Package 'vaccine'

July 5, 2026

Type Package

Title Statistical Tools for Immune Correlates Analysis of Vaccine Clinical Trial Data

Version 1.4.0

Description Various semiparametric and nonparametric statistical tools for immune correlates analysis of vaccine clinical trial data. This includes calculation of summary statistics and estimation of risk, vaccine efficacy, controlled effects (controlled risk and controlled vaccine efficacy), and mediation effects (natural direct effect, natural indirect effect, proportion mediated). See Gilbert P, Fong Y, Kenny A, and Carone, M (2022) <doi:10.1093/biostatistics/kxac024> and Fay MP and Follmann DA (2023) <doi:10.48550/arXiv.2208.06465>.

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

VignetteBuilder knitr

Imports dplyr (>= 1.0.10), fdrtool (>= 1.2.17), ggplot2 (>= 3.4.0), ggpubr (>= 0.5.0), methods (>= 4.2.2), rlang (>= 1.0.6), Rsolnp (>= 1.16), splines (>= 4.2.2), stats (>= 4.2.2), SuperLearner (>= 2.0.28), survival (>= 3.4.0), survML (>= 0.0.0.9000), truncnorm (>= 1.0.8), e1071 (>= 1.7.12), gam (>= 1.22.3), ranger (>= 0.14.0), utils (>= 4.3.2), MASS (>= 7.3.61)

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

Depends R (>= 2.10)

Config/testthat/edition 3

URL https://avi-kenny.github.io/vaccine/

BugReports https://github.com/Avi-Kenny/vaccine/issues

NeedsCompilation no

Author Avi Kenny [aut, cre]

Maintainer Avi Kenny <avi.kenny@gmail.com>

Repository CRAN

Date/Publication 2026-07-05 13:40:02 UTC

Contents

as_table 2
diagnostics 3
est_ce 4
est_med 6
est_overall 7
hvtn505 8
load_data 9
params_ce_cox 10
params_ce_np 11
params_med_cox 13
params_med_np 13
plot_ce 15
summary_stats 16
trim 17

Index **19**

`as_table` *Create table of estimates*

Description

Format estimates returned by `est_ce` as a table

Usage

```
as_table(..., which = "CR", labels = NA)
```

Arguments

`...`	One or more objects of class `"vaccine_est"` returned by `est_ce`.
`which`	One of c("CR", "CVE"); controls whether the table contains CR or CVE values.
`labels`	A character vector of length equal to length(list(...)) representing curve labels

Value

A table of CR or CVE values

Examples

```
data(hvtn505)
dat <- load_data(time="HIVwk28preunblfu", event="HIVwk28preunbl", vacc="trt",
                 marker="IgG_V2", covariates=c("age","BMI","bhvrisk"),
                 weights="wt", ph2="casecontrol", data=hvtn505)

ests_cox <- est_ce(dat=dat, type="Cox", t_0=578)
ests_np <- est_ce(dat=dat, type="NP", t_0=578)
ests_table <- as_table(ests_cox, ests_np)
head(ests_table)
```

`diagnostics` *Run diagnostics*

Description

Run a set of diagnostic plots. Note that for this function to work, `est_ce` must be run with `return_extras=T`.

Usage

```
diagnostics(obj)
```

Arguments

`obj`	An object of class `vaccine_est` returned by `est_ce`

Value

A combined plot of model diagnostics

Examples

```
data(hvtn505)
dat <- load_data(time="HIVwk28preunblfu", event="HIVwk28preunbl", vacc="trt",
                 marker="IgG_V2", covariates=c("age","BMI","bhvrisk"),
                 weights="wt", ph2="casecontrol", data=hvtn505)

ests_np <- est_ce(dat=dat, type="NP", t_0=578, return_extras=TRUE)
diagnostics(ests_np)
```

est_ce *Estimate controlled effect curves*

Description

Estimate controlled risk (CR) curves and/or controlled vaccine efficacy (CVE) curves. See references for definitions of these curves.

Usage

```
est_ce(
  dat,
  type = "Cox",
  t_0,
  cr = TRUE,
  cve = FALSE,
  cr_placebo_arm = F,
 s_out = seq(from = min(dat$s, na.rm = TRUE), to = max(dat$s, na.rm = TRUE), l = 101),
  ci_type = "transformed",
  placebo_risk_method = "KM",
  return_p_value = FALSE,
  return_extras = FALSE,
  params_cox = params_ce_cox(),
  params_np = params_ce_np()
)
```

Arguments

dat	A data object returned by load_data
type	One of c("Cox", "NP"). This specifies whether to estimate the curve(s) using a marginalized Cox proportional hazards model or using a monotone-constrained nonparametric estimator.
t_0	Time point of interest
cr	Boolean. If TRUE, the controlled risk (CR) curve is computed and returned.
cve	Boolean. If TRUE, the controlled vaccine efficacy (CVE) curve is computed and returned.
cr_placebo_arm	Boolean. If TRUE, the CR curve is estimated for the placebo arm instead of the vaccine arm.
s_out	A numeric vector of s-values (on the biomarker scale) for which cve(s) and/or cr(s) are computed. Defaults to a grid of 101 points between the min and max biomarker values.
ci_type	One of c("transformed", "truncated", "regular", "none"). If ci_type="transformed", confidence intervals are computed on the logit(CR) and/or log(1-CVE) scale to ensure that confidence limits lie within [0,1] for CR and/or lie within (-inf,1] for CVE. If ci_type="truncated", confidence limits are constructed on the CR and/or

CVE scale but truncated to lie within [0,1] for CR and/or lie within (-inf,1] for CVE. If ci_type="regular", confidence limits are not transformed or truncated. If ci_type="none", confidence intervals are not computed.

`placebo_risk_method`
: One of c("KM", "Cox"). Method for estimating overall risk in the placebo group. "KM" computes a Kaplan-Meier estimate and "Cox" computes an estimate based on a marginalized Cox model survival curve. Only relevant if cve=TRUE.

`return_p_value`
: Boolean; if TRUE, a P-value corresponding to the null hypothesis that the CVE curve is flat is returned. The type of P-value corresponds to the `type` argument.

`return_extras`
: Boolean; if TRUE, objects useful for debugging are returned.

`params_cox`
: A list of options returned by `params_ce_cox` that are relevant if type="Cox".

`params_np`
: A list of options returned by `params_ce_np` that are relevant if type="NP".

Value

A list of the form `list(cr=list(...), cve=list(...))` containing CR and/or CVE estimates. Each of the inner lists contains the following:

- `s`: a vector of marker values corresponding to s_out
- `est`: a vector of point estimates
- `ci_lower`: a vector of confidence interval lower limits
- `ci_upper`: a vector of confidence interval upper limits

References

Gilbert P, Fong Y, Kenny A, and Carone, M (2022). A Controlled Effects Approach to Assessing Immune Correlates of Protection. <doi:10.1093/biostatistics/kxac024>

Examples

```
data(hvtn505)
dat <- load_data(time="HIVwk28preunblfu", event="HIVwk28preunbl", vacc="trt",
                 marker="IgG_V2", covariates=c("age","BMI","bhvrisk"),
                 weights="wt", ph2="casecontrol", data=hvtn505)

ests_cox <- est_ce(dat=dat, type="Cox", t_0=578)
ests_np <- est_ce(dat=dat, type="NP", t_0=578)
```

`est_med` *Estimate mediation effects*

Description

Estimate mediation effects, including the natural direct effect (NDE), the natural indirect effect (NIE), and the proportion mediated (PM). See references for definitions of these objects.

Usage

```
est_med(
  dat,
  type = "NP",
  t_0,
  nde = TRUE,
  nie = TRUE,
  pm = TRUE,
  scale = "RR",
  params_cox = params_med_cox(),
  params_np = params_med_np()
)
```

Arguments

`dat`	A data object returned by load_data
`type`	One of c("NP", "Cox"). This specifies whether to estimate the effects using a marginalized Cox proportional hazards model or using a nonparametric estimator.
`t_0`	Time point of interest
`nde`	Boolean. If TRUE, the natural direct effect is computed and returned.
`nie`	Boolean. If TRUE, the natural indirect effect is computed and returned.
`pm`	Boolean. If TRUE, the proportion mediated is computed and returned.
`scale`	One of c("RR", "VE"). This determines whether NDE and NIE estimates and CIs are computed on the risk ratio (RR) scale or the vaccine efficacy (VE) scale. The latter equals one minus the former.
`params_cox`	A list of options returned by `params_med_cox` that are relevant if type="Cox".
`params_np`	A list of options returned by `params_med_np` that are relevant if type="NP".

Value

A dataframe containing the following columns:

- `effect`: one of c("NDE", "NIE", "PM")
- `est`: point estimate

- se: standard error of point estimate
- ci_lower: a confidence interval lower limit
- ci_upper: a confidence interval upper limit

References

Fay MP and Follmann DA (2023). Mediation Analyses for the Effect of Antibodies in Vaccination <doi:10.48550/arXiv.2208.06465>

Examples

```
data(hvtn505)
dat <- load_data(time="HIVwk28preunblfu", event="HIVwk28preunbl", vacc="trt",
                 marker="IgG_V2", covariates=c("age","BMI","bhvrisk"),
                 weights="wt", ph2="casecontrol", data=hvtn505)

ests_cox <- est_med(dat=dat, type="Cox", t_0=578)
ests_np <- est_med(dat=dat, type="NP", t_0=578)
```

---

est_overall *Estimate overall risk and vaccine efficacy*

Description

Estimate overall risk and vaccine efficacy.

Usage

```
est_overall(dat, t_0, method = "Cox", risk = TRUE, ve = TRUE)
```

Arguments

dat	A data object returned by load_data
t_0	Time point of interest
method	One of c("KM", "Cox"), corresponding to either a Kaplan-Meier estimator ("KM") or a marginalized Cox proportional hazards model ("Cox").
risk	Boolean. If TRUE, the controlled risk (CR) curve is computed.
ve	Boolean. If TRUE, the controlled vaccine efficacy (CVE) curve is computed.

Value

A dataframe containing estimates

Examples

```
data(hvtn505)
dat <- load_data(time="HIVwk28preunblfu", event="HIVwk28preunbl", vacc="trt",
                 marker="IgG_V2", covariates=c("age","BMI","bhvrisk"),
                 weights="wt", ph2="casecontrol", data=hvtn505)
est_overall(dat=dat, t_0=578, method="KM")
```

`hvtn505` *HVTN 505 Dataset*

Description

A dataset from the HVTN 505 clinical trial. This is a filtered version of the original dataset (which contains 2,502 rows) based on the week28 variable, which defines eligibility for inclusion in a correlates analysis. Additionally, two rows were removed due to missing data.

Usage

```
data(hvtn505)
```

Format

A data frame with 2,302 rows and 10 variables:

- pub_id: Unique individual identifier
- trt: Treatment Assignment: 1=Vaccine, 0=Placebo
- HIVwk28preunbl: Indicator of HIV-1 infection diagnosis on/after study week 28 (day 196) prior to Unblinding Date (Apr 22, 2013).
- HIVwk28preunblfu: Follow-up time (in days) for HIV-1 infection diagnosis endpoint as of the Unblinding Date (22Apr2013) occuring on/after study week 28 (day 196).
- age: Age (in years) at randomization
- BMI: Body Mass Index: (Weight in Kg)/(Height in meters)**2
- bhvrisk: Baseline behavioral risk score
- casecontrol: Indicator of inclusion in the case-control cohort
- wt: Inverse-probability-of-sampling weights, for the case-control cohort
- IgG_env: IgG Binding to gp120/140
- IgG_V2: IgG Binding to V1V2
- IgG_V3: IgG Binding to V3

Source

https://atlas.scharp.org/project/HVTN%20Public%20Data/HVTN%20505/begin.view

load_data *Load and format data object*

Description

This function takes in user-supplied data and returns a data object that can be read in by summary_stats, est_ce, est_med, and other estimation functions. Data is expected to come from a vaccine clinical trial, possibly involving two-phase sampling and possibly including a biomarker of interest.

Usage

```
load_data(
  time,
  event,
  vacc,
  marker,
  covariates,
  weights,
  ph2,
  strata = NA,
  data,
  covariates_ph2 = FALSE
)
```

Arguments

time	A character string; the name of the numeric variable representing observed event or censoring times.
event	A character string; the name of the binary variable corresponding to whether the observed time represents an event time (1) or a censoring time (0). Either integer (0/1) or Boolean (T/F) values are allowed.
vacc	A character string; the name of the binary variable denoting whether the individual is in the vaccine group (1) or the placebo group (0). Accepts either integer (0/1) or Boolean (T/F) values.
marker	A character string; the name of the numeric variable of biomarker values.
covariates	A character vector; the names of the covariate columns. Columns values should be either numeric, binary, or factors. Character columns will be converted into factors.
weights	A character string; the name of the numeric variable containing inverse-probability-of-sampling (IPS) weights.
ph2	A character string; the name of the binary variable representing whether the individual is in the phase-two cohort (1) or not (0). Accepts either integer (0/1) or Boolean (T/F) values.
strata	A character string; the name of the variable containing strata identifiers (for two-phase sampling strata).

data	A dataframe containing the vaccine trial data.
covariates_ph2	A boolean; if at least one of the covariates is measured only in the phase-two cohort, set this to TRUE.

Value

An object of class vaccine_dat.

Examples

```
data(hvtn505)
dat <- load_data(time="HIVwk28preunblfu", event="HIVwk28preunbl", vacc="trt",
                 marker="IgG_V2", covariates=c("age","BMI","bhvrisk"),
                 weights="wt", ph2="casecontrol", data=hvtn505)
```

---

params_ce_cox *Set parameters controlling Cox model estimation of controlled effect curves*

Description

This should be used in conjunction with est_ce to set parameters controlling Cox model estimation of controlled effect curves; see examples.

Usage

```
params_ce_cox(spline_df = NA, spline_knots = NA, edge_ind = FALSE)
```

Arguments

spline_df	An integer; if the marker is modeled flexibly within the Cox model linear predictor as a natural cubic spline, this option controls the degrees of freedom in the spline; knots are chosen to be equally spaced across the range of the marker.
spline_knots	A numeric vector; as an alternative to specifying spline_df, the exact locations of the knots in the spline (including boundary knots) can be specified with this option.
edge_ind	Boolean. If TRUE, an indicator variable corresponding to the lower limit of the marker will be included in the Cox model linear predictor.

Value

A list of options.

Examples

```
data(hvtn505)
dat <- load_data(time="HIVwk28preunblfu", event="HIVwk28preunbl", vacc="trt",
                 marker="IgG_V2", covariates=c("age","BMI","bhvrisk"),
                 weights="wt", ph2="casecontrol", data=hvtn505)

ests_cox <- est_ce(
  dat = dat,
  type = "Cox",
  t_0 = 578,
  params_cox = params_ce_cox(spline_df=4)
)
```

`params_ce_np` *Set parameters controlling nonparametric estimation of controlled effect curves*

Description

This should be used in conjunction with `est_ce` to set parameters controlling nonparametric estimation of controlled effect curves; see examples.

Usage

```
params_ce_np(
  dir = "decr",
  edge_corr = FALSE,
  grid_size = list(y = 101, s = 101, x = 5),
  surv_type = "survML-G",
  density_type = "binning",
  density_bins = 15,
  deriv_type = "m-spline",
  convex_type = "GCM"
)
```

Arguments

`dir`	One of c("decr", "incr"); controls the direction of monotonicity. If dir="decr", it is assumed that CR decreases as a function of the marker. If dir="incr", it is assumed that CR increases as a function of the marker.
`edge_corr`	Boolean. If TRUE, the "edge correction" is performed to adjust for bias near the marker lower limit (see references). It is recommended that the edge correction is only performed if there are at least (roughly) 10 events corresponding to the marker lower limit.

`grid_size` A list with keys `y`, `s`, and `x`; controls the rounding of data values. Decreasing the grid size values results in shorter computation times, and increasing the values results in more precise estimates. If grid_size$s=101, this means that a grid of 101 equally-spaced points (defining 100 intervals) will be created from min(S) to max(S), and each S value will be rounded to the nearest grid point. For grid_size$y, a grid will be created from 0 to t_0, and then extended to max(Y). For grid_size$x, a separate grid is created for each covariate column (binary/categorical covariates are ignored).

`surv_type` One of c("Cox", "survSL", "survML-G", "survML-L"); controls the method to use to estimate the conditional survival and conditional censoring functions. If type="Cox", a survival function based on a Cox proportional hazard model will be used. If type="survSL", the Super Learner method of Westling 2023 is used. If type="survML-G", the global survival stacking method of Wolock 2022 is used. If type="survML-L", the local survival stacking method of Polley 2011 is used.

`density_type` One of c("binning", "parametric"); controls the method to use to estimate the density ratio f(S|X)/f(S).

`density_bins` An integer; if density_type="binning", the number of bins to use. If density_bins=0, the number of bins will be selected via cross-validation.

`deriv_type` One of c("m-spline", "linear"); controls the method to use to estimate the derivative of the CR curve. If deriv_type="linear", a linear spline is constructed based on the midpoints of the jump points of the estimated function (plus the estimated function evaluated at the endpoints), which is then numerically differentiated. deriv_type="m-spline" is similar to deriv_type="linear" but smooths the set of points (using the method of Fritsch and Carlson 1980) before differentiating.

`convex_type` One of c("GCM", "CLS"). Whether the greatest convex minorant ("GCM") or convex least squares ("CLS") projection should be used in the smoothing of the primitive estimator Gamma_n. convex_type="CLS" is experimental and should be used with caution.

Value

A list of options.

Examples

```
data(hvtn505)
dat <- load_data(time="HIVwk28preunblfu", event="HIVwk28preunbl", vacc="trt",
                 marker="IgG_V2", covariates=c("age","BMI","bhvrisk"),
                 weights="wt", ph2="casecontrol", data=hvtn505)

ests_np <- est_ce(
  dat = dat,
  type = "NP",
  t_0 = 578,
  params_np = params_ce_np(edge_corr=TRUE, surv_type="survML-L")
)
```

params_med_cox *Set parameters controlling Cox model estimation of mediation effects*

Description

This should be used in conjunction with est_med to set parameters controlling Cox model estimation of mediation effects; see examples.

Usage

```
params_med_cox(edge_ind = F)
```

Arguments

edge_ind	Boolean. If TRUE, an indicator variable corresponding to the lower limit of the marker will be included in the Cox model linear predictor.

Value

A list of options.

Examples

```
data(hvtn505)
dat <- load_data(time="HIVwk28preunblfu", event="HIVwk28preunbl", vacc="trt",
                 marker="IgG_V2", covariates=c("age","BMI","bhvrisk"),
                 weights="wt", ph2="casecontrol", data=hvtn505)

ests_med <- est_med(
  dat = dat,
  type = "Cox",
  t_0 = 578,
  params_np = params_med_cox(edge_ind=TRUE)
)
```

params_med_np *Set parameters controlling nonparametric estimation of mediation effects*

Description

This should be used in conjunction with est_med to set parameters controlling nonparametric estimation of mediation effects; see examples.

Usage

```
params_med_np(
  grid_size = list(y = 101, s = 101, x = 5),
  surv_type = "survML-G",
  density_type = "binning",
  density_bins = 15
)
```

Arguments

grid_size	A list with keys y, s, and x; controls the rounding of data values. Decreasing the grid size values results in shorter computation times, and increasing the values results in more precise estimates. If grid_size$s=101, this means that a grid of 101 equally-spaced points (defining 100 intervals) will be created from min(S) to max(S), and each S value will be rounded to the nearest grid point. For grid_size$y, a grid will be created from 0 to t_0, and then extended to max(Y). For grid_size$x, a separate grid is created for each covariate column (binary/categorical covariates are ignored).
surv_type	One of c("Cox", "survSL", "survML-G", "survML-L"); controls the method to use to estimate the conditional survival and conditional censoring functions. If type="Cox", a survival function based on a Cox proportional hazard model will be used. If type="survSL", the Super Learner method of Westling 2023 is used. If type="survML-G", the global survival stacking method of Wolock 2022 is used. If type="survML-L", the local survival stacking method of Polley 2011 is used.
density_type	One of c("binning", "parametric"); controls the method to use to estimate the density ratio f(S|X)/f(S).
density_bins	An integer; if density_type="binning", the number of bins to use. If density_bins=0, the number of bins will be selected via cross-validation.

Value

A list of options.

Examples

```
data(hvtn505)
dat <- load_data(time="HIVwk28preunblfu", event="HIVwk28preunbl", vacc="trt",
                 marker="IgG_V2", covariates=c("age","BMI","bhvrisk"),
                 weights="wt", ph2="casecontrol", data=hvtn505)

ests_med <- est_med(
  dat = dat,
  type = "NP",
  t_0 = 578,
  params_np = params_med_np(surv_type="survML-L")
)
```

plot_ce *Plotting controlled effect curves*

Description

Plot CR and/or CVE curves

Usage

```
plot_ce(
  ...,
  which = "CR",
  density_type = "none",
  dat = NA,
  dat_alt = NA,
  zoom_x = "zoom in",
  zoom_y = "zoom out",
  labels = NA,
  bw = "ucv",
  adjust = 1
)
```

Arguments

...	One or more objects of class "vaccine_est" returned by est_ce.
which	One of c("CR", "CVE"); controls whether to plot CR curves or CVE curves.
density_type	One of c("none", "kde", "kde edge"). Controls the type of estimator used for the background marker density plot. For "none", no density plot is displayed. For "kde", a weighted kernel density estimator is used. For "kde edge", a modified version of "kde" is used that allows for a possible point mass at the left edge of the marker distribution.
dat	The data object originally passed into est_ce, used for plotting densities. It is only necessary to pass this in if density_type is not set to "none".
dat_alt	Alternative data object; a list containing one or more dataframes, each of the form data.frame(s=..., weights=...). Column s contains biomarker values and column weights contains corresponding two-phase sampling weights. This can be used as an alternative to specifying dat, and is particularly useful for plotting multiple densities on a single plot. If plotting multiple densities, the order of the dataframes should correspond to the order of "vaccine_est" objects passed in. See examples.
zoom_x	Either one of c("zoom in", "zoom out") or a vector of length 2. Controls the zooming on the X-axis. The default "zoom in" will set the zoom limits to the plot estimates. Choosing "zoom out" will set the zoom limits to show the entire distribution of the marker. Entering a vector of length 2 will set the left and right zoom limits explicitly.

zoom_y	Either "zoom out" or a vector of length 2. Controls the zooming on the Y-axis. The default "zoom out" will show the entire vertical range of the estimates. Entering a vector of length 2 will set the lower and upper zoom limits explicitly.
labels	A character vector of length equal to the length of list(...), representing plot labels. Only used if length(list(...))>1.
bw	Bandwidth type for kernel density; passed to stats::density
adjust	Bandwidth adjustment factor for kernel density; passed to stats::density

Value

A plot of CR/CVE estimates

Examples

```
# Plot one curve
data(hvtn505)
dat <- load_data(time="HIVwk28preunblfu", event="HIVwk28preunbl", vacc="trt",
                 marker="IgG_V2", covariates=c("age","BMI","bhvrisk"),
                 weights="wt", ph2="casecontrol", data=hvtn505)
ests_cox <- est_ce(dat=dat, type="Cox", t_0=578)
plot_ce(ests_cox, density_type="kde", dat=dat)

# Trim display of plot according to quantiles of the biomarker distribution
ests_cox_tr <- trim(ests_cox, dat=dat, quantiles=c(0.05,0.95))
plot_ce(ests_cox_tr, density_type="kde", dat=dat)

# Plot multiple curves (same biomarker)
ests_np <- est_ce(dat=dat, type="NP", t_0=578)
plot_ce(ests_cox, ests_np, density_type="kde", dat=dat)

# Plot multiple curves (two different biomarkers)
dat2 <- load_data(time="HIVwk28preunblfu", event="HIVwk28preunbl", vacc="trt",
                  marker="IgG_env", covariates=c("age","BMI","bhvrisk"),
                  weights="wt", ph2="casecontrol", data=hvtn505)
ests_cox2 <- est_ce(dat=dat2, type="Cox", t_0=578)
dat_alt <- list(
  data.frame(s=dat$s[dat$a==1], weights=dat$weights[dat$a==1]),
  data.frame(s=dat2$s[dat2$a==1], weights=dat2$weights[dat2$a==1])
)
plot_ce(ests_cox, ests_cox2, density_type="kde", dat_alt=dat_alt)
```

---

summary_stats *Calculate summary statistics*

Description

TO DO

Usage

```
summary_stats(dat, quietly = FALSE)
```

Arguments

dat	A data object returned by 'load_data'.
quietly	Boolean. If true, output will not be printed.

Value

A list containing values of various summary statistics.

Examples

```
data(hvtn505)
dat <- load_data(time="HIVwk28preunblfu", event="HIVwk28preunbl", vacc="trt",
                 marker="IgG_V2", covariates=c("age","BMI","bhvrisk"),
                 weights="wt", ph2="casecontrol", data=hvtn505)
summary_stats(dat=dat)
```

---

trim *Trim data for plotting/reporting*

Description

Removes a subset of estimates returned by est_ce

Usage

```
trim(ests, dat, quantiles, placebo = FALSE)
```

Arguments

ests	An object of class "vaccine_est" returned by est_ce.
dat	The data object originally passed into est_ce.
quantiles	A vector of length 2 representing the quantiles of the marker distribution at which to trim the data; if, for example, quantiles=c(0.1,0.9) is specified, values outside the 10 (weighted) quantiles of the marker distribution will be trimmed.
placebo	Boolean; if TRUE, quantiles are computed based on the marker distribution in the placebo arm instead of the vaccine arm

Value

A modified copy of ests with the data trimmed.

Examples

```
data(hvtn505)
dat <- load_data(time="HIVwk28preunblfu", event="HIVwk28preunbl", vacc="trt",
                 marker="IgG_V2", covariates=c("age","BMI","bhvrisk"),
                 weights="wt", ph2="casecontrol", data=hvtn505)
ests_cox <- est_ce(dat=dat, type="Cox", t_0=578)
ests_cox_tr <- trim(ests_cox, dat=dat, quantiles=c(0.05,0.95))
plot_ce(ests_cox_tr, density_type="kde", dat=dat)
```

Index

∗ **datasets**
hvtn505, 8

as_table, 2

diagnostics, 3

est_ce, *2, 3*, 4, *9–11*, *15*, *17*
est_med, 6, *9*, *13*
est_overall, 7

hvtn505, 8

load_data, 9

params_ce_cox, *5*, 10
params_ce_np, *5*, 11
params_med_cox, *6*, 13
params_med_np, *6*, 13
plot_ce, 15

summary_stats, *9*, 16

trim, 17