.summary.naive . . . 19
preprocess . . . 20

print.summary.elastic_net 21
print.summary.horseshoe 21
print.summary.naive 22
print.summary.naivepop 22
simul_covariates 23
simul_data 24
simul_pfs 25
subgroups 26
summary.elastic_net 27
summary.horseshoe 28
summary.naive 29
summary.naivepop 29
survival_curves 30
trt_horseshoe 31

Index **32**

`ahr_estimation` *Average Hazard Ratio Estimation*

Description

Function to obtain the average hazard ratio.

Usage

```
ahr_estimation(x_subg, dummy_subg, est_coef, h0, gamma = 1)
```

Arguments

`x_subg`	(`matrix`) the matrix with the subgroup covariates.
`dummy_subg`	(`matrix`) the dummy matrix with the subgroup covariates.
`est_coef`	(`matrix`) the estimated coefficients from the fitted model.
`h0`	(`numeric`) the vector with the cumulative baseline hazard.
`gamma`	(`scalar`) numeric value defining the weights to obtain the average hazard ratio. Default is 1 (in this case the average hazard ratio obtained can be interpreted as the odds of concordance).

Value

Average hazard ratio

Examples

```
ahr_estimation(
  elastic_net_surv$design1, elastic_net_surv$dummy1,
  elastic_net_surv$est_coef, elastic_net_surv$h0
)
```

`ahr_from_km` *Average Hazard Estimation based on Kaplan-Meier Estimates*

Description

Average Hazard Estimation based on Kaplan-Meier Estimates

Usage

```
ahr_from_km(resp, trt, data, status, t_quantile = 0.99)
```

Arguments

`resp`	(`string`) the response variable name.
`trt`	(`string`) the treatment variable name. The treatment variable must be a factor with 2 levels where the first level is the control and the second one the treatment.
`data`	(`data frame`) the data frame with the variables.
`status`	(`string`) the status variable name in survival data.
`t_quantile`	(`number`) definition of the quantile.

Details

Estimates can be unstable due to the variability of the Kaplan-Meier estimates in the tails. The `t_quantile` argument can address this, e.g. by setting it to 0.95, the 5% highest times will be discarded. By default, the 1% highest times are discarded.

Value

The estimated average hazard ratio of the treatment (second level of `trt`) versus the control (first level of `trt`).

Examples

```
ahr_from_km("tt_pfs", "arm", example_data, "ev_pfs")
ahr_from_km("tt_pfs", "arm", example_data, "ev_pfs", t_quantile = 0.95)
```

compare *Compare Treatment Estimate Methods*

Description

Function to create a data set with the results for all the methods in order to compare them.

Usage

```
compare(...)
```

Arguments

`...`	Objects of class `bonsaiforest`. We write them separated by a comma.

Value

List with a `data.frame` of the estimated subgroup treatment effects, with the overall treatment effect and with the response type of our data.

Examples

```
compare(naivepop_fit_surv, naive_fit_surv, elastic_net_fit_surv)
```

design_dummy1 *Design Dummy Subgroup x_1a*

Description

The dummy design matrix from `example_data` considering just observations from subgroup x_1a.

Usage

```
design_dummy1
```

Format

A `matrix` with 0 and 1 indicating the subgroups of each of the observations.

Source

This is an artificial example.

design_matrix1 *Design Matrix Subgroup x_1a*

Description

The design matrix from `example_data` considering just observations from subgroup x_1a.

Usage

```
design_matrix1
```

Format

A `matrix` with 0 and 1 indicating the subgroups of each of the observations and their interactions with the treatment variable.

Source

This is an artificial example.

elastic_net *Elastic Net Penalization Model Estimation*

Description

Function to fit the elastic net penalization model to the data. This model penalizes the interaction between the covariates and the treatment but leaves unpenalized the main effects.

Usage

```
elastic_net(
  resp,
  trt,
  subgr,
  covars,
  data,
  resptype = c("survival", "binary"),
  alpha,
  status = NULL
)
```

Arguments

resp	(string) the response variable name.
trt	(string) the treatment variable name. The treatment variable must be a factor with 2 levels where the first level is the control and the second one the treatment.
subgr	(character) vector with the name of the subgroup variables from which we want to obtain the subgroup treatment effect. They have to be factor variables with the subgroups as levels.
covars	(character) vector with the name of the variables that we want to include in the model. They have to be factor variables with the subgroups as levels. The subgr variables have to be included here.
data	(data frame) the data frame with the variables.
resptype	(string) the type of data used. Can be "survival" or "binary".
alpha	(scalar) the elastic net mixing parameter with values between 0 and 1. The special case of alpha=1 corresponds to a lasso penalty and the case of alpha=0 to a ridge penalty.
status	(string) only for "survival" resptype, the status variable name in survival data.

Value

List with fit, model, resptype, data, alpha, design_matrix, design_dummy, y, subgr_names.

Examples

```
elastic_net(
  "tt_pfs", "arm", c("x_1", "x_2"), c("x_1", "x_2", "x_3"),
  example_data, "survival", 1, "ev_pfs"
)
```

---

elastic_net_fit_bin *Elastic Net Fit Binary*

Description

A fitted model with the elastic_net function and binary data.

Usage

```
elastic_net_fit_bin
```

Format

A `elastic_net` object fitted with `example_data`, considering as subgroup variables all the covariates x_1, ..., x_10 and `binary` response type.

Source

This is an artificial example.

`elastic_net_fit_surv` *Elastic Net Fit Survival*

Description

A fitted model with the elastic_net function and survival data.

Usage

```
elastic_net_fit_surv
```

Format

A `elastic_net` object fitted with `example_data`, considering as subgroup variables all the covariates x_1, ..., x_10 and `survival` response type.

Source

This is an artificial example.

`elastic_net_surv` *H0, Coefficients Elastic Net Survival Model and Matrices*

Description

List with the cumulative baseline hazard, with the estimated coefficients of the model fitted with an elastic net and survival data, with the general design matrix and with the design and dummy matrices for subgroup x_1a.

Usage

```
elastic_net_surv
```

Format

A `list` with three elements:

- `h0`: the cumulative baseline hazard.
- `est_coef`: the estimated coefficients of the fitted model.
- `x`: the design matrix of the model.
- `dummy1`: the dummy matrix of the model for subgroup x_1a.
- `design1`: the design matrix of the model for subgroup x_1a.

Source

This is an artificial example.

`est_coef_bin1` *Estimated coefficients elastic net Binary*

Description

The coefficients of the model `elastic_net_fit_bin` when lambda takes the value that leads to minimal cross validation error.

Usage

```
est_coef_bin1
```

Format

A `matrix` with the coefficients of the elastic net model.

Source

This is an artificial example.

example_data *Example data*

Description

A simulated data set for a realistic scenario of a clinical trial.

Usage

```
example_data
```

Format

A `data.frame` with 1000 rows and 14 columns:

- `id`: Integer variable with the id of the patient.
- `arm`: Factor variable with 2 levels: control and treatment.
- `x_1`, `x_2`, `x_3`, `x_4`, `x_5`, `x_6`, `x_7`, `x_8`, `x_9`, `x_10`: Factor variables that represent covariates of the patients (e.g. sex, region, race, etc.).
- `tt_pfs`: Continuous variable with the progression free survival time of each patient.
- `ev_pfs`: Binary variable with the progression free survival status of each patient. 0 indicates that the event of interest did not happen and 1 indicates that it did.

Source

This is an artificial dataset.

generate_stacked_data *Generation of Stacked Data by Subgroups*

Description

Function to generate stacked data by the subgroups considered in the model.

Usage

```
generate_stacked_data(
  base_model,
  subgroup_model,
  data,
  resptype = c("survival", "binary")
)
```

Arguments

base_model	(formula) the formula of the basic model that is going to be fitted. Depending on resptype it will be the formula corresponding to a coxph model or to a glm model. In the case of "survival" a formula like Surv(time, status) ~ trt would be expected and in the "binary" case a formula like y ~ trt.
subgroup_model	(formula) the formula with all the subgroup variables.
data	(data frame) the data frame with the variables.
resptype	(string) the type of data used. Can be "survival" or "binary".

Value

Data frame of the stacked data.

Examples

```
generate_stacked_data(Surv(tt_pfs, ev_pfs) ~ arm, ~ x_1 + x_2, example_data, "survival")
```

---

horseshoe *Bayesian Shrinkage Model Estimation*

Description

Function to fit a bayesian shrinkage model with a regularized horseshoe prior on the interaction coefficients between the subgrouping covariates and the treatment.

Usage

```
horseshoe(
  resp,
  trt,
  subgr,
  covars,
  data,
  resptype = c("survival", "binary"),
  status = NULL,
  ...
)
```

Arguments

resp	(string) the response variable name.
trt	(string) the treatment variable name.The treatment variable must be a factor with 2 levels where the first level is the control and the second one the treatment.
subgr	(character) vector with the name of the subgroup variables from which we want to obtain the subgroup treatment effect. They have to be factor variables with the subgroups as levels.
covars	(character) vector with the name of the variables that we want to include in the model. They have to be factor variables with the subgroups as levels. The subgr variables have to be included here.
data	(data frame) the data frame with the variables.
resptype	(string) the type of data used. Can be "survival" or "binary".
status	(string) only for "survival" resptype, the status variable name in survival data.
...	Additional arguments from the brm function.

Value

List with fit, model, resptype, data, alpha, design_matrix, design_dummy, y, subgr_names.

Examples

```
horseshoe("ev_pfs", "arm", c("x_1", "x_2"), c("x_1", "x_2", "x_3"),
  example_data, "binary",
  chains = 1, seed = 0, control = list(adapt_delta = 0.95),
  iter = 50 # In practice, you need to omit this or set it much higher!
)
```

---

horseshoe_fit_bin *Horseshoe Fit Binary*

Description

A fitted model with the horseshoe function and binary data.

Usage

```
horseshoe_fit_bin
```

Format

A horseshoe object fitted with example_data, considering as subgroup variables just the covariates x_1 and x_2 and binary response type.

Source

This is an artificial example.

---

horseshoe_fit_surv *Horseshoe Fit Survival*

Description

A fitted model with the horseshoe function and survival data.

Usage

```
horseshoe_fit_surv
```

Format

A horseshoe object fitted with example_data, considering as subgroup variables just the covariates x_1 and x_2 and survival response type.

Source

This is an artificial example.

---

lor_estimation *Estimation of Log-Odds Ratio*

Description

Function to estimate the subgroup log-odds ratio for binary data.

Usage

```
lor_estimation(x_subg, dummy_subg, est_coef)
```

Arguments

x_subg	(matrix) the matrix with the subgroup covariates.
dummy_subg	(matrix) the dummy matrix with the subgroup covariates.
est_coef	(matrix) the estimated coefficients from the fitted model.

Value

Log-odds ratio of the studied subgroup.

Examples

```
lor_estimation(design_matrix1, design_dummy1, est_coef_bin1)
```

`naive` *Naive Model Estimation*

Description

Function to fit the naive models to the data of each one of the subgroups.

Usage

```
naive(
  resp,
  trt,
  subgr,
  data,
  resptype = c("survival", "binary"),
  status = NULL
)
```

Arguments

`resp`	(`string`) the response variable name.
`trt`	(`string`) the treatment variable name. The treatment variable must be a factor with 2 levels where the first level is the control and the second one the treatment.
`subgr`	(`character`) vector with the name of the subgrouping variables. These variables have to be `factor` variables with the subgroups as levels.
`data`	(`data frame`) the data frame with the variables.
`resptype`	(`string`) the type of data used. Can be "survival" or "binary".
`status`	(`string`) only for "survival" `resptype`, the status variable name in survival data.

Value

List with `fit`, `estimates`, `model`, `resptype`, `data`.

Examples

```
naive("tt_pfs", "arm", c("x_1", "x_2"), example_data, "survival", "ev_pfs")
```

`naive_fit_bin` *Naive Fit Binary*

Description

A fitted model with the naive function and binary data.

Usage

```
naive_fit_bin
```

Format

A `naive` object fitted with `example_data`, considering as subgroup variables all the covariates x_1, ..., x_10 and `binary` response type.

Source

This is an artificial example.

`naive_fit_surv` *Naive Fit Survival*

Description

A fitted model with the naive function and survival data.

Usage

```
naive_fit_surv
```

Format

A `naive` object fitted with `example_data`, considering as subgroup variables all the covariates x_1, ..., x_10 and `survival` response type.

Source

This is an artificial example.

naivepop *Naive Overall Population Model Estimation*

Description

Function to fit the naive overall population model to the data. This model is going to estimate the subgroup treatment effects with the overall population treatment effect.

Usage

```
naivepop(resp, trt, data, resptype = c("survival", "binary"), status = NULL)
```

Arguments

resp	(string) the response variable name.
trt	(string) the treatment variable name. The treatment variable must be a factor with 2 levels where the first level is the control and the second one the treatment.
data	(data frame) the data frame with the variables.
resptype	(string) the type of data used. Can be "survival" or "binary".
status	(string) only for "survival" resptype, the status variable name in survival data.

Value

List with fit, model, resptype, data.

Examples

```
naivepop("tt_pfs", "arm", example_data, "survival", "ev_pfs")
```

naivepop_fit_bin *Naivepop Fit Binary*

Description

A fitted model with the naivepop function and binary data.

Usage

```
naivepop_fit_bin
```

Format

A naivepop object fitted with example_data, considering as subgroup variables all the covariates x_1, ..., x_10 and binary response type.

Source

This is an artificial example.

---

naivepop_fit_surv *Naivepop Fit Survival*

Description

A fitted model with the naivepop function and survival data.

Usage

```
naivepop_fit_surv
```

Format

A naivepop object fitted with example_data, considering as subgroup variables all the covariates x_1, ..., x_10 and survival response type.

Source

This is an artificial example.

---

plot.compare.data *Compare Forest Plots*

Description

Function to obtain a forest plot with all the different fitted models in order to compare their performance.

Usage

```
## S3 method for class 'compare.data'
plot(x, ...)
```

Arguments

x	(compare.data) object with the data of treatment effects estimated with the different methods.
...	Additional arguments to plot.

Value

Forest plot with all the methods that are compared.

Examples

```
plot(compare(naivepop_fit_surv, naive_fit_surv, elastic_net_fit_surv))
```

`plot.summary.elastic_net`
Forest plot Summary Elastic Net

Description

Plot the forest plot of the summary of a `elastic_net` object.

Usage

```
## S3 method for class 'summary.elastic_net'
plot(x, ...)
```

Arguments

`x`	(`summary.elastic_net`) summary of a `elastic_net` object.
`...`	Additional arguments to plot.

Value

Forest plot

Examples

```
plot(summary(elastic_net_fit_surv))
```

plot.summary.horseshoe *Forest plot Summary Horseshoe*

Description

Plot the forest plot of the summary of a horseshoe object.

Usage

```
## S3 method for class 'summary.horseshoe'
plot(x, ...)
```

Arguments

x	(summary.horseshoe) summary of a horseshoe object.
...	Additional arguments to plot.

Value

Forest plot

Examples

```
plot(summary(horseshoe_fit_bin))
```

plot.summary.naive *Forest plot Summary Naive*

Description

Plot the forest plot of the summary of a naive object.

Usage

```
## S3 method for class 'summary.naive'
plot(x, ...)
```

Arguments

x	(summary.naive) summary of a naive object.
...	Additional arguments to plot.

Value

Forest plot

Examples

```
plot(summary(naive_fit_surv))
```

`preprocess` *Data Preprocessing*

Description

Function to preprocess the data and obtain the design matrices corresponding to a model with just interaction of the covariates with the treatment, to a model with just the main effects and to a model just with the covariates and one-hot encoding.

Usage

```
preprocess(trt, subgr, covars, data)
```

Arguments

`trt`	(`string`) the treatment variable name. The treatment variable must be a factor with 2 levels where the first level is the control and the second one the treatment.
`subgr`	(`character`) vector with the name of the subgroup variables from which we want to obtain the subgroup treatment effect.
`covars`	(`character`) vector with the name of the variables that we want to include in the model. The `subgr` variables have to be included here.
`data`	(`data frame`) the data frame with the variables.

Value

List with `design_ia`, `design_main`, `design_dummy`, `subgr_names`.

Examples

```
preprocess("arm", c("x_1", "x_2"), c("x_1", "x_2", "x_3"), example_data)
```

print.summary.elastic_net

Print Function for Elastic Net Summary

Description

Function that prints the summary of a elastic_net object

Usage

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

Arguments

x	(summary.elastic_net) Summary of a elastic_net object.
...	Other arguments of print.

Value

The summary of the object.

Examples

```
print(summary(elastic_net_fit_bin))
```

print.summary.horseshoe

Print Function for Horseshoe Summary

Description

Function that prints the summary of a horseshoe object

Usage

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

Arguments

x	(summary.horseshoe) Summary of a horseshoe object.
...	Other arguments of print.

Value

The summary of the object.

Examples

```
print(summary(horseshoe_fit_bin))
```

`print.summary.naive` *Print Function for Naive Summary*

Description

Function that prints the summary of a naive object

Usage

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

Arguments

`x`	(`summary.naive`) Summary of a `naive` object.
`...`	Other arguments of print.

Value

The summary of the object.

Examples

```
print(summary(naive_fit_bin))
```

`print.summary.naivepop`
Print Function for Naivepop Summary

Description

Function that prints the summary of a naivepop object

Usage

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

Arguments

x	(summary.naivepop) Summary of a naivepop object.
...	Other arguments of print.

Value

The summary of the object.

Examples

```
print(summary(naivepop_fit_bin))
```

---

simul_covariates *Generation of a Design Matrix for Simulations*

Description

This function uses a block diagonal covariance matrix for the underlying multivariate normal data to create the design matrix in blocks of 10, see the details.

Usage

```
simul_covariates(n, p_catvar = 10, add_contvars = FALSE, arm_factor = FALSE)
```

Arguments

n	(count) number of rows (observations).
p_catvar	(count) number of covariates (excluding treatment arm).
add_contvars	(flag) whether to add continuous covariates.
arm_factor	(flag) whether to make the arm variable a factor.

Details

The following pattern is repeated for the covariate blocks:

- The first 5 covariates are uncorrelated with everything.
- The covariates 6 to 8 have "moderate" correlation (0.25) between each other.
- The covariates 9 and 10 have "high" correlation (0.5).

By default, only the resulting categorical covariates obtained by thresholding are included. Optionally also the original continuous covariates are included in the returned design matrix.

Value

The design matrix.

Examples

```
simul_covariates(n = 10, p_catvar = 3, add_contvars = FALSE)
simul_covariates(n = 10, p_catvar = 3, add_contvars = TRUE)
simul_covariates(n = 10, p_catvar = 3, add_contvars = TRUE, arm_factor = TRUE)
```

---

simul_data *Simulate Covariates and Progression Free Survival Data*

Description

This combines the covariates simulation via `simul_covariates()` with 10 categorical covariates, and the PFS simulation via `simul_pfs()`.

Usage

```
simul_data(n, add_interaction = FALSE, coefs, ...)
```

Arguments

`n`	(`count`) number of patients.
`add_interaction`	(`flag`) whether to add interaction terms between covariates 1 and 2.
`coefs`	(`numeric`) named vector of coefficients to set.
`...`	additional parameters apart from the linear predictor values needed for `simul_pfs()`.

Details

Regression coefficients are for an AFT with over-parametrized dummy coding for arm-subgroup interactions.

Value

A combined `data.frame` with the `id` column, the design matrix and the PFS outcomes.

Examples

```
set.seed(321)
simul_data(
  n = 100,
  coefs = c(arm1 = 1),
  sigma_aft = 1,
  recr_duration = 0.2,
  rate_cens = 2,
  n_events = 20
)
```

`simul_pfs` *Simulation of Progression Free Survival Times*

Description

Simulation of Progression Free Survival Times

Usage

```
simul_pfs(
  lp_aft,
  sigma_aft,
  recr_duration,
  rate_cens,
  n_events,
  add_uncensored_pfs = FALSE
)
```

Arguments

`lp_aft`	(`numeric`) linear predictor values for the accelerate failure time model (AFT).
`sigma_aft`	(`number`) standard deviation for the AFT model.
`recr_duration`	(`number`) duration of recruitment.
`rate_cens`	(`number`) rate for the exponentially distributed censoring process.
`n_events`	(`count`) number of events to reach for the study end.
`add_uncensored_pfs`	(`flag`) whether to add the uncensored PFS as well to the resulting `data.frame`.

Value

A `data.frame` with columns `tt_pfs` (PFS time) and `ev_pfs` (corresponding event indicator with 1 for an event and 0 for censored), and optionally `tt_pfs_uncens`.

Examples

```
set.seed(123)
simul_pfs(
  lp_aft = rnorm(100),
  sigma_aft = 1,
  recr_duration = 0.2,
  rate_cens = 2,
  n_events = 20
)
```

---

subgroups *Subgroup Treatment Effect*

Description

Function to obtain the estimated treatment effect in each one of the considered subgroups.

Usage

```
subgroups(object, est_coef, h0 = NULL, gamma = 1)
```

Arguments

`object`	(`bonsaiforest`) the fitted bonsaiforest object.
`est_coef`	(`matrix`) the estimated coefficients from the fitted model.
`h0`	(`numeric`) the vector with the cumulative baseline hazard. Present just for `resptype` survival.
`gamma`	(`scalar`) numeric value defining the weights to obtain the average hazard ratio. Default is 1 (in this case the average hazard ratio obtained can be interpreted as the odds of concordance).

Value

`data.frame` with the subgroup names and with the estimated subgroup treatment effects.

Examples

```
subgroups(
  elastic_net_fit_surv, elastic_net_surv$est_coef,
  elastic_net_surv$h0
)
```

`summary.elastic_net` *Summary Elastic Net Function*

Description

Function to obtain the naive subgroup treatment effects of an object fitted with the `elastic_net` function.

Usage

```
## S3 method for class 'elastic_net'
summary(object, gamma = 1, l = NULL, lambda = NULL, ...)
```

Arguments

`object`	(`elastic_net`) the elastic_net object.
`gamma`	(`scalar`) numeric value defining the weights to obtain the average hazard ratio. Default is 1 (in this case the average hazard ratio obtained can be interpreted as the odds of concordance). Just needed when using survival data.
`l`	(`scalar`) the maximum value of time that wants to be studied to obtain the average hazard ratio. Default is the maximum value of time when there was an event. Just needed when using survival data.
`lambda`	(`scalar`) the penalization constant in the elastic net. Default is the value that leads to minimal cross validation error.
`...`	Arguments of summary

Value

Object of class `summary.elastic_net` which is a `list` with the estimated subgroup treatment effects, the `resptype`, the confidence level and the value of `alpha`.

Examples

```
summary(elastic_net_fit_surv)
```

summary.horseshoe *Summary Horseshoe Function*

Description

Function to obtain the estimated subgroup treatment effects from a horseshoe model and a credible interval for them.

Usage

```
## S3 method for class 'horseshoe'
summary(object, conf = 0.95, gamma = 1, l = NULL, m = 50, ...)
```

Arguments

object	(horseshoe) the horseshoe object.
conf	(scalar) the level of the credible intervals. Default is 0.95.
gamma	(scalar) numeric value defining the weights to obtain the average hazard ratio. Default is 1 (in this case the average hazard ratio obtained can be interpreted as the odds of concordance). Just needed when using survival data.
l	(scalar) the maximum value of time that wants to be studied to obtain the average hazard ratio. Default is the maximum value of time when there was an event. Just needed when using survival data.
m	(scalar) the value that defines the equally spaced time points where the survival curves are going to be studied. Default is 50. Just needed when using survival data.
...	Arguments of summary.

Value

Object of class summary.horseshoe which is a list with the approximated posterior distribution of the treatment effects, a data.frame with the estimated subgroup treatment effect (with the median) and the bounds of the credible intervals, the resptype and the confidence level.

Examples

```
summary(horseshoe_fit_bin)
```

summary.naive *Summary Naive*

Description

Function to obtain the naive subgroup treatment effects of an object fitted with the `naive` function.

Usage

```
## S3 method for class 'naive'
summary(object, conf = 0.95, ...)
```

Arguments

`object`	(`naive`) the naive object.
`conf`	(`scalar`) the confidence level of the intervals. Default is 0.95.
`...`	Arguments of summary.

Value

Object of class `summary.elastic_net` which is a `list` with a `data.frame` with 4 columns (the subgroup variables, the estimated treatment effect and the low and high bounds of the confidence interval of the treatment effect), the `resptype` and the confidence level.

Examples

```
summary(naive_fit_surv)
```

summary.naivepop *Summary Naivepop Function*

Description

Function to obtain the overall subgroup treatment effect of a naivepop object.

Usage

```
## S3 method for class 'naivepop'
summary(object, ...)
```

Arguments

`object`	(`naivepop`) the naivepop object.
`...`	Arguments of summary.

Value

Object of class `summary.naivepop` which is a `list` with the estimated subgroup treatment effects and the `resptype`.

Examples

```
summary(naivepop_fit_surv)
```

`survival_curves` *Average Survival Curves*

Description

Function to obtain the average survival curve from all individual survival curves.

Usage

```
survival_curves(x, h0, est_coef)
```

Arguments

`x`	(`matrix`) the matrix with the subgroup covariates.
`h0`	(`numeric`) the vector with the cumulative baseline hazard.
`est_coef`	(`matrix`) the estimated coefficients from the fitted model.

Value

The survival probabilities at the sorted event times.

Examples

```
survival_curves(
  elastic_net_surv$x, elastic_net_surv$h0,
  elastic_net_surv$est_coef
)
```

`trt_horseshoe` *Subgroup Treatment Effect Horseshoe*

Description

Function to obtain the estimated posterior distribution of the subgroup treatment effects considering a horseshoe fitted model.

Usage

```
trt_horseshoe(object, gamma = 1, l = NULL, m = 50)
```

Arguments

`object`	(`horseshoe`) the horseshoe object.
`gamma`	(`scalar`) numeric value defining the weights to obtain the average hazard ratio. Default is 1 (in this case the average hazard ratio obtained can be interpreted as the odds of concordance). Just needed when using survival data.
`l`	(`scalar`) the maximum value of time that wants to be studied to obtain the average hazard ratio. Default is the maximum value of time when there was an event. Just needed when using survival data.
`m`	(`scalar`) the value that defines the equally spaced time points where the survival curves are going to be studied. Default is 50. Just needed when using survival data.

Value

Approximated posterior distribution of the subgroup treatment effects.

Examples

```
trt_horseshoe(horseshoe_fit_surv, m = 1)
```

Index

∗ **datasets**
  design_dummy1, 5
  design_matrix1, 6
  elastic_net_fit_bin, 7
  elastic_net_fit_surv, 8
  elastic_net_surv, 8
  est_coef_bin1, 9
  example_data, 10
  horseshoe_fit_bin, 12
  horseshoe_fit_surv, 13
  naive_fit_bin, 15
  naive_fit_surv, 15
  naivepop_fit_bin, 16
  naivepop_fit_surv, 17

ahr_estimation, 3
ahr_from_km, 4

compare, 5

design_dummy1, 5
design_matrix1, 6

elastic_net, 6
elastic_net_fit_bin, 7
elastic_net_fit_surv, 8
elastic_net_surv, 8
est_coef_bin1, 9
example_data, 10

generate_stacked_data, 10

horseshoe, 11
horseshoe_fit_bin, 12
horseshoe_fit_surv, 13

lor_estimation, 13

naive, 14
naive_fit_bin, 15
naive_fit_surv, 15
naivepop, 16
naivepop_fit_bin, 16
naivepop_fit_surv, 17

plot.compare.data, 17
plot.summary.elastic_net, 18
plot.summary.horseshoe, 19
plot.summary.naive, 19
preprocess, 20
print.summary.elastic_net, 21
print.summary.horseshoe, 21
print.summary.naive, 22
print.summary.naivepop, 22

simul_covariates, 23
simul_covariates(), *24*
simul_data, 24
simul_pfs, 25
simul_pfs(), *24*
subgroups, 26
summary.elastic_net, 27
summary.horseshoe, 28
summary.naive, 29
summary.naivepop, 29
survival_curves, 30

trt_horseshoe, 31