

Package ‘causaldrf’

October 18, 2015

Type Package

Title Tools for Estimating Causal Dose Response Functions

Version 0.2

Date 2015-10-16

Description Functions and data to estimate causal dose response functions given continuous, ordinal, or binary treatments.

License MIT + file LICENSE

LazyData TRUE

Depends R(>= 3.1.2)

Imports mgcv, splines, stats, survey,

Suggests BayesTree, dplyr, foreign, Hmisc, knitr, MASS, nnet, reshape2, rmarkdown, sas7bdat, testthat, tidy

NeedsCompilation no

Author Douglas Galagate [aut, cre]

Maintainer Douglas Galagate <galagated@gmail.com>

Repository CRAN

Date/Publication 2015-10-18 01:10:16

R topics documented:

add_spl_est	2
aipwee_est	4
bart_est	7
gam_est	9
get_ci	11
hi_est	11
hi_sim_data	14
iptw_est	15
ismw_est	17
iw_est	20
nmes_data	22

nw_est	23
overlap_fun	25
prop_spline_est	27
reg_est	29
scalar_wts	32
sim_data	33
t_mod	34
wtrg_est	36

Index	38
--------------	-----------

add_spl_est	<i>The additive spline estimator</i>
-------------	--------------------------------------

Description

additive spline estimator described in Bia et al. (2014).

Usage

```
add_spl_est(Y,
            treat,
            treat_formula,
            data,
            grid_val,
            knot_num,
            treat_mod,
            link_function,
            ...)
```

Arguments

Y	is the the name of the outcome variable contained in data.
treat	is the name of the treatment variable contained in data.
treat_formula	an object of class "formula" (or one that can be coerced to that class) that regresses treat on a linear combination of X: a symbolic description of the model to be fitted.
data	is a dataframe containing Y, treat, and X.
grid_val	contains the treatment values to be evaluated.
knot_num	is the number of knots used in outcome model
treat_mod	a description of the error distribution to be used in the model for treatment. Options include: "Normal" for normal model, "LogNormal" for lognormal model, "Poisson" for Poisson model, "NegBinom" for negative binomial model, "Gamma" for gamma model.
link_function	is either "log", "inverse", or "identity" for the "Gamma" treat_mod.
...	additional arguments to be passed to the outcome regression fitting function.

Value

add_spl_est returns an object of class "causaldrf", a list that contains the following components:

param	parameter estimates for a add_spl fit.
t_mod	the result of the treatment model fit.
out_mod	the result of the outcome model fit.
call	the matched call.

References

Schafer, J.L., Galagate, D.L. (2015). Causal inference with a continuous treatment and outcome: alternative estimators for parametric dose-response models. *Manuscript in preparation*.

Bia, Michela, et al. (2014). A Stata package for the application of semiparametric estimators of dose response functions. *Stata Journal* **14.3**, 580-604.

See Also

[nw_est](#), [iw_est](#), [hi_est](#), [gam_est](#), [bart_est](#), etc. for other estimates.

[t_mod](#), [overlap_fun](#) to prepare the data for use in the different estimates.

Examples

```
## Example from Schafer (2015).
example_data <- sim_data
add_spl_list <- add_spl_est(Y = Y,
  treat = T,
  treat_formula = T ~ B.1 + B.2 + B.3 + B.4 + B.5 + B.6 + B.7 + B.8,
  data = example_data,
  grid_val = seq(8, 16, by = 1),
  knot_num = 3,
  treat_mod = "Normal")

sample_index <- sample(1:1000, 100)
plot(example_data$T[sample_index],
  example_data$Y[sample_index],
  xlab = "T",
  ylab = "Y",
  main = "additive spline estimate")

lines(seq(8, 16, by = 1),
  add_spl_list$param,
  lty = 2,
  lwd = 2,
  col = "blue")
legend('bottomright',
  "additive spline estimate",
  lty=2,
  lwd = 2,
  col = "blue",
```

```

    bty='Y', cex=1)

rm(example_data, add_spl_list, sample_index)

## See Vignette for more examples.

```

aipwee_est

Prediction with a residual bias correction estimator

Description

This method combines the regression estimator with a residual bias correction for estimating a parametric ADRF.

Usage

```

aipwee_est(Y,
           treat,
           covar_formula = ~ 1,
           covar_lin_formula = ~ 1,
           covar_sq_formula = ~ 1,
           data,
           e_treat_1 = NULL,
           e_treat_2 = NULL,
           e_treat_3 = NULL,
           e_treat_4 = NULL,
           degree = 1,
           wt = NULL,
           method = "same",
           spline_df = NULL,
           spline_const = 1,
           spline_linear = 1,
           spline_quad = 1)

```

Arguments

Y	is the the name of the outcome variable contained in data.
treat	is the name of the treatment variable contained in data.
covar_formula	is the formula to describe the covariates needed to estimate the constant term: ~ X.1 + Can include higher order terms or interactions. i.e. ~ X.1 + I(X.1^2) + X.1 * X.2 + . . . Don't forget the tilde before listing the covariates.
covar_lin_formula	is the formula to describe the covariates needed to estimate the linear term, t: ~ X.1 + Can include higher order terms or interactions. i.e. ~ X.1 + I(X.1^2) + X.1 * X.2 + . . . Don't forget the tilde before listing the covariates.

covar_sq_formula	is the formula to describe the covariates needed to estimate the quadratic term, t^2 : $\sim X.1 + \dots$. Can include higher order terms or interactions. i.e. $\sim X.1 + I(X.1^2) + X.1 * X.2 + \dots$. Don't forget the tilde before listing the covariates.
data	is a dataframe containing Y, treat, and X.
e_treat_1	a vector, representing the conditional expectation of treat from T_mod.
e_treat_2	a vector, representing the conditional expectation of treat ² from T_mod.
e_treat_3	a vector, representing the conditional expectation of treat ³ from T_mod.
e_treat_4	a vector, representing the conditional expectation of treat ⁴ from T_mod.
degree	is 1 for linear and 2 for quadratic outcome model.
wt	is weight used in lsfit for outcome regression. Default is wt = NULL.
method	is "same" if the same set of covariates are used to estimate the constant, linear, and/or quadratic term. If method = "different", then different sets of covariates can be used to estimate the constant, linear, and/or quadratic term. covar_lin_formula and covar_sq_formula must be specified if method = "different".
spline_df	degrees of freedom. The default, spline_df = NULL, corresponds to no knots.
spline_const	is the number of spline terms needed to estimate the constant term.
spline_linear	is the number of spline terms needed to estimate the linear term.
spline_quad	is the number of spline terms needed to estimate the quadratic term.

Value

aipwee_est returns an object of class "causaldrf_lsfit", a list that contains the following components:

param	parameter estimates for a add_spl fit.
t_mod	the result of the treatment model fit.
out_mod	the result of the outcome model fit.
call	the matched call.

References

Schafer, J.L., Galagate, D.L. (2015). Causal inference with a continuous treatment and outcome: alternative estimators for parametric dose-response models. *Manuscript in preparation*.

See Also

[iptw_est](#), [ismw_est](#), [reg_est](#), [wtrg_est](#), `##` etc. for other estimates.

[t_mod](#), [overlap_fun](#) to prepare the data for use in the different estimates.

Examples

```

## Example from Schafer (2015).

example_data <- sim_data

t_mod_list <- t_mod(treat = T,
  treat_formula = T ~ B.1 + B.2 + B.3 + B.4 + B.5 + B.6 + B.7 + B.8,
  data = example_data,
  treat_mod = "Normal")

cond_exp_data <- t_mod_list$T_data
full_data <- cbind(example_data, cond_exp_data)

aipwee_list <- aipwee_est(Y = Y,
  treat = T,
  covar_formula = ~ B.1 + B.2 + B.3 + B.4 + B.5 + B.6 + B.7 + B.8,
  covar_lin_formula = ~ 1,
  covar_sq_formula = ~ 1,
  data = example_data,
  e_treat_1 = full_data$est_treat,
  e_treat_2 = full_data$est_treat_sq,
  e_treat_3 = full_data$est_treat_cube,
  e_treat_4 = full_data$est_treat_quartic,
  degree = 1,
  wt = NULL,
  method = "same",
  spline_df = NULL,
  spline_const = 1,
  spline_linear = 1,
  spline_quad = 1)

sample_index <- sample(1:1000, 100)

plot(example_data$T[sample_index],
  example_data$Y[sample_index],
  xlab = "T",
  ylab = "Y",
  main = "aipwee estimate")

abline(aipwee_list$param[1],
  aipwee_list$param[2],
  lty = 2,
  lwd = 2,
  col = "blue")

legend('bottomright',
  "aipwee estimate",
  lty = 2,
  lwd = 2,
  col = "blue",
  bty='Y',

```

```

      cex=1)
  rm(example_data, t_mod_list, cond_exp_data, full_data, aipwee_list, sample_index)

```

 bart_est

The BART estimator

Description

This function estimates the ADRF using bart.

Usage

```

bart_est(Y,
         treat,
         outcome_formula,
         data,
         grid_val,
         ...)

```

Arguments

Y	is the the name of the outcome variable contained in data.
treat	is the name of the treatment variable contained in data.
outcome_formula	is the formula used for fitting the outcome surface. gps is one of the independent variables to use in the outcome_formula. ie. $Y \sim \text{treat} + X.1 + X.2 + \dots$ or a variation of this.
data	is a dataframe containing Y, treat, and X.
grid_val	contains the treatment values to be evaluated.
...	additional arguments to be passed to the bart() outcome function.

Details

The mean DRF is estimated

Value

bart_est returns an object of class "causaldrf_simple", a list that contains the following components:

param	parameter estimates for a bart fit.
out_mod	the result of the bart fit.
call	the matched call.

References

Schafer, J.L., Galagate, D.L. (2015). Causal inference with a continuous treatment and outcome: alternative estimators for parametric dose-response models. *Manuscript in preparation*.

Hill, Jennifer L. (2011). Bayesian nonparametric modeling for causal inference. *Journal of Computational and Graphical Statistics* **20.1** (2011).

See Also

[nw_est](#), [iw_est](#), [hi_est](#), [gam_est](#), [add_spl_est](#), etc. for other estimates.

[t_mod](#), [overlap_fun](#) to prepare the data for use in the different estimates.

Examples

```
## Example from Schafer (2015). bart takes a few minutes to run (depending on computer).

example_data <- sim_data

## Not run:
# This estimate takes a long time to run...
bart_list <- bart_est(Y = Y,
  treat = T,
  outcome_formula = Y ~ T + B.1 + B.2 + B.3 + B.4 + B.5 + B.6 + B.7 + B.8,
  data = example_data,
  grid_val = seq(8, 16, by = 1))

sample_index <- sample(1:1000, 100)

plot(example_data$T[sample_index],
  example_data$Y[sample_index],
  xlab = "T",
  ylab = "Y",
  main = "bart estimate")

lines(seq(8, 16, by = 1),
  bart_list$param,
  lty = 2,
  lwd = 2,
  col = "blue")

legend('bottomright',
  "bart estimate",
  lty=2,
  lwd = 2,
  col = "blue",
  bty='Y',
  cex=1)

## End(Not run)

rm(example_data, bart_list, sample_index)
```


gam_est

*The GAM estimator***Description**

Similar to the HI method, but it has spline basis terms in the outcome model. It has the same gps fitting scheme as the HI method, but the outcome model relies on a GAM fit.

Usage

```
gam_est(Y,
        treat,
        treat_formula,
        data,
        grid_val,
        treat_mod,
        link_function,
        ...)
```

Arguments

Y	is the the name of the outcome variable contained in data.
treat	is the name of the treatment variable contained in data.
treat_formula	an object of class "formula" (or one that can be coerced to that class) that regresses treat on a linear combination of X: a symbolic description of the model to be fitted.
data	is a dataframe containing Y and treat and X.
grid_val	contains the treatment values to be evaluated.
treat_mod	a description of the error distribution to be used in the model for treatment. Options include: "Normal" for normal model, "LogNormal" for lognormal model, "Poisson" for Poisson model, "NegBinom" for negative binomial model, "Gamma" for gamma model.
link_function	is either "log", "inverse", or "identity" for the "Gamma" treat_mod.
...	additional arguments to be passed to the gam() outcome function.

Value

gam_est returns an object of class "causaldrf", a list that contains the following components:

param	parameter estimates for a gam fit.
t_mod	the result of the treatment model fit.
out_mod	the result of the outcome model fit.
call	the matched call.

References

Schafer, J.L., Galagate, D.L. (2015). Causal inference with a continuous treatment and outcome: alternative estimators for parametric dose-response models. *Manuscript in preparation*.

See Also

[nw_est](#), [iw_est](#), [hi_est](#), [gam_est](#), [add_spl_est](#), [bart_est](#), etc. for other estimates.

[t_mod](#), [overlap_fun](#) to prepare the data for use in the different estimates.

Examples

```
## Example from Schafer (2015).

example_data <- sim_data

gam_list <- gam_est(Y = Y,
  treat = T,
  treat_formula = T ~ B.1 + B.2 + B.3 + B.4 + B.5 + B.6 + B.7 + B.8,
  data = example_data,
  grid_val = seq(8, 16, by = 1),
  treat_mod = "Normal")

sample_index <- sample(1:1000, 100)

plot(example_data$T[sample_index],
  example_data$Y[sample_index],
  xlab = "T",
  ylab = "Y",
  main = "gam estimate")

lines(seq(8, 16, by = 1),
  gam_list$param,
  lty = 2,
  lwd = 2,
  col = "blue")

legend('bottomright',
  "gam estimate",
  lty=2,
  lwd = 2,
  col = "blue",
  bty='Y',
  cex=1)

rm(example_data, gam_list, sample_index)
```

get_ci	<i>This calculates an upper and lower bound from bootstrap matrix</i>
--------	---

Description

This function takes a matrix containing the bootstrapped coefficients from a parametric ADRF estimator and returns upper and lower 95 percent confidence lines.

Usage

```
get_ci(grid_val,
       coef_mat,
       degree)
```

Arguments

grid_val	is the vector of grid values on treat axis
coef_mat	contains the bootstrapped parameter estimates.
degree	is 1 for linear and 2 for quadratic outcome model

Value

get_ci returns upper and lower 95 percent confidence lines.

hi_est	<i>The Hirano and Imbens estimator</i>
--------	--

Description

This function estimates the gps function and values and estimates the mean DRF. gps score is based on different treatment models. Treatment is linearly related to Xs.

Usage

```
hi_est(Y,
      treat,
      treat_formula,
      outcome_formula,
      data,
      grid_val,
      treat_mod,
      link_function,
      ...)
```

Arguments

Y	is the the name of the outcome variable contained in data.
treat	is the name of the treatment variable contained in data.
treat_formula	an object of class "formula" (or one that can be coerced to that class) that regresses treat on a linear combination of X: a symbolic description of the model to be fitted.
outcome_formula	is the formula used for fitting the outcome surface. gps is one of the independent variables to use in the outcome_formula. ie. $Y \sim \text{treat} + I(\text{treat}^2) + \text{gps} + I(\text{gps}^2) + \text{treat} * \text{gps}$ or a variation of this. Use gps as the name of the variable representing the gps in outcome_formula.
data	is a dataframe containing Y, treat, and X.
grid_val	contains the treatment values to be evaluated.
treat_mod	a description of the error distribution to be used in the model for treatment. Options include: "Normal" for normal model, "LogNormal" for lognormal model, "Poisson" for Poisson model, "NegBinom" for negative binomial model, "Gamma" for gamma model, "Binomial" for binomial model.
link_function	For treat_mod = "Gamma" (fitted using glm) alternatives are "log" or "inverse". For treat_mod = "Binomial" (fitted using glm) alternatives are "logit", "probit", "cauchit", "log" and "cloglog".
...	additional arguments to be passed to the outcome lm() function.

Details

The ADRF is estimated

Value

hi_est returns an object of class "causaldrf", a list that contains the following components:

param	parameter estimates for a hi fit.
t_mod	the result of the treatment model fit.
out_mod	the result of the outcome model fit.
call	the matched call.

References

Schafer, J.L., Galagate, D.L. (2015). Causal inference with a continuous treatment and outcome: alternative estimators for parametric dose-response models. *Manuscript in preparation*.

See Also

[nw_est](#), [iw_est](#), [hi_est](#), [gam_est](#), [add_spl_est](#), [bart_est](#), etc. for other estimates.
[t_mod](#), [overlap_fun](#) to prepare the data for use in the different estimates.

Examples

```

## Example from Schafer (2015).

example_data <- sim_data

hi_list <- hi_est(Y = Y,
  treat = T,
  treat_formula = T ~ B.1 + B.2 + B.3 + B.4 + B.5 + B.6 + B.7 + B.8,
  outcome_formula = Y ~ T + I(T^2) + gps + I(gps^2) + T * gps,
  data = example_data,
  grid_val = seq(8, 16, by = 1),
  treat_mod = "Normal")

sample_index <- sample(1:1000, 100)

plot(example_data$T[sample_index],
  example_data$Y[sample_index],
  xlab = "T",
  ylab = "Y",
  main = "hi estimate")

lines(seq(8, 16, by = 1),
  hi_list$param,
  lty = 2,
  lwd = 2,
  col = "blue")

legend('bottomright',
  "hi estimate",
  lty=2,
  lwd = 2,
  col = "blue",
  bty='Y',
  cex=1)

rm(example_data, hi_list, sample_index)

## Example from van der Wal, Willem M., and Ronald B. Geskus. (2011)
#Simulate data with continuous confounder and outcome, binomial exposure.
#Marginal causal effect of exposure on outcome: 10.
n <- 1000
simdat <- data.frame(l = rnorm(n, 10, 5))
a.lin <- simdat$l - 10
pa <- exp(a.lin)/(1 + exp(a.lin))
simdat$a <- rbinom(n, 1, prob = pa)
simdat$y <- 10*simdat$a + 0.5*simdat$l + rnorm(n, -10, 5)
simdat[1:5,]
temp_hi <- hi_est(Y = y,
  treat = a,
  treat_formula = a ~ l,
  outcome_formula = y ~ gps,

```

```
data = simdat,  
grid_val = c(0, 1),  
treat_mod = "Binomial",  
link_function = "logit")  
  
temp_hi[[1]] # estimated coefficients
```

hi_sim_data

Simulated data from Hirano and Imbens (2004).

Description

Simulated data used in the paper "The propensity score with continuous treatments."

Usage

```
data(hi_sim_data)
```

Format

A data frame with 1000 rows and 6 variables:

Details

A dataset containing hi_sim_data.

Source

use the hi_sample function

References

Hirano, Keisuke, and Guido W. Imbens. "The propensity score with continuous treatments." *Applied Bayesian modeling and causal inference from incomplete-data perspectives* (2004): 73-84.

Moodie, Erica EM, and David A. Stephens. "Estimation of dose-response functions for longitudinal data using the generalised propensity score." *Statistical methods in medical research* **21.2** (2012): 149-166.

Examples

```
## Example from Hirano and Imbens (2004).  
data(hi_sim_data)  
head(hi_sim_data)
```

 iptw_est

The inverse probability of treatment weighting (iptw) method

Description

The iptw method or importance weighting method estimates the ADRF by weighting the data with stabilized or non-stabilized weights.

Usage

```
iptw_est(Y,
        treat,
        treat_formula,
        numerator_formula,
        data,
        degree,
        treat_mod,
        link_function,
        ...)
```

Arguments

Y	is the the name of the outcome variable contained in data.
treat	is the name of the treatment variable contained in data.
treat_formula	an object of class "formula" (or one that can be coerced to that class) that regresses treat on a linear combination of X: a symbolic description of the model to be fitted.
numerator_formula	an object of class "formula" (or one that can be coerced to that class) that regresses treat on a linear combination of X: a symbolic description of the model to be fitted. i.e. $treat \sim 1$.
data	is a dataframe containing Y, treat, and X.
degree	is 1 for linear and 2 for quadratic outcome model.
treat_mod	a description of the error distribution to be used in the model for treatment. Options include: "Normal" for normal model, "LogNormal" for lognormal model, "Poisson" for Poisson model, "NegBinom" for negative binomial model, "Gamma" for gamma model, "Binomial" for binomial model, "Ordinal" for ordinal model, "Multinomial" for multinomial model.
link_function	specifies the link function between the variables in numerator or denominator and exposure, respectively. For <code>treat_mod = "Gamma"</code> (fitted using glm) alternatives are "log" or "inverse". For <code>treat_mod = "Binomial"</code> (fitted using glm) alternatives are "logit", "probit", "cauchit", "log" and "cloglog". For <code>treat_mod = "Multinomial"</code> this argument is ignored, and multinomial logistic regression models are always used (fitted using multinom). For <code>treat_mod = "Ordinal"</code> (fitted using polr) alternatives are "logit", "probit", "cauchit", and "cloglog".

... additional arguments to be passed to the low level treatment regression fitting functions.

Value

iptw_est returns an object of class "causaldrf", a list that contains the following components:

param	parameter estimates for a iptw fit.
t_mod	the result of the treatment model fit.
num_mod	the result of the numerator model fit.
weights	the estimated weights.
weight_data	the weights.
out_mod	the outcome model.
call	the matched call.

References

Schafer, J.L., Galagate, D.L. (2015). Causal inference with a continuous treatment and outcome: alternative estimators for parametric dose-response models. *Manuscript in preparation*.

van der Wal, Willem M., and Ronald B. Geskus. "IPW: an R package for inverse probability weighting." *Journal of Statistical Software* **43.13** (2011): 1-23.

See Also

[iptw_est](#), [ismw_est](#), [reg_est](#), [aipwee_est](#), [wtrg_est](#), etc. for other estimates.
[t_mod](#), [overlap_fun](#) to prepare the data for use in the different estimates.

Examples

```
## Example from Schafer (2015).

example_data <- sim_data

iptw_list <- iptw_est(Y = Y,
  treat = T,
  treat_formula = T ~ B.1 + B.2 + B.3 + B.4 + B.5 + B.6 + B.7 + B.8,
  numerator_formula = T ~ 1,
  data = example_data,
  degree = 1,
  treat_mod = "Normal")

sample_index <- sample(1:1000, 100)

plot(example_data$T[sample_index],
  example_data$Y[sample_index],
  xlab = "T",
  ylab = "Y",
  main = "iptw estimate")
```



```

abline(iptw_list$param[1],
       iptw_list$param[2],
       lty=2,
       lwd = 2,
       col = "blue")

legend('bottomright',
       "iptw estimate",
       lty=2,
       lwd = 2,
       col = "blue",
       bty='Y',
       cex=1)

rm(example_data, iptw_list, sample_index)

## Example from van der Wal, Willem M., and Ronald B. Geskus. (2011)
## Simulate data with continuous confounder and outcome, binomial exposure.
## Marginal causal effect of exposure on outcome: 10.
n <- 1000
simdat <- data.frame(l = rnorm(n, 10, 5))
a.lin <- simdat$l - 10
pa <- exp(a.lin)/(1 + exp(a.lin))
simdat$a <- rbinom(n, 1, prob = pa)
simdat$y <- 10*simdat$a + 0.5*simdat$l + rnorm(n, -10, 5)
simdat[1:5,]
temp_iptw <- iptw_est(Y = y,
                     treat = a,
                     treat_formula = a ~ 1,
                     numerator_formula = a ~ 1,
                     data = simdat,
                     degree = 1,
                     treat_mod = "Binomial",
                     link_function = "logit")

temp_iptw[[1]] # estimated coefficients

```

Description

This method estimates the ADRF by using weighting matrices instead of scalars. The weight matrices require conditional expectations of the treatment and higher order conditional expectations. It uses outputs from the `t_mod` function.

Usage

```
ismw_est(Y,
         treat,
         data,
         e_treat_1,
         e_treat_2,
         e_treat_3,
         e_treat_4,
         degree )
```

Arguments

Y	is the the name of the outcome variable contained in data.
treat	is the name of the treatment variable contained in data.
data	is a dataframe containing Y, treat, and X.
e_treat_1	a vector, representing the conditional expectation of treat from t_mod.
e_treat_2	a vector, representing the conditional expectation of treat^2 from t_mod.
e_treat_3	a vector, representing the conditional expectation of treat^3 from t_mod.
e_treat_4	a vector, representing the conditional expectation of treat^4 from t_mod.
degree	is 1 for linear and 2 for quadratic outcome model.

Details

This function uses the outputs of the t_mod function as inputs.

Value

ismw_est returns an object of class "causaldrf_simple", a list that contains the following components:

param	the estimated parameters.
call	the matched call.

References

Schafer, J.L., Galagate, D.L. (2015). Causal inference with a continuous treatment and outcome: alternative estimators for parametric dose-response models. *Manuscript in preparation*.

See Also

[iptw_est](#), [ismw_est](#), [reg_est](#), [aipwee_est](#), [wtrg_est](#), etc. for other estimates.
[t_mod](#), [overlap_fun](#) to prepare the data for use in the different estimates.

Examples

```
## Example from Schafer (2015).

example_data <- sim_data

t_mod_list <- t_mod(treat = T,
  treat_formula = T ~ B.1 + B.2 + B.3 + B.4 + B.5 + B.6 + B.7 + B.8,
  data = example_data,
  treat_mod = "Normal")

cond_exp_data <- t_mod_list$T_data

full_data <- cbind(example_data, cond_exp_data)

ismw_list <- ismw_est(Y = Y,
  treat = T,
  data = full_data,
  e_treat_1 = full_data$est_treat,
  e_treat_2 = full_data$est_treat_sq,
  e_treat_3 = full_data$est_treat_cube,
  e_treat_4 = full_data$est_treat_quartic,
  degree = 1)

sample_index <- sample(1:1000, 100)

plot(example_data$T[sample_index],
  example_data$Y[sample_index],
  xlab = "T",
  ylab = "Y",
  main = "ismw estimate")

abline(ismw_list$param[1],
  ismw_list$param[2],
  lty=2,
  lwd = 2,
  col = "blue")

legend('bottomright',
  "ismw estimate",
  lty=2,
  lwd = 2,
  col = "blue",
  bty='Y',
  cex=1)

rm(example_data, t_mod_list, cond_exp_data, full_data, ismw_list, sample_index)

## Example from Hirano & Imbens (2004).
## Put example here....
```

iw_est	<i>The inverse weighting estimator (nonparametric method)</i>
--------	---

Description

This is a nonparametric method that estimates the CDRF by using a local linear regression of Y on $treat$ with weighted kernel function. For details, see Flores et. al. (2012).

Usage

```
iw_est(Y,
      treat,
      treat_formula,
      data,
      grid_val,
      bandw,
      treat_mod,
      link_function,
      ...)
```

Arguments

<code>Y</code>	is the the name of the outcome variable contained in <code>data</code> .
<code>treat</code>	is the name of the treatment variable contained in <code>data</code> .
<code>treat_formula</code>	an object of class "formula" (or one that can be coerced to that class) that regresses <code>treat</code> on a linear combination of X : a symbolic description of the model to be fitted.
<code>data</code>	is a dataframe containing <code>Y</code> , <code>treat</code> , and X .
<code>grid_val</code>	contains the treatment values to be evaluated.
<code>bandw</code>	is the bandwidth. Default is 1.
<code>treat_mod</code>	a description of the error distribution to be used in the model for treatment. Options include: "Normal" for normal model, "LogNormal" for lognormal model, "Poisson" for Poisson model, "NegBinom" for negative binomial model, "Gamma" for gamma model.
<code>link_function</code>	is either "log", "inverse", or "identity" for the "Gamma" <code>treat_mod</code> .
<code>...</code>	additional arguments to be passed to the treatment regression function.

Value

`iw_est` returns an object of class "causaldrf", a list that contains the following components:

<code>param</code>	parameter estimates for a iw fit.
<code>t_mod</code>	the result of the treatment model fit.
<code>call</code>	the matched call.

References

Schafer, J.L., Galagate, D.L. (2015). Causal inference with a continuous treatment and outcome: alternative estimators for parametric dose-response models. *Manuscript in preparation*.

Flores, Carlos A., et al. "Estimating the effects of length of exposure to instruction in a training program: the case of job corps." *Review of Economics and Statistics* **94.1** (2012): 153-171.

See Also

[nw_est](#), [iw_est](#), [hi_est](#), [gam_est](#), [add_spl_est](#), [bart_est](#), etc. for other estimates.

Examples

```
## Example from Schafer (2015).

example_data <- sim_data

iw_list <- iw_est(Y = Y,
                 treat = T,
                 treat_formula = T ~ B.1 + B.2 + B.3 + B.4 + B.5 + B.6 + B.7 + B.8,
                 data = example_data,
                 grid_val = seq(8, 16, by = 1),
                 bandw = bw.SJ(example_data$T),
                 treat_mod = "Normal")

sample_index <- sample(1:1000, 100)

plot(example_data$T[sample_index],
     example_data$Y[sample_index],
     xlab = "T",
     ylab = "Y",
     main = "iw estimate")

lines(seq(8, 16, by = 1),
      iw_list$param,
      lty = 2,
      lwd = 2,
      col = "blue")

legend('bottomright',
      "iw estimate",
      lty=2,
      lwd = 2,
      col = "blue",
      bty='Y',
      cex=1)

rm(example_data, iw_list, sample_index)

## Example from Imai & van Dyk (2004).

data("nmes_data")
```

```

head(nmes_data)
# look at only people with medical expenditures greater than 0
nmes_nonzero <- nmes_data[which(nmes_data$TOTALEXP > 0), ]

iw_list <- iw_est(Y = TOTALEXP,
                 treat = packyears,
                 treat_formula = packyears ~ LASTAGE + I(LASTAGE^2) +
                   AGESMOKE + I(AGESMOKE^2) + MALE + RACE3 + beltuse +
                   educate + marital + SREGION + POVSTALB,
                 data = nmes_nonzero,
                 grid_val = seq(5, 100, by = 5),
                 bandw = bw.SJ(nmes_nonzero$packyears),
                 treat_mod = "LogNormal")

set.seed(307)
sample_index <- sample(1:nrow(nmes_nonzero), 100)

plot(nmes_nonzero$packyears[sample_index],
     nmes_nonzero$TOTALEXP[sample_index],
     xlab = "packyears",
     ylab = "TOTALEXP",
     main = "iw estimate")

lines(seq(5, 100, by = 5),
      iw_list$iw_param,
      lty = 2,
      lwd = 2,
      col = "blue")

legend('topright',
      "iw estimate",
      lty=2,
      lwd = 2,
      col = "blue",
      bty='Y',
      cex = 1)
abline(0, 0)

```

nmes_data

Data set containing data from the National Medical Expenditure Survey (NMES)

Description

Data set from the NMES. with 9708 observations and 12 variables.

Usage

```
data(nmes_data)
```

Format

A dataset containing 9708 observations and 12 variables.

References

Imai, K., & van Dyk, D.A. (2004). Causal Inference With General Treatment Regimes: Generalizing the Propensity Score. *Journal of the American Statistical Association*, **99**(467).

National Center for Health Services Research and Health Care Technology Assessment. NATIONAL MEDICAL EXPENDITURE SURVEY, 1987: INSTITUTIONAL POPULATION COMPONENT. Rockville, MD: Westat, Inc. [producer], 1987. Ann Arbor, MI: Inter-university Consortium for Political and Social Research [distributor], 1990. doi:10.3886/ICPSR09280.v1

Bryer, Jason M. "TriMatch: An R Package for Propensity Score Matching of Non-binary Treatments." The R User Conference, useR! 2013 July 10-12 2013 University of Castilla-La Mancha, Albacete, Spain. Vol. 10. No. 30. 2013.

Examples

```
data(nmes_data)
head(nmes_data)
```

nw_est

The Nadaraya-Watson modified estimator

Description

This is a kernel based regression method that uses a kernel as a weighting function. The normal kernel is weighted by the inverse of the estimated gps. See Flores et al. (2012) for more details.

Usage

```
nw_est(Y,
       treat,
       treat_formula,
       data,
       grid_val,
       bandw,
       treat_mod,
       link_function,
       ...)
```

Arguments

Y	is the the name of the outcome variable contained in data.
treat	is the name of the treatment variable contained in data.
treat_formula	an object of class "formula" (or one that can be coerced to that class) that regresses treat on a linear combination of X: a symbolic description of the model to be fitted.

data	is a dataframe containing Y and treat and X.
grid_val	contains the treatment values to be evaluated.
bandw	is the bandwidth. Default is 1.
treat_mod	a description of the error distribution to be used in the model for treatment. Options include: "Normal" for normal model, "LogNormal" for lognormal model, "Poisson" for Poisson model, "NegBinom" for negative binomial model, "Gamma" for gamma model.
link_function	is either "log", "inverse", or "identity" for the "Gamma" treat_mod.
...	additional arguments to be passed to the treatment regression function.

Details

The mean DRF is estimated

Value

nw_est returns an object of class "causaldrf", a list that contains the following components:

param	parameter estimates for a nw fit.
t_mod	the result of the treatment model fit.
call	the matched call.

References

Schafer, J.L., Galagate, D.L. (2015). Causal inference with a continuous treatment and outcome: alternative estimators for parametric dose-response models. *Manuscript in preparation*.

Flores, Carlos A., et al. "Estimating the effects of length of exposure to instruction in a training program: the case of job corps." *Review of Economics and Statistics* **94.1** (2012): 153-171.

See Also

[nw_est](#), [iw_est](#), [hi_est](#), [gam_est](#), [add_spl_est](#), [bart_est](#), etc. for other estimates.

[t_mod](#), [overlap_fun](#) to prepare the data for use in the different estimates.

Examples

```
## Example from Schafer (2015).

example_data <- sim_data

nw_list <- nw_est(Y = Y,
                 treat = T,
                 treat_formula = T ~ B.1 + B.2 + B.3 + B.4 + B.5 + B.6 + B.7 + B.8,
                 data = example_data,
                 grid_val = seq(8, 16, by = 1),
                 bandw = bw.SJ(example_data$T),
                 treat_mod = "Normal")
```



```

sample_index <- sample(1:1000, 100)

plot(example_data$T[sample_index],
      example_data$Y[sample_index],
      xlab = "T",
      ylab = "Y",
      main = "nw estimate")

lines(seq(8, 16, by = 1),
      nw_list$param,
      lty = 2,
      lwd = 2,
      col = "blue")

legend('bottomright',
      "nw estimate",
      lty=2,
      lwd = 2,
      col = "blue",
      bty='Y',
      cex=1)

rm(example_data, nw_list, sample_index)

```

 overlap_fun

This function creates an overlapping dataset

Description

This function ensures that the units overlap according to the estimated gps values. The overlapping dataset depends on the number of classes `n_class` to subclassify on.

Usage

```

overlap_fun(Y,
            treat,
            treat_formula,
            data_set,
            n_class,
            treat_mod,
            link_function,
            ...)

```

Arguments

`Y` is the the name of the outcome variable contained in data.
`treat` is the name of the treatment variable contained in data.

treat_formula	an object of class "formula" (or one that can be coerced to that class) that regresses treat on a linear combination of X: a symbolic description of the model to be fitted.
data_set	is a dataframe containing Y, treat, and X.
n_class	is the number of classes to split gps into.
treat_mod	a description of the error distribution to be used in the model for treatment. Options include: "Normal" for normal model, "LogNormal" for lognormal model, "Poisson" for Poisson model, "NegBinom" for negative binomial model, "Gamma" for gamma model.
link_function	is either "log", "inverse", or "identity" for the "Gamma" treat_mod.
...	additional arguments to be passed to the treatment regression function

Value

overlap_fun returns a list containing the following elements:

overlap_dataset	dataframe containing overlapping data.
median_vec	a vector containing median values.
overlap_treat_result	the resulting treatment fit.

References

- Schafer, J.L., Galagate, D.L. (2015). Causal inference with a continuous treatment and outcome: alternative estimators for parametric dose-response models. *Manuscript in preparation*.
- Bia, Michela, et al. "A Stata package for the application of semiparametric estimators of dose response functions." *Stata Journal* **14.3** (2014): 580-604.

See Also

[iptw_est](#), [ismw_est](#), [reg_est](#), [aipwee_est](#), [wtrg_est](#), etc. for other estimates.
[t_mod](#), [overlap_fun](#) to prepare the data for use in the different estimates.

Examples

```
## Example from Schafer (2015).

example_data <- sim_data

overlap_list <- overlap_fun(Y = Y,
  treat = T,
  treat_formula = T ~ B.1 + B.2 + B.3 + B.4 + B.5 + B.6 + B.7 + B.8,
  data_set = example_data,
  n_class = 3,
  treat_mod = "Normal")

overlapped_data <- overlap_list$overlap_dataset
```

```
summary(overlapped_data)

rm(example_data, overlap_list, overlapped_data)
```

prop_spline_est *The propensity-spline prediction estimator*

Description

This method estimates the linear or quadratic parameters of the ADRF by estimating a least-squares fit on the basis functions which are composed of combinations of the covariates, propensity spline basis, and treatment values.

Usage

```
prop_spline_est(Y,
               treat,
               covar_formula = ~ 1,
               covar_lin_formula = ~ 1,
               covar_sq_formula = ~ 1,
               data,
               e_treat_1 = NULL,
               degree = 1,
               wt = NULL,
               method = "same",
               spline_df = NULL,
               spline_const = 1,
               spline_linear = 1,
               spline_quad = 1)
```

Arguments

`Y` is the the name of the outcome variable contained in `data`.

`treat` is the name of the treatment variable contained in `data`.

`covar_formula` is the formula to describe the covariates needed to estimate the constant term: $\sim X.1 + \dots$. Can include higher order terms or interactions. i.e. $\sim X.1 + I(X.1^2) + X.1 * X.2 + \dots$. Don't forget the tilde before listing the covariates.

`covar_lin_formula` is the formula to describe the covariates needed to estimate the linear term, `t`: $\sim X.1 + \dots$. Can include higher order terms or interactions. i.e. $\sim X.1 + I(X.1^2) + X.1 * X.2 + \dots$. Don't forget the tilde before listing the covariates.

`covar_sq_formula` is the formula to describe the covariates needed to estimate the quadratic term, `t2`: $\sim X.1 + \dots$. Can include higher order terms or interactions. i.e. $\sim X.1 + I(X.1^2) + X.1 * X.2 + \dots$. Don't forget the tilde before listing the covariates.

data	is a dataframe containing Y, treat, and X.
e_treat_1	a vector, representing the conditional expectation of treat from T_mod. Or, plug in gps estimates here to create splines from the gps values.
degree	is 1 for linear and 2 for quadratic outcome model.
wt	is weight used in lsfit for outcome regression. Default is wt = NULL.
method	is "same" if the same set of covariates are used to estimate the constant, linear, and/or quadratic term with no spline terms. If method = "different", then different sets of covariates can be used to estimate the constant, linear, and/or quadratic term. covar_lin_formula and covar_sq_formula must be specified if method = "different".
spline_df	degrees of freedom. The default, spline_df = NULL, corresponds to no knots.
spline_const	is the number of spline terms to include when estimating the constant term.
spline_linear	is the number of spline terms to include when estimating the linear term.
spline_quad	is the number of spline terms to include when estimating the quadratic term.

Value

prop_spline_est returns an object of class "causaldrf_lsfit", a list that contains the following components:

param	the estimated parameters.
out_mod	the result of the outcome model fit using lsfit.
call	the matched call.

References

Schafer, J.L., Galagate, D.L. (2015). Causal inference with a continuous treatment and outcome: alternative estimators for parametric dose-response models. *Manuscript in preparation*.

See Also

[iptw_est](#), [ismw_est](#), [reg_est](#), [aipwee_est](#), [wtrg_est](#), etc. for other estimates.
[t_mod](#), [overlap_fun](#) to prepare the data for use in the different estimates.

Examples

```
## Example from Schafer (2015).

example_data <- sim_data

t_mod_list <- t_mod(treat = T,
  treat_formula = T ~ B.1 + B.2 + B.3 + B.4 + B.5 + B.6 + B.7 + B.8,
  data = example_data,
  treat_mod = "Normal")

cond_exp_data <- t_mod_list$T_data
full_data <- cbind(example_data, cond_exp_data)
```

```
prop_spline_list <- prop_spline_est(Y = Y,
                                   treat = T,
                                   covar_formula = ~ B.1 + B.2 + B.3 + B.4 + B.5 + B.6 + B.7 + B.8,
                                   covar_lin_formula = ~ 1,
                                   covar_sq_formula = ~ 1,
                                   data = example_data,
                                   e_treat_1 = full_data$est_treat,
                                   degree = 1,
                                   wt = NULL,
                                   method = "different",
                                   spline_df = 5,
                                   spline_const = 4,
                                   spline_linear = 4,
                                   spline_quad = 4)

sample_index <- sample(1:1000, 100)

plot(example_data$T[sample_index],
      example_data$Y[sample_index],
      xlab = "T",
      ylab = "Y",
      main = "propensity spline estimate")

abline(prop_spline_list$param[1],
       prop_spline_list$param[2],
       lty = 2,
       col = "blue",
       lwd = 2)

legend('bottomright',
       "propensity spline estimate",
       lty = 2,
       bty = 'Y',
       cex = 1,
       col = "blue",
       lwd = 2)

rm(example_data, prop_spline_list, sample_index)
```

Description

This method estimates the linear or quadratic parameters of the ADRF by estimating a least-squares fit on the basis functions which are composed of combinations of the covariates and treatment values.

Usage

```
reg_est(Y,
        treat,
        covar_formula,
        covar_lin_formula = NULL,
        covar_sq_formula = NULL,
        data,
        degree,
        wt = NULL,
        method = "same")
```

Arguments

Y is the the name of the outcome variable contained in data.

treat is the name of the treatment variable contained in data.

covar_formula is the formula to describe the covariates needed to estimate the constant term:
 $\sim X.1 + \dots$. Can include higher order terms or interactions. i.e. $\sim X.1 + I(X.1^2) + X.1 * X.2 + \dots$.
 Don't forget the tilde before listing the covariates.

covar_lin_formula is the formula to describe the covariates needed to estimate the linear term, t:
 $\sim X.1 + \dots$. Can include higher order terms or interactions. i.e. $\sim X.1 + I(X.1^2) + X.1 * X.2 + \dots$.
 Don't forget the tilde before listing the covariates.

covar_sq_formula is the formula to describe the covariates needed to estimate the quadratic term,
 t^2 : $\sim X.1 + \dots$. Can include higher order terms or interactions. i.e.
 $\sim X.1 + I(X.1^2) + X.1 * X.2 + \dots$. Don't forget the tilde before
 listing the covariates.

data is a dataframe containing Y, treat, and X.

degree is 1 for linear and 2 for quadratic outcome model.

wt is weight used in lsfit for outcome regression. Default is wt = NULL.

method is "same" if the same set of covariates are used to estimate the constant, linear, and/or quadratic term. If method = "different", then different sets of covariates can be used to estimate the constant, linear, and/or quadratic term. covar_lin_formula and covar_sq_formula must be specified if method = "different".

Value

reg_est returns an object of class "causaldrf_lsfit", a list that contains the following components:

param the estimated parameters.

out_mod the result of the outcome model fit using lsfit.

call the matched call.

References

Schafer, J.L., Galagate, D.L. (2015). Causal inference with a continuous treatment and outcome: alternative estimators for parametric dose-response models. *Manuscript in preparation*.

See Also

[iptw_est](#), [ismw_est](#), [aipwee_est](#), [wtrg_est](#), etc. for other estimates.

[t_mod](#), [overlap_fun](#) to prepare the data for use in the different estimates.

Examples

```
## Example from Schafer (2015).

example_data <- sim_data

reg_list <- reg_est(Y = Y,
                  treat = T,
                  covar_formula = ~ B.1 + B.2 + B.3 + B.4 + B.5 + B.6 + B.7 + B.8,
                  covar_lin_formula = ~ 1,
                  covar_sq_formula = ~ 1,
                  data = example_data,
                  degree = 1,
                  wt = NULL,
                  method = "same")

sample_index <- sample(1:1000, 100)

plot(example_data$T[sample_index],
     example_data$Y[sample_index],
     xlab = "T",
     ylab = "Y",
     main = "regression estimate")

abline(reg_list$param[1],
       reg_list$param[2],
       lty = 2,
       col = "blue",
       lwd = 2)

legend('bottomright',
      "regression estimate",
      lty = 2,
      bty = 'Y',
      cex = 1,
      col = "blue",
      lwd = 2)

rm(example_data, reg_list, sample_index)
```

 scalar_wts

This function calculates scalar weights for use in other models

Description

This function calculates the scalar weights

Usage

```
scalar_wts(treat,
           treat_formula,
           numerator_formula,
           data,
           treat_mod,
           link_function,
           ...)
```

Arguments

treat	is the name of the treatment variable contained in data.
treat_formula	an object of class "formula" (or one that can be coerced to that class) that regresses treat on a linear combination of X: a symbolic description of the model to be fitted.
numerator_formula	an object of class "formula" (or one that can be coerced to that class) that regresses treat on a linear combination of X: a symbolic description of the model to be fitted. i.e. $treat \sim 1$.
data	is a dataframe containing treat, and X.
treat_mod	a description of the error distribution to be used in the model for treatment. Options include: "Normal" for normal model, "LogNormal" for lognormal model, "Poisson" for Poisson model, "NegBinom" for negative binomial model, "Gamma" for gamma model.
link_function	is either "log", "inverse", or "identity" for the "Gamma" treat_mod.
...	additional arguments to be passed to the treatment regression fitting function.

Value

scalar_wts returns an object of class "causaldrf_wts", a list that contains the following components:

param	summary of estimated weights.
t_mod	the result of the treatment model fit.
num_mod	the result of the numerator model fit.
weights	estimated weights for each unit.
call	the matched call.

References

Schafer, J.L., Galagate, D.L. (2015). Causal inference with a continuous treatment and outcome: alternative estimators for parametric dose-response models. *Manuscript in preparation*.

See Also

[iptw_est](#), [ismw_est](#), [reg_est](#), [aipwee_est](#), [wtrg_est](#), etc. for other estimates.
[t_mod](#), [overlap_fun](#) to prepare the data for use in the different estimates.

Examples

```
## Example from Schafer (2015).

example_data <- sim_data

scalar_wts_list <- scalar_wts(treat = T,
                             treat_formula = T ~ B.1 + B.2 + B.3 + B.4 + B.5 + B.6 + B.7 + B.8,
                             numerator_formula = T ~ 1,
                             data = example_data,
                             treat_mod = "Normal")

sample_index <- sample(1:1000, 100)

plot(example_data$T[sample_index],
      scalar_wts_list$weights[sample_index],
      xlab = "T",
      ylab = "weights",
      main = "scalar_wts")

rm(example_data, scalar_wts_list, sample_index)
```

sim_data

Simulated data from Schafer and Galagate (2015).

Description

Simulated data used in the paper "Causal inference with a continuous treatment and outcome: alternative estimators for parametric dose-response models".

Usage

```
data(sim_data)
```

Format

A data frame with 1000 rows and 20 variables:

Details

A dataset containing sim_data.

Value

(A.1, A.2, A.3, A.4, A.5, A.6, A.7, A.8) are the true measured covariates.

(B.1, B.2, B.3, B.4, B.5, B.6, B.7, B.8) are the transformed covariates.

T	treatment
Theta.1	unit level intercept
Theta.2	unit level slope
Y	outcome

Source

use the draw_sample function

References

Schafer, J.L., Galagate, D.L. (2015). Causal inference with a continuous treatment and outcome: alternative estimators for parametric dose-response models. *Manuscript in preparation*.

Examples

```
## Example from Schafer (2015).
data(sim_data)
head(sim_data)
```

t_mod	<i>A function to estimate conditional expected values and higher order moments</i>
-------	--

Description

This function fits a glm regression specified by the user to estimate conditional moments.

Usage

```
t_mod(treat,
      treat_formula,
      data,
      treat_mod,
      link_function,
      ...)
```

Arguments

treat	is the name of the treatment variable contained in data.
treat_formula	an object of class "formula" (or one that can be coerced to that class) that regresses treat on a linear combination of X: a symbolic description of the model to be fitted.
data	is a dataframe containing Y, treat, and X.
treat_mod	a description of the error distribution to be used in the model for treatment. Options include: "Normal" for normal model, "LogNormal" for lognormal model, "Poisson" for Poisson model, "NegBinom" for negative binomial model, "Gamma" for gamma model.
link_function	is either "log", "inverse", or "identity" for the "Gamma" treat_mod.
...	additional arguments to be passed to the low level treatment regression fitting functions.

Value

t_mod returns a list containing the following elements:

T_data	a dataframe containing estimated treatment, estimated treatment squared, estimated treatment cube, estimated treatment quartic, and estimated gps.
T_result	the result of the treatment model fit.

References

Schafer, J.L., Galagate, D.L. (2015). Causal inference with a continuous treatment and outcome: alternative estimators for parametric dose-response models. *Manuscript in preparation*.

See Also

[ismw_est](#), [reg_est](#), [wtrg_est](#), [aipwee_est](#), etc. for other estimates.
[overlap_fun](#) to prepare the data for use in the different estimates.

Examples

```
## Example from Schafer (2015).

example_data <- sim_data

t_mod_list <- t_mod(treat = T,
                   treat_formula = T ~ B.1 + B.2 + B.3 + B.4 + B.5 + B.6 + B.7 + B.8,
                   data = example_data,
                   treat_mod = "Normal")

cond_exp_data <- t_mod_list$T_data

full_data <- cbind(example_data, cond_exp_data)

rm(example_data, t_mod_list, cond_exp_data, full_data)
```

wtrg_est

The weighted regression method

Description

This method uses weight matrices to estimate parameters for the quadratic or linear fits.

Usage

```
wtrg_est(Y,
        treat,
        covar_formula,
        data,
        e_treat_1,
        e_treat_2,
        e_treat_3,
        e_treat_4,
        degree)
```

Arguments

Y	is the output
treat	is the treatment variable
covar_formula	is the formula for the covariates model of the form: $\sim X.1 + \dots$
data	will contain all the data: X, treat, and Y
e_treat_1	is estimated treatment
e_treat_2	is estimated treatment squared
e_treat_3	is estimated treatment cubed
e_treat_4	is estimated treatment to the fourth
degree	is 1 for linear fit and 2 for quadratic fit

Value

wtrg_est returns an object of class "causaldrf", a list that contains the following components:

param	the estimated parameters.
call	the matched call.

References

Schafer, J.L., Galagate, D.L. (2015). Causal inference with a continuous treatment and outcome: alternative estimators for parametric dose-response models. *Manuscript in preparation*.

See Also

[iptw_est](#), [ismw_est](#), [reg_est](#), [aipwee_est](#), [wtrg_est](#), etc. for other estimates.
[t_mod](#), [overlap_fun](#) to prepare the data for use in the different estimates.

Examples

```
## Example from Schafer (2015).

example_data <- sim_data

t_mod_list <- t_mod(treat = T,
  treat_formula = T ~ B.1 + B.2 + B.3 + B.4 + B.5 + B.6 + B.7 + B.8,
  data = example_data,
  treat_mod = "Normal")

cond_exp_data <- t_mod_list$T_data
full_data <- cbind(example_data, cond_exp_data)

wtrg_list <- wtrg_est(Y = Y,
  treat = T,
  covar_formula = ~ B.1 + B.2 + B.3 + B.4 + B.5 + B.6 + B.7 + B.8,
  data = example_data,
  e_treat_1 = full_data$est_treat,
  e_treat_2 = full_data$est_treat_sq,
  e_treat_3 = full_data$est_treat_cube,
  e_treat_4 = full_data$est_treat_quartic,
  degree = 1)

sample_index <- sample(1:1000, 100)

plot(example_data$T[sample_index],
  example_data$Y[sample_index],
  xlab = "T",
  ylab = "Y",
  main = "weighted regression estimate")

abline(wtrg_list$param[1],
  wtrg_list$param[2],
  lty = 2,
  lwd = 2,
  col = "blue")

legend('bottomright',
  "weighted regression estimate",
  lty = 2,
  lwd = 2,
  col = "blue",
  bty='Y',
  cex=1)

rm(example_data, t_mod_list, cond_exp_data, full_data, wtrg_list, sample_index)
```

Index

*Topic **datasets**

hi_sim_data, 14

nmes_data, 22

sim_data, 33

add_spl_est, 2, 8, 10, 12, 21, 24

aipwee_est, 4, 16, 18, 26, 28, 31, 33, 35, 37

bart_est, 3, 7, 10, 12, 21, 24

gam_est, 3, 8, 9, 10, 12, 21, 24

get_ci, 11

hi_est, 3, 8, 10, 11, 12, 21, 24

hi_sim_data, 14

iptw_est, 5, 15, 16, 18, 26, 28, 31, 33, 37

ismw_est, 5, 16, 17, 18, 26, 28, 31, 33, 35, 37

iw_est, 3, 8, 10, 12, 20, 21, 24

nmes_data, 22

nw_est, 3, 8, 10, 12, 21, 23, 24

overlap_fun, 3, 5, 8, 10, 12, 16, 18, 24, 25,
26, 28, 31, 33, 35, 37

prop_spline_est, 27

reg_est, 5, 16, 18, 26, 28, 29, 33, 35, 37

scalar_wts, 32

sim_data, 33

t_mod, 3, 5, 8, 10, 12, 16, 18, 24, 26, 28, 31,
33, 34, 37

wtrg_est, 5, 16, 18, 26, 28, 31, 33, 35, 36, 37