Package 'causalweight'

September 25, 2026

Title Estimation Methods for Causal Inference Based on Inverse Probability Weighting and Doubly Robust Estimation

Version 1.1.6

Maintainer Hugo Bodory <hugo.bodory@unisg.ch>

Description Various estimators of causal effects based on inverse probability weighting, doubly robust estimation, and double machine learning. Specifically, the package includes methods for estimating average treatment effects, direct and indirect effects in causal mediation analysis, and dynamic treatment effects based on different identification strategies (unconfoundedness, instruments, difference-in-differences, regression discontinuity designs).

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Depends R (>= 3.5.0), ranger

Imports mvtnorm, np, stats, utils, hdm, clubSandwich, glmnet, xgboost, kernlab, fastDummies, grf, checkmate, nnls, sandwich

Config/roxygen2/version 8.1.0

NeedsCompilation no

Author Hugo Bodory [aut, cre] (ORCID: <https://orcid.org/0000-0002-3645-1204>), Martin Huber [aut] (ORCID: <https://orcid.org/0000-0002-8590-9402>), Jannis Kueck [aut] (ORCID: <https://orcid.org/0000-0003-4367-0285>)

Repository CRAN

Date/Publication 2026-09-25 08:30:09 UTC

Contents

ATETDML 2
attrlateweight 4
c401k 6
catehetDML 8
coffeeleaflet 10
coupon 12

couponsretailer . . . 13
creditcard . . . 15
detectIV . . . 16
didcontDML . . . 19
didcontDMLpanel . . . 21
didDML . . . 24
didmedDML . . . 26
didmedDMLpanel . . . 28
didweight . . . 30
dyntreatDML . . . 31
games . . . 34
identificationDML . . . 36
india . . . 39
ivnr . . . 40
JC . . . 42
labormarket . . . 44
lateweight . . . 46
medDML . . . 48
medlateweight . . . 51
medqteDML . . . 53
medweight . . . 56
medweightcont . . . 59
paneltestDML . . . 61
RDDcovar . . . 62
rkd . . . 64
swissexper . . . 66
testmedident . . . 68
treatcontDML . . . 70
treatDML . . . 72
treatselDML . . . 74
treatweight . . . 77
ubduration . . . 79
wexpect . . . 81

Index **83**

ATETDML *ATET Estimation for Binary Treatments using Double Machine Learning*

Description

This function estimates the average treatment effect on the treated (ATET) for a binary treatment. Combines estimation based on (doubly robust) efficient score functions with double machine learning to control for confounders in a data-driven way.

Usage

```
ATETDML(
  y,
  d,
  x,
  MLmethod = "lasso",
  est = "dr",
  trim = 0.05,
  cluster = NULL,
  k = 3
)
```

Arguments

y	Outcome variable. Should not contain missing values.
d	Treatment group indicator (binary). Should not contain missing values.
x	Covariates to be controlled for. Should not contain missing values.
MLmethod	Machine learning method for estimating nuisance parameters using the SuperLearner package. Must be one of "lasso" (default), "randomforest", "xgboost", "svm", "ensemble", or "parametric".
est	Estimation method. Must be one of "dr" (default) for doubly robust, "ipw" for inverse probability weighting (not doubly robust!), or "reg" for regression (not doubly robust!).
trim	Trimming threshold (in percent) for discarding observations with propensity scores too close to one. Default is 0.05, implying that treatment propensity scores larger than 1-0.05=0.95 (such that the probability to be not treated is below 0.05) are trimmed.
cluster	Optional clustering variable for calculating cluster-robust standard errors.
k	Number of folds in k-fold cross-fitting. Default is 3.

Details

This function estimates the Average Treatment Effect on the Treated (ATET) under conditional independence, assuming that confounders jointly affecting the treatment and the outcome can be controlled for by observed covariates. Estimation is based on the (doubly robust) efficient score functions for potential outcomes in combination with double machine learning with cross-fitting, see Chernozhukov et al (2018). To this end, one part of the data is used for estimating the model parameters of the treatment and outcome equations based machine learning. The other part of the data is used for predicting the efficient score functions. The roles of the data parts are swapped (using k-fold cross-fitting) and the ATET is estimated based on averaging the predicted efficient score functions in the total sample. Besides double machine learning, the function also provides inverse probability weighting and regression adjustment methods (which are, however, not doubly robust).

Value

A list with the following components:

`ATET`: Estimate of the Average Treatment Effect on the Treated (ATET) in the post-treatment period.

`se`: Standard error of the ATET estimate.

`pval`: P-value of the ATET estimate.

`trimmed`: Number of discarded (trimmed) observations.

`treat`: Treatments of untrimmed observations.

`outcome`: Outcomes of untrimmed observations.

`pscores`: Treatment propensity scores of untrimmed observations.

`outcomepred`: Conditional outcome predictions under nontreatment of untrimmed observations.

References

Chernozhukov, V., Chetverikov, D., Demirer, M., Duflo, E., Hansen, C., Newey, W., Robins, J. (2018): "Double/debiased machine learning for treatment and structural parameters", The Econometrics Journal, 21, C1-C68.

Examples

```
## Not run:
n=2000                              # sample size
p=50                                # number of covariates
s=2                                 # number of covariates that are confounders
x=matrix(rnorm(n*p),ncol=p)         # covariate matrix
beta=c(rep(0.25,s), rep(0,p-s))     # coefficients determining degree of confounding
d=(x%*%beta+rnorm(n)>0)*1           # treatment equation
y=x%*%beta+0.5*d+rnorm(n)           # outcome equation
# The true ATE is equal to 0.5
output=ATETDML(y,d,x)
cat("ATET: ",round(c(output$ATET),3),", standard error: ",
    round(c(output$se),3), ", p-value: ",round(c(output$pval),3))
output$ntrimmed

## End(Not run)
```

---

attrlateweight *Local average treatment effect estimation in multiple follow-up periods with outcome attrition based on inverse probability weighting*

Description

Instrumental variable-based evaluation of local average treatment effects using weighting by the inverse of the instrument propensity score.

Usage

```
attrlateweight(
  y1,
  y2,
  s1,
  s2,
  d,
  z,
  x0,
  x1,
  weightmax = 0.1,
  boot = 1999,
  cluster = NULL
)
```

Arguments

y1	Outcome variable in the first outcome period.
y2	Outcome variable in the second outcome period.
s1	Selection indicator for first outcome period. Must be one if y1 is observed (non-missing) and zero if y1 is not observed (missing).
s2	Selection indicator for second outcome period. Must be one if y1 is observed (non-missing) and zero if y1 is not observed (missing).
d	Treatment, must be binary (either 1 or 0), must not contain missings.
z	Instrument for the endogenous treatment, must be binary (either 1 or 0), must not contain missings.
x0	Baseline (pre-instrument) confounders of the instrument and outcome, must not contain missings.
x1	Confounders in outcome period 1 (may include outcomes of period 1 y1)
weightmax	Trimming rule based on the maximum relative weight a single observation may obtain in estimation - observations with higher weights are discarded. Default is 0.1 (no observation can be assigned more than 10 percent of weights)
boot	Number of bootstrap replications for estimating standard errors. Default is 1999.
cluster	A cluster ID for block or cluster bootstrapping when units are clustered rather than iid. Must be numerical. Default is NULL (standard bootstrap without clustering).

Details

Estimation of local average treatment effects of a binary endogenous treatment on outcomes in two follow up periods that are prone to attrition. Treatment endogeneity is tackled by a binary instrument that is assumed to be conditionally valid given observed baseline confounders x0. Outcome attrition is tackled by either assuming that it is missing at random (MAR), i.e. selection w.r.t. observed variables d, z, x0, x1 (in the case of y2), and s1 (in the case of y2); or by assuming latent ignorability (LI), i.e. selection w.r.t. the treatment compliance type as well as z, x0, x1 (in the case of y2),

and s1 (in the case of y2). Units are weighted by the inverse of their conditional instrument and selection propensities, which are estimated by probit regression. Standard errors are obtained by bootstrapping the effect.

Value

An attrlateweight object contains one component results:

results: a 4X4 matrix containing the effect estimates in the first row ("effects"), standard errors in the second row ("se"), p-values in the third row ("p-value"), and the number of trimmed observations due to too large weights in the fourth row ("trimmed obs"). The first column provides the local average treatment effect (LATE) on y1 among compliers under missingness at random (MAR). The second column provides the local average treatment effect (LATE) on y2 under missingness at random (MAR). The third column provides the local average treatment effect (LATE) on y1 under latent ignorability (LI). The forth column provides the local average treatment effect (LATE) on y2 under latent ignorability (LI).

References

Frölich, M., Huber, M. (2014): "Treatment Evaluation With Multiple Outcome Periods Under Endogeneity and Attrition", Journal of the American Statistical Association, 109, 1697-1711.

Examples

```
# A little example with simulated data (4000 observations)
## Not run:
n=4000
e=(rmvnorm(n,rep(0,3), matrix(c(1,0.3,0.3,  0.3,1,0.3,  0.3,0.3,1),3,3) ))
x0=runif(n,0,1)
z=(0.25*x0+rnorm(n)>0)*1
d=(1.2*z-0.25*x0+e[,1]>0.5)*1
y1_star=0.5*x0+0.5*d+e[,2]
s1=(0.25*x0+0.25*d+rnorm(n)>-0.5)*1
y1=s1*y1_star
x1=(0.5*x0+0.5*rnorm(n))
y2_star=0.5*x0+x1+d+e[,3]
s2=s1*((0.25*x0+0.25*x1+0.25*d+rnorm(n)>-0.5)*1)
y2=s2*y2_star
# The true LATEs on y1 and y2 are equal to 0.5 and 1, respectively.
output=attrlateweight(y1=y1,y2=y2,s1=s1,s2=s2,d=d,z=z,x0=x0,x1=x1,boot=19)
round(output$results,3)
## End(Not run)
```

---

c401k *401(k) Pension Plan Participation and Net Financial Assets*

Description

Cross-sectional data with 9,275 observations including 11 variables on eligibility for and participation in 401(k) pension plans, along with income and demographic information.

Usage

```
data(c401k)
```

Format

A data frame with 9275 rows and 11 variables:

pira participation in IRA, participation = 1

nettfa net family financial assets in $1000

p401k participation in 401(k), participation = 1

e401k eligibility for 401(k), eligible = 1

inc income

incsq income square

marr marital status, married = 1

male sex, male = 1

age age

agesq age square

fsize family size

Details

An exemplary dataset to illustrate the usage of instrumental variable methods. The data include both a binary outcome (`pira`) and a continuous outcome (`nettfa`). The treatment is participation in 401(k), `p401k`. Eligibility for 401(k), `e401k`, is used as an instrument for `p401k`.

Source

The data were originally distributed as `c401k` in the **LARF** package (which has since been archived and is no longer maintained on CRAN) and are included here for continued availability. They derive from Wooldridge's data sets (Wooldridge 2010), originally entitled "401ksubs.dta" in Stata format, available at https://www.stata.com/texts/eacsap/.

References

Wooldridge, Jeffrey M. (2010). *Econometric Analysis of Cross Section and Panel Data*. 2nd Edition. MIT Press.

`catehetDML` *Testing effect homogeneity across studies with double machine learning*

Description

Testing effect homogeneity across studies with double machine learning

Usage

```
catehetDML(
  y,
  d,
  x,
  z,
  MLmethod = "lasso",
  k = 5,
  L = 4,
  trim = 0.05,
  normalized = TRUE,
  seed = 123
)
```

Arguments

`y`	Dependent variable, must not contain missings.
`d`	Treatment variable, must be binary (0/1), must not contain missings.
`x`	Covariates, must not contain missings.
`z`	Study indicator across which effect homogeneity is tested, must not contain missings. If `z` is a factor, a character vector, or a numeric vector taking only integer values, each distinct value defines one study and contrasts are formed for all but the last of them. If `z` is numeric and not integer-valued, it is discretised into `L` bins of equal size, each of which is then treated as one study.
`MLmethod`	Machine learning method for estimating the nuisance parameters based on the `SuperLearner` package. Must be either `"lasso"` (default) for lasso estimation, `"randomforest"` for random forests, `"xgboost"` for xg boosting, `"svm"` for support vector machines, `"ensemble"` for using an ensemble algorithm based on all previously mentioned machine learners, or `"parametric"` for linear or logit regression.
`k`	Number of folds in k-fold cross-fitting. Default is 5.
`L`	Number of bins into which `z` is discretised if `z` is numeric and not integer-valued. Ignored otherwise, in which case the number of studies is determined by the distinct values of `z`. Default is 4.

trim	Trimming rule for discarding observations whose estimated joint probabilities of receiving a treatment state and belonging to a study given x are smaller than trim or larger than 1-trim (to avoid too small denominators in weighting by the inverse of these probabilities). Default is 0.05.
normalized	If set to TRUE, then the inverse probability-based weights are normalized such that they average to one within each combination of treatment state and study. Default is TRUE.
seed	Default is 123.

Details

Tests whether the conditional average treatment effect (CATE) of a binary treatment d on an outcome y given covariates x is homogeneous across the studies indexed by z. The null hypothesis is that for each study, the CATE within that study coincides with the CATE in the remaining studies at any value of x. If all studies are randomized experiments, a rejection indicates that treatment effects vary with characteristics that differ across studies but are not contained in x, which speaks to the external validity of the experimental effects. If some studies are experimental while others are observational, a rejection may in addition reflect unobserved confounding in the observational studies, such that differences between the estimands may arise from unobserved confounding, effect heterogeneity, or both. See Armendariz and Huber (2026). Estimation is based on a Neyman-orthogonal score that is quadratic in the doubly robust double difference of conditional mean outcomes across treatment states and studies, in the spirit of Apfel, Hatamyar, Huber and Kueck (2024), combined with k-fold cross-fitting. The nuisance parameters, namely the conditional mean outcomes $E[Y|D = d, X, Z = z]$ and the joint probabilities $Pr(D = d, Z = z|X)$, are estimated by the machine learner selected in MLmethod. The test statistic is asymptotically normal under the null hypothesis.

Value

A list with the following elements:

teststat: estimate of the target parameter, which is zero under the null hypothesis of effect homogeneity across studies.

se: standard error of teststat.

pval: p-value of the two-sided test of the null hypothesis, based on the t-statistic teststat/se.

n_eff: effective sample size after trimming and inverse probability weighting.

n_eff_z: effective sample size for each study-specific contrast.

ntrimmed: number of observations discarded by trimming.

studies: labels of the studies for which contrasts are formed.

psi: individual contributions to the score function.

w_full: inverse probability-based weights, zero for trimmed observations.

References

Armendariz, A., Huber, M. (2026): Testing Effect Homogeneity and Confounding in High-Dimensional Experimental and Observational Studies. arXiv:2602.19703.

Apfel, N., Hatamyar, J., Huber, M., Kueck, J. (2024): Learning control variables and instruments for causal analysis in observational data. arXiv:2407.04448.

Examples

```
## Not run:
# Two studies with homogeneous CATEs, such that the test should not reject.
set.seed(123)
n <- 2000
x <- matrix(rnorm(n * 5), ncol = 5)
z <- sample(1:2, n, replace = TRUE)                    # two studies
d <- rbinom(n, 1, plogis(0.5 * x[, 1] + 0.5 * (z == 2)))
y <- d + x[, 2] + rnorm(n)                             # CATE is 1 in both studies
output <- catehetDML(y = y, d = d, x = x, z = z)
output$teststat
output$pval

# Same design, but with CATEs differing across studies, such that the test should reject.
y <- d + 1 * (z == 2) * d + x[, 2] + rnorm(n)          # CATE is 1 vs. 2
output <- catehetDML(y = y, d = d, x = x, z = z)
output$teststat
output$pval

## End(Not run)
```

`coffeeleaflet` *Information leaflet on coffee production and environmental awareness of high school / university students in Bulgaria*

Description

A dataset on the impact of an information leaflet about coffee production on students' awareness about environmental issues collected at Bulgarian highschools and universities in the year 2015.

Usage

```
coffeeleaflet
```

Format

A data frame with 522 rows and 48 variables:

grade school grade

sex 1=male, 0=female

age age in years

mob month of birth

bulgnationality dummy for Bulgarian nationality

langbulg dummy for Bulgarian mother tongue

mumage mother's age in years

mumedu mother's education (1=lower secondary or less, 2=upper secondary, 3=higher)

mumprof mother's profession (1=manager, 2=specialist, 3=worker, 4=self-employed,5=not working,6=retired,7=other)

dadage father's age in years

dadedu father's education (1=lower secondary or less, 2=upper secondary, 3=higher)

dadprof father's profession (1=manager, 2=specialist, 3=worker, 4=self-employed,5=not working,6=retired,7=other)

material material situation of the family (1=very bad,...,5=very good)

withbothpar dummy for living with both parents

withmum dummy for living with mother only

withdad dummy for living with father only

withneither dummy for living with neither mother nor father

oldsiblings number of older siblings

youngsiblings numer of younger siblings

schoolmaths school dummy (for highschool with maths specialization)

schoolrakdelsve school dummy

schoolvazov school dummy

schoolfinance school dummy

schoolvarna school dummy (for highschool in city of Varna)

schoolspanish school dummy (for Spanish highschool)

schooltechuni school dummy (for technical university)

schoolvidin school dummy (for highschool in city of Vidin)

schooluni school dummy (for university)

citysofia dummy for the capital city of Sofia

cityvarna dummy for the city of Varna

cityvidin dummy for the city of Vidin

treatment treatment (1=leaflet on environmental impact of coffee growing, 0=control group)

drinkcoffee drinks coffee (1=never, 2=not more than 1 time per week,3=several times per week, 4=1 time per day, 5=several times per day)

cupsest outcome: guess how many cups of coffee per capita are consumed in Bulgaria per year

devi_cupsest outcome: deviation of guess from true coffee consumption per capita and year in Bulgaria

impworldecon outcome: assess the importance of coffee for world economy (1=not at all important,..., 5=very important)

impincome assess the importance of coffee as a source of income for people in Africa and Latin America (1=not at all important,..., 5=very important)

awarewaste outcome: awareness of waste production due to coffee production (1=not aware,..., 5=fully aware)

awarepesticide outcome: awareness of pesticide use due to coffee production (1=not aware,..., 5=fully aware)

awaredeforestation outcome: awareness of deforestation due to coffee production (1=not aware,..., 5=fully aware)

awarewastewater outcome: awareness of waste water due to coffee production (1=not aware,..., 5=fully aware)

awarebiodiversityloss outcome: awareness of biodiversity loss due to coffee production (1=not aware,..., 5=fully aware)

awareunfairworking outcome: awareness of unfair working conditions due to coffee production (1=not aware,..., 5=fully aware)

reusepurposeful outcome: can coffee waste be reused purposefully (1=no, 2=maybe, 3=yes)

reusesoil outcome: can coffee waste be reused as soil (1=no, 2=maybe, 3=yes)

choiceprice importance of price when buying coffee (1=not important at all,..., 5=very important, 6=I don't drink coffee)

choicetastepleasure importance of pleasure or taste when buying coffee (1=not important at all,..., 5=very important, 6=I don't drink coffee)

choiceenvironsocial importance of environmental or social impact when buying coffee (1=not important at all,..., 5=very important, 6=I don't drink coffee)

References

Faldzhiyskiy, S. (Ecosystem Europe, Bulgaria) and Huber, M. (University of Fribourg): "The impact of an information leaflet about coffee production on students' awareness about environmental issues".

Examples

```
## Not run:
data(coffeeleaflet)
attach(coffeeleaflet)
data=na.omit(cbind(awarewaste,treatment,grade,sex,age))
# effect of information leaflet (treatment) on awareness of waste production
treatweight(y=data[,1],d=data[,2],x=data[,3:5],boot=199)
## End(Not run)
```

---

coupon *Data on daily spending and coupon receipt (selective subsample) This data set is a selective subsample of the data set "couponsretailer" which was constructed for illustrative purposes.*

Description

Data on daily spending and coupon receipt (selective subsample) This data set is a selective subsample of the data set "couponsretailer" which was constructed for illustrative purposes.

Usage

```
coupon
```

Format

A data frame with 1293 rows and 9 variables:

dailyspending outcome: customer's daily spending at the retailer in a specific period

coupons treatment: 1 = customer received at least one coupon in that period; 0 = customer did not receive any coupon

coupons_preperiod coupon reception in previous period: 1 = customer received at least one coupon; 0 = customer did not receive any coupon

dailyspending_preperiod daily spending at the retailer in previous period

income_bracket income group: 1 = lowest to 12 = highest

age_range age of customer: 1 = 18-25; 2 = 26-35; 3 = 36-45; 4 = 46-55; 5 = 56-70; 6 = 71 plus

married marital status: 1 = married; 0 = unmarried

rented dwelling type: 1 = rented; 0 = owned

family_size number of family members: 1 = 1; 2 = 2; 3 = 3; 4 = 4; 5 = 5 plus

---

couponsretailer *Data on daily spending and coupon receipt A dataset containing information on the purchasing behavior of 1582 retail store customers across 32 coupon campaigns.*

Description

Data on daily spending and coupon receipt A dataset containing information on the purchasing behavior of 1582 retail store customers across 32 coupon campaigns.

Usage

```
couponsretailer
```

Format

A data frame with 50,624 rows and 27 variables:

customer_id customer identifier

period period of observation: 1 = 1st period to 32 = last period

age_range age of customer: 1 = 18-25; 2 = 26-35; 3 = 36-45; 4 = 46-55; 5 = 56-70; 6 = 71 plus

married marital status: 1 = married; 0 = unmarried

rented dwelling type: 1 = rented; 0 = owned

family_size number of family members: 1 = 1; 2 = 2; 3 = 3; 4 = 4; 5 = 5 plus

income_bracket income group: 1 = lowest to 12 = highest

dailyspending_preperiod customer's daily spending at the retailer in previous period

purchase_ReadyEatFood_preperiod purchases of ready-to-eat food in previous period: 1 = yes, 0 = no

purchase_MeatSeafood_preperiod purchases of meat and seafood products in previous period: 1 = yes, 0 = no

purchase_OtherFood_preperiod purchases of other food products in previous period: 1 = yes, 0 = no

purchase_Drugstore_preperiod purchases of drugstore products in previous period: 1 = yes, 0 = no

purchase_OtherNonfood_preperiod purchases of other non-food products in previous period: 1 = yes, 0 = no

coupons_Any_preperiod coupon reception in previous period: 1 = customer received at least one coupon; 0 = customer did not receive any coupon

coupons_ReadyEatFood_preperiod coupon reception in previous period: 1 = customer received at least one ready-to-eat food coupon; 0 = customer did not receive any ready-to-eat food coupon

coupons_MeatSeafood_preperiod coupon reception in previous period: 1 = customer received at least one meat/seafood coupon; 0 = customer did not receive any meat/seafood coupon

coupons_OtherFood_preperiod coupon reception in previous period: 1 = customer received at least one coupon applicable to other food items; 0 = customer did not receive any coupon applicable to other food items

coupons_Drugstore_preperiod coupon reception in previous period: 1 = customer received at least one drugstore coupon; 0 = customer did not receive any drugstore coupon

coupons_OtherNonfood_preperiod coupon reception in previous period: 1 = customer received at least one coupon applicable to other non-food items; 0 = customer did not receive any coupon applicable to other non-food items

coupons_Any_redeemed_preperiod coupon redemption in previous period: 1 = customer redeemed at least one coupon; 0 = customer did not redeem any coupon

coupons_Any treatment: 1 = customer received at least one coupon in current period; 0 = customer did not receive any coupon

coupons_ReadyEatFood treatment: 1 = customer received at least one ready-to-eat food coupon; 0 = customer did not receive any ready-to-eat food coupon

coupons_MeatSeafood treatment: 1 = customer received at least one meat/seafood coupon; 0 = customer did not receive any meat/seafood coupon

coupons_OtherFood treatment: 1 = customer received at least one coupon applicable to other food items; 0 = customer did not receive any coupon applicable to other food items

coupons_Drugstore treatment: 1 = customer received at least one drugstore coupon; 0 = customer did not receive any drugstore coupon

coupons_OtherNonfood treatment: 1 = customer received at least one coupon applicable to other non-food items; 0 = customer did not receive any coupon applicable to other non-food items

dailyspending outcome: customer's daily spending at the retailer in current period

References

Langen, Henrika, and Huber, Martin (2023): "How causal machine learning can leverage marketing strategies: Assessing and improving the performance of a coupon campaign." PLoS ONE, 18 (1): e0278937.

`creditcard` *Expenditure and Default Data*

Description

Cross-section data on the credit history for a sample of applicants for a type of credit card.

Usage

```
creditcard
```

Format

A data frame with 1319 rows and 10 variables:

expenditure average monthly credit card expenditure

reports number of major derogatory reports

age age in years plus twelfths of a year

income yearly income (in USD 10,000)

owner 1 if owns their home, 0 if rent

selfempl 1 if self employed, 0 if not

dependents number of dependents

months months living at current address

majorcards number of major credit cards held

active number of active credit accounts

Source

Online complements to Greene (2003). Table F21.4. https://pages.stern.nyu.edu/~wgreene/Text/tables/tablelist5.htm

References

Greene, W.H. (2003). *Econometric Analysis*, 5th edition. Upper Saddle River, NJ: Prentice Hall.

detectIV *Detection of instruments and control variables and validity testing with double machine learning*

Description

Tests the validity of a pre-specified instrument and pre-specified control variables (mode 1) or learns the partition of instruments and control variables from the data (mode 2), following the two-step procedure of Apfel, Hatamyar, Huber, and Kueck (2025).

Mode 1 — user-specified x **and** z**:** The user provides the covariates x and a single candidate instrument z directly. The function skips instrument selection and runs only the validity test: it tests whether z is conditionally independent of y given d and x.

Mode 2 — data-driven selection via q**:** The user provides a matrix q of candidate variables without specifying which are instruments and which are controls. The function runs the full two-stage procedure: (1) a first-stage DML relevance test selects candidate instruments associated with d, and (2) the validity test retains those that are conditionally independent of y.

Usage

```
detectIV(
  y,
  d,
  z = NULL,
  x = NULL,
  q = NULL,
  alpha = 0.1/log(length(y)),
  critval = 0.3,
  MLmethod = "lasso",
  k = 3,
  trim = 0.01,
  L = 4,
  seed = 123
)
```

Arguments

y	Outcome variable, numeric vector. Must not contain missings.
d	Treatment variable, numeric vector. Must not contain missings.
z	**Mode 1 only.** Candidate instrument, numeric vector. Must not contain missings. Provide together with x; leave NULL (default) to use mode 2 with q.
x	**Mode 1 only.** Covariate matrix. Must not contain missings. Provide together with z; leave NULL (default) to use mode 2 with q.
q	**Mode 2 only.** Matrix of candidate variables whose columns are iteratively considered as potential instruments (with the remaining columns serving as control variables). Must not contain missings and must have at least two columns. Leave NULL (default) to use mode 1 with x and z.

alpha	**Mode 2 only.** Significance level for the first-stage relevance test. A candidate passes if its DML p-value is below alpha. The paper recommends 0.1 / log(n). Default is 0.1 / log(length(y)).
critval	Significance level for the exclusion-restriction test. Candidates whose p-value *exceeds* critval are classified as valid IVs (null of conditional independence not rejected). Note that a higher value is a more conservative choice; Apfel et al. recommend 0.3. Default is 0.3.
MLmethod	Machine learning method for estimating nuisance parameters via the SuperLearner package. Must be one of "lasso" (default), "randomforest", "xgboost", "svm", "ensemble", or "parametric".
k	Number of folds in k-fold cross-fitting. Default is 3.
trim	Trimming threshold for propensity scores: observations whose estimated propensity score falls below trim or above 1 - trim are discarded. Default is 0.01.
L	Number of partition bins for non-binary instruments with 15 or more unique values. Variables with fewer than 15 unique values use their natural categories. Default is 4.
seed	Random seed. Default is 123.

Details

Mode 1 runs the validity test of Apfel et al. (2025) directly on the user-supplied z and x, testing $H_0 : E[Y \mid D, X] = E[Y \mid D, X, Z]$. This is appropriate when the researcher has a single prespecified candidate IV and wants to assess its validity (and that of the control variables) without selection.

Mode 2 implements the full data-driven algorithm of Apfel et al. (2025). For a matrix Q of p candidate variables it constructs partitions $P_j = \{Z = Q_j,\ X = Q_{[j]}\}$ and:

1. Fits a DML partially linear regression (PLR) of d on each column Q_j controlling for $Q_{[j]}$, retaining columns whose p-value is below alpha as first-stage candidates $\hat{S}$.
2. Applies the validity test to each candidate in $\hat{S}$, classifying those with p-value above critval as valid IVs $\hat{V}$.

The detected set is $\hat{P}_{\text{pass}} = \hat{S} \cap \hat{V}$.

In both modes the validity test uses the doubly robust score of Apfel et al. (2025): eq. (7) for binary instruments and the multi-partition score of eq. (18) / Appendix C for non-binary instruments.

Value

Mode 1 returns a list with:

teststat Test statistic (should be near zero under the null).

se Standard error of the test statistic.

pval P-value of the exclusion-restriction test.

n_eff Number of observations retained after trimming.

critval Significance level used for the exclusion test.

Mode 2 returns a list with:

`valid_IVs` Integer vector of column indices of q detected as valid IVs. Empty if none found.

`firststage_pass` Column indices passing the first-stage test ($\hat{S}$).

`firststage_pvals` Named numeric vector of first-stage p-values for all columns of q.

`exclusion_pvals` Named numeric vector of exclusion-restriction p-values. NA for columns not in $\hat{S}$.

`exclusion_results` Named list of full test output objects for each first-stage candidate.

`alpha` Significance level used for the first-stage test.

`critval` Significance level used for the exclusion test.

References

Apfel, N., Hatamyar, J., Huber, M., Kueck, J. (2025): "Learning control variables and instruments for causal analysis in observational data," arXiv:2407.04448.

Huber, M., Kloiber, K., Laffers, L. (2026): "Testing Full Mediation of Treatment Effects and the Identifiability of Causal Mechanisms," arXiv:2603.04109.

Examples

```
## Not run:
set.seed(42)
n <- 2000; p <- 10
Sigma <- outer(1:p, 1:p, function(i, j) 0.5^abs(i - j))
Q_raw <- mvtnorm::rmvnorm(n, rep(0, p), Sigma)
pis   <- 1 / (1 + exp(-2 * Q_raw))
Q     <- matrix(rbinom(n * p, 1, pis), nrow = n)
beta  <- c(0.8 / (1:4), rep(0, 5))
W <- rnorm(n); U <- rnorm(n); V <- rnorm(n)
D <- as.numeric(Q[, 1:9] %*% beta + Q[, p] + V > 0)
Y <- D + Q[, 1:9] %*% beta + W + U   # Q[, p] has no direct effect on Y

# Mode 1: test a pre-specified instrument (last column) directly
res1 <- detectIV(y = Y, d = D, z = Q[, p], x = Q[, 1:9])
cat("Mode 1 p-value:", round(res1$pval, 3), "\n")

# Mode 2: learn instruments and controls from Q
res2 <- detectIV(y = Y, d = D, q = Q, alpha = 0.1 / log(n))
cat("Mode 2 column indices of detected valid IVs:", res2$valid_IVs, "\n")

## End(Not run)
```

didcontDML *Continuous Difference-in-Differences using Double Machine Learning for Repeated Cross-Sections*

Description

This function estimates the average treatment effect on the treated of a continuously distributed treatment in repeated cross-sections based on a Difference-in-Differences (DiD) approach using double machine learning to control for time-varying confounders in a data-driven manner. It supports estimation under various machine learning methods and uses k-fold cross-fitting.

Usage

```
didcontDML(
  y,
  d,
  t,
  dtreat,
  dcontrol,
  t0 = 0,
  t1 = 1,
  controls,
  MLmethod = "lasso",
  psmethod = 1,
  trim = 0.1,
  lognorm = FALSE,
  bw = NULL,
  bwfactor = 0.7,
  cluster = NULL,
  k = 3
)
```

Arguments

y	Outcome variable. Should not contain missing values.
d	Treatment variable in the treatment period of interest. Should be continuous and not contain missing values.
t	Time variable indicating outcome periods. Should not contain missing values.
dtreat	Value of the treatment under treatment (in the treatment period of interest). This value would be 1 for binary treatments.
dcontrol	Value of the treatment under control (in the treatment period of interest). This value would be 0 for binary treatments.
t0	Value indicating the pre-treatment outcome period. Default is 0.
t1	Value indicating the post-treatment outcome period in which the effect is evaluated. Default is 1.

controls	Covariates and/or previous treatment history to be controlled for. Should not contain missing values.
MLmethod	Machine learning method for estimating nuisance parameters using the SuperLearner package. Must be one of "lasso" (default), "randomforest", "xgboost", "svm", "ensemble", or "parametric".
psmethod	Method for computing generalized propensity scores. Set to 1 for estimating conditional treatment densities using the treatment as dependent variable (imposing the parametric assumption of a Gaussian conditional treatment distribution), or 2 for using the treatment kernel weights as dependent variable (nonparametric). Default is 1.
trim	Trimming threshold (in percent) for discarding observations with too much influence within any subgroup defined by the treatment group and time. Default is 0.1.
lognorm	Logical indicating if log-normal transformation should be applied when estimating conditional treatment densities using the treatment as dependent variable (parametric). Default is FALSE.
bw	Bandwidth for kernel density estimation. Default is NULL, implying that the bandwidth is calculated based on the rule-of-thumb.
bwfactor	Factor by which the bandwidth is multiplied. Default is 0.7 (undersmoothing).
cluster	Optional clustering variable for calculating standard errors.
k	Number of folds in k-fold cross-fitting. Default is 3.

Details

This function estimates the Average Treatment Effect on the Treated (ATET) by Difference-in-Differences in repeated cross-sections while controlling for confounders using double machine learning. The function supports different machine learning methods for estimating nuisance parameters and performs k-fold cross-fitting to improve estimation accuracy. The function also handles binary and continuous outcomes, and provides options for trimming and bandwidth adjustments in kernel density estimation.

Value

A list with the following components:

ATET: Estimate of the Average Treatment Effect on the Treated.

se: Standard error of the ATET estimate.

trimmed: Number of discarded (trimmed) observations.

pval: P-value.

pscores: Propensity scores of untrimmed observations (4 columns): under treatment in period t1, under treatment in period t0, under control in period t1, under control in period t0.

outcomes: Conditional outcomes of untrimmed observations (3 columns): in treatment group in period t0, in control group in period t1, in control group in period t0.

treat: Treatment status of untrimmed observations.

time: Time period of untrimmed observations.

References

Chernozhukov, V., Chetverikov, D., Demirer, M., Duflo, E., Hansen, C., Newey, W., Robins, J. (2018): "Double/debiased machine learning for treatment and structural parameters", The Econometrics Journal, 21, C1-C68.

Haddad, M., Huber, M., Medina-Reyes, J., Zhang, L. (2024): "Difference-in-Differences with Time-varying Continuous Treatments using Double/Debiased Machine Learning", working paper, University of Fribourg.

Examples

```
## Not run:
# Example with simulated data
n=2000
t=rep(c(0, 1), each=n/2)
x=0.5*rnorm(n)
u=runif(n,0,2)
d=x+u+rnorm(n)
y=(2*d+x)*t+u+rnorm(n)
# true effect is 2
results=didcontDML(y=y, d=d, t=t, dtreat=1, dcontrol=0, controls=x, MLmethod="lasso")
cat("ATET: ", round(results$ATET, 3), ", Standard error: ", round(results$se, 3))

## End(Not run)
```

---

didcontDMLpanel *Continuous Difference-in-Differences using Double Machine Learning for Panel Data*

Description

This function estimates the average treatment effect on the treated of a continuously distributed treatment in panel data based on a Difference-in-Differences (DiD) approach using double machine learning to control for time-varying confounders in a data-driven manner. It supports estimation under various machine learning methods and uses k-fold cross-fitting.

Usage

```
didcontDMLpanel(
  ydiff,
  d,
  t,
  dtreat,
  dcontrol,
  t1 = 1,
  controls,
  MLmethod = "lasso",
  psmethod = 1,
```

```
  trim = 0.1,
  lognorm = FALSE,
  bw = NULL,
  bwfactor = 0.7,
  cluster = NULL,
  k = 3
)
```

Arguments

ydiff	Outcome difference between two pre- and post-treatment periods. Should not contain missing values.
d	Treatment variable in the treatment period of interest. Should be continuous and not contain missing values.
t	Time variable indicating outcome periods. Should not contain missing values.
dtreat	Value of the treatment under treatment (in the treatment period of interest). This value would be 1 for binary treatments.
dcontrol	Value of the treatment under control (in the treatment period of interest). This value would be 0 for binary treatments.
t1	Value indicating the post-treatment outcome period in which the effect is evaluated, which is the later of the two periods used to generate the outcome difference in ydiff. For instance, if the pre-treatment outcome is measured in period 0 and the post-treatment outcome is measured in period 1 to generate ydiff, then t1 is equal to 1. Default is 1.
controls	Covariates and/or previous treatment history to be controlled for. Should not contain missing values.
MLmethod	Machine learning method for estimating nuisance parameters using the SuperLearner package. Must be one of "lasso" (default), "randomforest", "xgboost", "svm", "ensemble", or "parametric".
psmethod	Method for computing generalized propensity scores. Set to 1 for estimating conditional treatment densities using the treatment as dependent variable (imposing the parametric assumption of a Gaussian conditional treatment distribution), or 2 for using the treatment kernel weights as dependent variable (nonparametric). Default is 1.
trim	Trimming threshold (in percent) for discarding observations with too much influence within any subgroup defined by the treatment group and time. Default is 0.1.
lognorm	Logical indicating if log-normal transformation should be applied when estimating conditional treatment densities using the treatment as dependent variable (parametric). Default is FALSE.
bw	Bandwidth for kernel density estimation. Default is NULL, implying that the bandwidth is calculated based on the rule-of-thumb.
bwfactor	Factor by which the bandwidth is multiplied. Default is 0.7 (undersmoothing).
cluster	Optional clustering variable for calculating standard errors.
k	Number of folds in k-fold cross-fitting. Default is 3.

Details

This function estimates the Average Treatment Effect on the Treated (ATET) by Difference-in-Differences in panel data while controlling for confounders using double machine learning. The function supports different machine learning methods for estimating nuisance parameters and performs k-fold cross-fitting to improve estimation accuracy. The function also handles binary and continuous outcomes, and provides options for trimming and bandwidth adjustments in kernel density estimation.

Value

A list with the following components:

`ATET`: Estimate of the Average Treatment Effect on the Treated.

`se`: Standard error of the ATET estimate.

`trimmed`: Number of discarded (trimmed) observations.

`pval`: P-value.

`pscores`: Propensity scores of untrimmed observations (2 columns): under treatment, under control.

`outcomepred`: Conditional outcome predictions of untrimmed observations.

`treat`: Treatment status of untrimmed observations.

References

Chernozhukov, V., Chetverikov, D., Demirer, M., Duflo, E., Hansen, C., Newey, W., Robins, J. (2018): "Double/debiased machine learning for treatment and structural parameters", The Econometrics Journal, 21, C1-C68.

Haddad, M., Huber, M., Medina-Reyes, J., Zhang, L. (2024): "Difference-in-Differences with Time-varying Continuous Treatments using Double/Debiased Machine Learning", working paper, University of Fribourg.

Examples

```
## Not run:
# Example with simulated data
n=1000
x=0.5*rnorm(n)
u=runif(n,0,2)
d=x+u+rnorm(n)
y0=u+rnorm(n)
y1=2*d+x+u+rnorm(n)
t=rep(1,n)
# true effect is 2
results=didcontDMLpanel(ydiff=y1-y0, d=d, t=t, dtreat=1, dcontrol=0, controls=x, MLmethod="lasso")
cat("ATET: ", round(results$ATET, 3), ", Standard error: ", round(results$se, 3))

## End(Not run)
```

didDML *Difference-in-Differences in Repeated Cross-Sections for Binary Treatments using Double Machine Learning*

Description

This function estimates the average treatment effect on the treated (ATET) in the post-treatment period for a binary treatment using a doubly robust Difference-in-Differences (DiD) approach for repeated cross-sections that is combined with double machine learning. It controls for (possibly time-varying) confounders in a data-driven manner and supports various machine learning methods for estimating nuisance parameters through k-fold cross-fitting.

Usage

```
didDML(
  y,
  d,
  t,
  x,
  MLmethod = "lasso",
  est = "dr",
  trim = 0.05,
  cluster = NULL,
  k = 3
)
```

Arguments

y	Outcome variable. Should not contain missing values.
d	Treatment group indicator (binary). Should not contain missing values.
t	Time period indicator (binary). Should be 1 for post-treatment period and 0 for pre-treatment period. Should not contain missing values.
x	Covariates to be controlled for. Should not contain missing values.
MLmethod	Machine learning method for estimating nuisance parameters using the SuperLearner package. Must be one of "lasso" (default), "randomforest", "xgboost", "svm", "ensemble", or "parametric".
est	Estimation method. Must be one of "dr" (default) for doubly robust, "ipw" for inverse probability weighting (not doubly robust!), or "reg" for regression (not doubly robust!).
trim	Trimming threshold (in percent) for discarding observations with too small propensity scores within any subgroup defined by the treatment group and time. Default is 0.05.
cluster	Optional clustering variable for calculating cluster-robust standard errors.
k	Number of folds in k-fold cross-fitting. Default is 3.

Details

This function estimates the Average Treatment Effect on the Treated (ATET) in the post-treatment period based on Difference-in-Differences in repeated cross-sections when controlling for confounders in a data-adaptive manner using double machine learning. The function supports different machine learning methods to estimate nuisance parameters (conditional mean outcomes and propensity scores) as well as cross-fitting to mitigate overfitting. Besides double machine learning, the function also provides inverse probability weighting and regression adjustment methods (which are, however, not doubly robust).

Value

A list with the following components:

`ATET`: Estimate of the Average Treatment Effect on the Treated (ATET) in the post-treatment period.

`se`: Standard error of the ATET estimate.

`pval`: P-value of the ATET estimate.

`trimmed`: Number of discarded (trimmed) observations.

`pscores`: Propensity scores of untrimmed observations (4 columns): under treatment in period 1, under treatment in period 0, under control in period 1, under control in period 0.

`outcomepred`: Conditional outcome predictions of untrimmed observations (3 columns): in treatment group in period 0, in control group in period 1, in control group in period 0.

`treat`: Treatment status of untrimmed observations.

`time`: Time period of untrimmed observations.

References

Chernozhukov, V., Chetverikov, D., Demirer, M., Duflo, E., Hansen, C., Newey, W., Robins, J. (2018): "Double/debiased machine learning for treatment and structural parameters", The Econometrics Journal, 21, C1-C68.

Zimmert, M. (2020): "Efficient difference-in-differences estimation with high-dimensional common trend confounding", arXiv preprint 1809.01643.

Examples

```
## Not run:
# Example with simulated data
n=4000                          # sample size
t=1*(rnorm(n)>0)                # time period
u=runif(n,0,1)                  # time constant unobservable
x= 0.25*t+runif(n,0,1)          # time varying covariate
d=1*(x+u+2*rnorm(n)>0)          # treatment
y=d*t+t+x+u+2*rnorm(n)          # outcome
# true effect is equal to 1
results=didDML(y=y, d=d, t=t, x=x)
cat("ATET: ", round(results$ATET, 3), ", Standard error: ", round(results$se, 3))

## End(Not run)
```

`didmedDML` *Difference-in-Differences for Mediation Analysis with Repeated Cross-Sections and Discrete Treatments Using Double Machine Learning*

Description

This function estimates the total effect, natural direct effect, and natural indirect effect for the treated group in the post-treatment period in repeated cross-sections with discrete treatments. Estimation is based on a difference-in-differences approach for mediation analysis combined with double machine learning to control for (possibly time-varying) confounders in a data-driven manner. The function supports various machine learning methods for estimating nuisance parameters through k-fold cross-fitting.

Usage

```
didmedDML(
  y,
  d,
  t,
  m,
  x,
  dtreat = 1,
  dcontrol = 0,
  MLmethod = "lasso",
  trim = 0.05,
  cluster = NULL,
  k = 3
)
```

Arguments

`y`	Outcome variable. Should not contain missing values.
`d`	Treatment group indicator (discrete). Should not contain missing values.
`t`	Time period indicator (binary). Should be 1 for post-treatment period and 0 for pre-treatment period. Should not contain missing values.
`m`	Mediator variable. Should not contain missing values.
`x`	Covariates to be controlled for. Should not contain missing values.
`dtreat`	Value of the treatment under treatment (in the treatment period of interest). Default is 1.
`dcontrol`	Value of the treatment under control (in the treatment period of interest). Default is 0.
`MLmethod`	Machine learning method for estimating nuisance parameters using the `SuperLearner` package. Must be one of `"lasso"` (default), `"randomforest"`, `"xgboost"`, `"svm"`, `"ensemble"`, or `"parametric"`.

trim	Trimming threshold for discarding observations with too small propensity scores in the treated group in the pre-treatment period and in the control group in either period. Default is 0.05.
cluster	Optional clustering variable for calculating cluster-robust standard errors.
k	Number of folds in k-fold cross-fitting. Default is 3.

Details

This function estimates the total effect, natural direct effect, and natural indirect effect for the treated group in the post-treatment period in repeated cross-sections with discrete treatments. Estimation is based on the difference-in-differences approach for mediation analysis proposed by Huber and Oberhänsli (2026). Specifically, these effects are computed from normalized sample analogs of the doubly robust expressions in equations (18) and (20). Double machine learning is used to control for confounders in a data-adaptive way. The function supports different machine learning methods for estimating nuisance parameters (conditional mean outcomes and propensity scores) as well as cross-fitting to mitigate overfitting.

Value

A list with the following components:

eff: Estimates of the total effect, natural direct effect, and natural indirect effect for the treated group in the post-treatment period.

se: Standard error of the estimates.

tval: t-value of the estimates.

pval: p-value of the estimates.

trimmed: Number of discarded (trimmed) observations.

References

Chernozhukov, V., Chetverikov, D., Demirer, M., Duflo, E., Hansen, C., Newey, W., Robins, J. (2018): "Double/debiased machine learning for treatment and structural parameters", The Econometrics Journal, 21, C1-C68.

Huber, M., and Oberhänsli, S. J. (2026): "Difference-in-differences for mediation analysis using double machine learning", arXiv preprint 2602.23877.

Examples

```
## Not run:
# Example with simulated data
n=4000                          # sample size
t=rbinom(n, 1, 0.5)             # time period
u=rnorm(n)                      # time constant unobservable
x=0.5*t+rnorm(n)                # time varying covariate
d=1*(x+0.5*u+rnorm(n)>0)        # treatment
m=x+0.5*d+rnorm(n)              # mediator
y=x+(1+d+m)*t+u+rnorm(n)        # outcome
# true NDET is equal to 1; true NIET is equal to 0.5; true ATET is equal to 1.5
didmedDML(y=y, d=d, t=t, m=m, x=x)
```

```
## End(Not run)
```

`didmedDMLpanel` *Difference-in-Differences for Mediation Analysis with Panel Data and Discrete Treatments Using Double Machine Learning*

Description

This function estimates the total effect, natural direct effect, and natural indirect effect for the treated group in the post-treatment period in panel data with discrete treatments. Estimation is based on a difference-in-differences approach for mediation analysis combined with double machine learning to control for (possibly time-varying) confounders in a data-driven manner. The function supports various machine learning methods for estimating nuisance parameters through k-fold cross-fitting.

Usage

```
didmedDMLpanel(
  y0,
  y1,
  d,
  m,
  x,
  dtreat = 1,
  dcontrol = 0,
  MLmethod = "lasso",
  trim = 0.05,
  cluster = NULL,
  k = 3
)
```

Arguments

`y0`	Outcome variable in the pre-treatment period. Should not contain missing values.
`y1`	Outcome variable in the post-treatment period. Should not contain missing values.
`d`	Treatment group indicator (discrete). Should not contain missing values.
`m`	Mediator variable. Should not contain missing values.
`x`	Covariates to be controlled for. Should not contain missing values.
`dtreat`	Value of the treatment under treatment (in the treatment period of interest). Default is 1.
`dcontrol`	Value of the treatment under control (in the treatment period of interest). Default is 0.

MLmethod	Machine learning method for estimating nuisance parameters using the SuperLearner package. Must be one of "lasso" (default), "randomforest", "xgboost", "svm", "ensemble", or "parametric".
trim	Trimming threshold for discarding observations with too small propensity scores in the control group. Default is 0.05.
cluster	Optional clustering variable for calculating cluster-robust standard errors.
k	Number of folds in k-fold cross-fitting. Default is 3.

Details

This function estimates the total effect, natural direct effect, and natural indirect effect for the treated group in the post-treatment period in panel data with discrete treatments. Estimation is based on the difference-in-differences approach for mediation analysis proposed by Huber and Oberhänsli (2026). Specifically, these effects are computed from normalized sample analogs of the doubly robust expressions in equations (27) and (28). Double machine learning is used to control for confounders in a data-adaptive way. The function supports different machine learning methods for estimating nuisance parameters (conditional mean outcomes and propensity scores) as well as cross-fitting to mitigate overfitting.

Value

A list with the following components:

eff: Estimates of the total effect, natural direct effect, and natural indirect effect for the treated group in the post-treatment period.

se: Standard error of the estimates.

tval: t-value of the estimates.

pval: p-value of the estimates.

trimmed: Number of discarded (trimmed) observations.

References

Chernozhukov, V., Chetverikov, D., Demirer, M., Duflo, E., Hansen, C., Newey, W., Robins, J. (2018): "Double/debiased machine learning for treatment and structural parameters", The Econometrics Journal, 21, C1-C68.

Huber, M., and Oberhänsli, S. J. (2026): "Difference-in-differences for mediation analysis using double machine learning", arXiv preprint 2602.23877.

Examples

```
## Not run:
# Example with simulated data
n=4000                          # sample size
u=rnorm(n)                      # time constant unobservable
x=rnorm(n)                      # covariate
d=1*(x+0.5*u+rnorm(n)>0)        # treatment
m=x+0.5*d+rnorm(n)              # mediator
y0=u+rnorm(n)                   # outcome in the pre-treatment period
```

```
y1=x+1+d+m+u+rnorm(n)             # outcome in the post-treatment period
# true NDET is equal to 1; true NIET is equal to 0.5; true ATET is equal to 1.5
didmedDMLpanel(y0=y0,y1=y1, d=d, m=m, x=x)

## End(Not run)
```

`didweight` *Difference-in-differences based on inverse probability weighting*

Description

Difference-in-differences-based estimation of the average treatment effect on the treated in the post-treatment period, given a binary treatment with one pre- and one post-treatment period. Permits controlling for differences in observed covariates across treatment groups and/or time periods based on inverse probability weighting.

Usage

```
didweight(y, d, t, x = NULL, boot = 1999, trim = 0.05, cluster = NULL)
```

Arguments

`y`	Dependent variable, must not contain missings.
`d`	Treatment, must be binary (either 1 or 0), must not contain missings.
`t`	Time period, must be binary, 0 for pre-treatment and 1 for post-treatment, must not contain missings.
`x`	Covariates to be controlled for by inverse probability weighting. Default is `NULL`.
`boot`	Number of bootstrap replications for estimating standard errors. Default is 1999.
`trim`	Trimming rule for discarding observations with extreme propensity scores in the 3 reweighting steps, which reweight (1) treated in the pre-treatment period, (2) non-treated in the post-treatment period, and (3) non-treated in the pre-treatment period according to the covariate distribution of the treated in the post-treatment period. Default is 0.05, implying that observations with a probability lower than 5 percent of not being treated in some weighting step are discarded.
`cluster`	A cluster ID for block or cluster bootstrapping when units are clustered rather than iid. Must be numerical. Default is `NULL` (standard bootstrap without clustering).

Details

Estimation of the average treatment effect on the treated in the post-treatment period based Difference-in-differences. Inverse probability weighting is used to control for differences in covariates across treatment groups and/or over time. That is, (1) treated observations in the pre-treatment period, (2) non-treated observations in the post-treatment period, and (3) non-treated observations in the pre-treatment period are reweighted according to the covariate distribution of the treated observations in the post-treatment period. The respective propensity scores are obtained by probit regressions.

Value

A didweight object contains 4 components, `eff`, `se`, `pvalue`, and `ntrimmed`.

`eff`: estimate of the average treatment effect on the treated in the post-treatment period.

`se`: standard error obtained by bootstrapping the effect.

`pvalue`: p-value based on the t-statistic.

`ntrimmed`: total number of discarded (trimmed) observations in any of the 3 reweighting steps due to extreme propensity score values.

References

Abadie, A. (2005): "Semiparametric Difference-in-Differences Estimators", The Review of Economic Studies, 72, 1-19.

Lechner, M. (2011): "The Estimation of Causal Effects by Difference-in-Difference Methods", Foundations and Trends in Econometrics, 4, 165-224.

Examples

```
# A little example with simulated data (4000 observations)
## Not run:
n=4000                          # sample size
t=1*(rnorm(n)>0)                # time period
u=rnorm(n)                      # time constant unobservable
x=0.5*t+rnorm(n)                # time varying covariate
d=1*(x+u+rnorm(n)>0)            # treatment
y=d*t+t+x+u+rnorm(n)            # outcome
# The true effect equals 1
didweight(y=y,d=d,t=t,x=x, boot=199)
## End(Not run)
```

`dyntreatDML` *Dynamic treatment effect evaluation with double machine learning*

Description

Dynamic treatment effect estimation for assessing the average effects of sequences of treatments (consisting of two sequential treatments). Combines estimation based on (doubly robust) efficient score functions with double machine learning to control for confounders in a data-driven way, as considered in Bodory, Huber, and Laffers (2022): "Evaluating (weighted) dynamic treatment effects by double machine learning".

Usage

```
dyntreatDML(
  y2,
  d1,
  d2,
```

```
  x0,
  x1,
  s = NULL,
  d1treat = 1,
  d2treat = 1,
  d1control = 0,
  d2control = 0,
  trim = 0.01,
  MLmethod = "lasso",
  fewsplits = FALSE,
  normalized = TRUE,
  debiasfirst = TRUE
)
```

Arguments

y2	Dependent variable in the second period (=outcome period), must not contain missings.
d1	Treatment in the first period, must be discrete, must not contain missings.
d2	Treatment in the second period, must be discrete, must not contain missings.
x0	Covariates in the baseline period (prior to the treatment in the first period), must not contain missings.
x1	Covariates in the first period (prior to the treatment in the second period), must not contain missings.
s	Indicator function for defining a subpopulation for whom the treatment effect is estimated as a function of the subpopulation's distribution of x0. Default is NULL (estimation of the treatment effect in the total population).
d1treat	Value of the first treatment in the treatment sequence. Default is 1.
d2treat	Value of the second treatment in the treatment sequence. Default is 1.
d1control	Value of the first treatment in the control sequence. Default is 0.
d2control	Value of the second treatment in the control sequence. Default is 0.
trim	Trimming rule for discarding observations with products of treatment propensity scores in the first and second period that are smaller than trim (to avoid too small denominators in weighting by the inverse of the propensity scores). Default is 0.01.
MLmethod	Machine learning method for estimating the nuisance parameters based on the SuperLearner package. Must be either "lasso" (default) for lasso estimation, "randomforest" for random forests, "xgboost" for xg boosting, "svm" for support vector machines, "ensemble" for using an ensemble algorithm based on all previously mentioned machine learners, or "parametric" for linear or logit regression.
fewsplits	If set to TRUE, the same training data are used for estimating a nested model of conditional mean outcomes, namely E[E[y2|d1,d2,x0,x1]|d1,x0]. If fewsplits is FALSE, the training data are split for the sequential estimation of the nested model. Default of fewsplits is FALSE.

`normalized` If set to `TRUE`, then the inverse probability-based weights are normalized such that they add up to 1 within treatment groups. Default is `TRUE`.

`debiasfirst` If set to `TRUE`, then the first step conditional mean outcome `E[y2|d1,d2,x0,x1]` is debiased before estimating the nested conditional mean outcome `E[E[y2|d1,d2,x0,x1]|d1,x0]`, following Bradic, Ji, and Zhang (2024): "High-dimensional inference for dynamic treatment effects". Default is `TRUE`.

Details

Estimation of the causal effects of sequences of two treatments under sequential conditional independence, assuming that all confounders of the treatment in either period and the outcome of interest are observed. Estimation is based on the (doubly robust) efficient score functions for potential outcomes, see e.g. Bodory, Huber, and Laffers (2020), in combination with double machine learning with cross-fitting, see Chernozhukov et al (2018). To this end, one part of the data is used for estimating the model parameters of the treatment and outcome equations based machine learning. The other part of the data is used for predicting the efficient score functions. The roles of the data parts are swapped (using 3-fold cross-fitting) and the average dynamic treatment effect is estimated based on averaging the predicted efficient score functions in the total sample. Standard errors are based on asymptotic approximations using the estimated variance of the (estimated) efficient score functions.

Value

A `dyntreatDML` object contains ten components, `effect`, `se`, `pval`, `ntrimmed`, `meantreat`, `meancontrol`, `psd1treat`, `psd2treat`, `psd1control`, and `psd2control` :

`effect`: estimate of the average effect of the treatment sequence.

`se`: standard error of the effect estimate.

`pval`: p-value of the effect estimate.

`ntrimmed`: number of discarded (trimmed) observations due to low products of propensity scores.

`meantreat`: Estimate of the mean potential outcome under the treatment sequence.

`meancontrol`: Estimate of the mean potential outcome under the control sequence.

`psd1treat`: P-score estimates for first treatment in treatment sequence.

`psd2treat`: P-score estimates for second treatment in treatment sequence.

`psd1control`: P-score estimates for first treatment in control sequence.

`psd2control`: P-score estimates for second treatment in control sequence.

References

Bodory, H., Huber, M., Laffers, L. (2022): "Evaluating (weighted) dynamic treatment effects by double machine learning", Econometrics Journal, 25, 628-648.

Bradic, J., Ji, W., Zhang, Y. (2024): "High-dimensional inference for dynamic treatment effects", The Annals of Statistics, 52, 415-440.

Chernozhukov, V., Chetverikov, D., Demirer, M., Duflo, E., Hansen, C., Newey, W., Robins, J. (2018): "Double/debiased machine learning for treatment and structural parameters", The Econometrics Journal, 21, C1-C68.

van der Laan, M., Polley, E., Hubbard, A. (2007): "Super Learner", Statistical Applications in Genetics and Molecular Biology, 6.

Examples

```
# A little example with simulated data (2000 observations)
## Not run:
n=2000
# sample size
p0=10
# number of covariates at baseline
s0=5
# number of covariates that are confounders at baseline
p1=10
# number of additional covariates in period 1
s1=5
# number of additional covariates that are confounders in period 1
x0=matrix(rnorm(n*p0),ncol=p0)
# covariate matrix at baseline
beta0=c(rep(0.25,s0), rep(0,p0-s0))
# coefficients determining degree of confounding for baseline covariates
d1=(x0%*%beta0+rnorm(n)>0)*1
# equation of first treatment in period 1
x1=matrix(rnorm(n*p1),ncol=p1)+matrix(0.1 * d1, nrow = n, ncol = p1)
# covariate matrix for covariates of period 1 (affected by 1st treatment d1)
beta1=c(rep(0.25,s1), rep(0,p1-s1))
# coefficients determining degree of confounding for covariates of period 1
d2=(x0%*%beta0+x1%*%beta1+0.5*d1+rnorm(n)>0)*1
# equation of second treatment in period 2
y2=x0%*%beta0+x1%*%beta1+1*d1+0.5*d2+rnorm(n)
# outcome equation in period 2
output=dyntreatDML(y2=y2,d1=d1,d2=d2,x0=x0,x1=x1,
       d1treat=1,d2treat=1,d1control=0,d2control=0)
cat("dynamic ATE: ",round(c(output$effect),3),", standard error: ",
    round(c(output$se),3), ", p-value: ",round(c(output$pval),3))
output$ntrimmed
# The true effect of the treatment sequence is 1.5
## End(Not run)
```

`games` *Sales of video games*

Description

A dataset containing information on 3956 video games, including sales as well as expert and user ratings.

Usage

```
games
```

Format

A data frame with 3956 rows and 9 variables:

name factor variable providing the name of the video game

genre factor variable indicating the genre of the game (e.g. Action, Sports...)

platform factor variable indicating the hardware platform of the game (e.g. PC,...)

esrbrating factor variable indicating the age recommendation for the game(E is age 6+, T is 13+, M is 17+)

publisher factor variable indicating the publisher of the game

year numeric variable indicating the year the video game was released

metascore numeric variable providing a weighted average rating of the game by professional critics

userscore numeric variable providing the average user rating of the game

sales numeric variable indicating the total global sales (in millions) of the game up to the year 2018

References

Wittwer, J. (2020): "Der Erfolg von Videospielen - eine empirische Untersuchung moeglicher Erfolgsfaktoren", BA thesis, University of Fribourg.

Examples

```
## Not run:
#load data
data(games)
#select non-missing observations
games_nomis=na.omit(games)
#turn year into a factor variable
games_nomis$year=factor(games_nomis$year)
#attach data
attach(games_nomis)
#load library for generating dummies
library(fastDummies)
#generate dummies for genre
dummies=dummy_cols(genre, remove_most_frequent_dummy = TRUE)
#drop original variable
genredummies=dummies[,2:ncol(dummies)]
#make dummies numeric
genredummies=apply(genredummies, 2, function(genredummies) as.numeric(genredummies))
#generate dummies for year
dummies=dummy_cols(year, remove_most_frequent_dummy = TRUE)
#drop original variable
yeardummies=dummies[,2:ncol(dummies)]
#make dummies numeric
yeardummies=apply(yeardummies, 2, function(yeardummies) as.numeric(yeardummies))
# mediation analysis with metascore as treatment, userscore as mediator, sales as outcome
x=cbind(genredummies,yeardummies)
output=medweightcont(y=sales,d=metascore, d0=60, d1=80, m=userscore, x=x, boot=199)
round(output$results,3)
output$ntrimmed
## End(Not run)
```

`identificationDML` *Testing identification with double machine learning*

Description

Testing identification with double machine learning

Usage

```
identificationDML(
  y,
  d,
  x,
  z,
  score = "DR",
  bootstrap = FALSE,
  ztreat = 1,
  zcontrol = 0,
  seed = 123,
  MLmethod = "lasso",
  k = 3,
  DR_parameters = list(s = NULL, normalized = TRUE, trim = 0.01),
  squared_parameters = list(zeta_sigma = min(0.5, 500/dim(y)[1])),
 bootstrap_parameters = list(B = 2000, importance = 0.95, alpha = 0.1, share = 0.5)
)
```

Arguments

`y`	Dependent variable, must not contain missings.
`d`	Treatment variable, must be discrete, must not contain missings.
`x`	Covariates, must not contain missings.
`z`	Instrument, must not contain missings.
`score`	Orthogonal score used for testing identification, either `"DR"` for using the average of the doubly robust (DR) score function (see Section 6 of Huber and Kueck, 2022) for testing, or `"squared"` for using squared differences in the conditional means outcomes (see Section 7 of Huber and Kueck, 2022). Default is `"DR"`. Note that this argument is ignored if `bootstrap=TRUE`.
`bootstrap`	If set to `TRUE`, testing identification is based on the DR score function within data-driven partitioning of the data (using a random forest with 200 trees) as described at the end of Sections 6 and 8 in Huber and Kueck (2022). Default is `FALSE`. Note that the argument `score` is ignored if `bootstrap=TRUE`.
`ztreat`	Value of the instrument in the "treatment" group. Default is 1.
`zcontrol`	Value of the instrument in the "control" group. Default is 0.
`seed`	Default is 123.

MLmethod Machine learning method for estimating the nuisance parameters based on the SuperLearner package. Must be either "lasso" (default) for lasso estimation, "randomforest" for random forests, "xgboost" for xg boosting, "svm" for support vector machines, "ensemble" for using an ensemble algorithm based on all previously mentioned machine learners, or "parametric" for linear or logit regression.

k Number of folds in k-fold cross-fitting. Default is 3.

DR_parameters List of input parameters to test identification using the doubly robust score: s: Indicator function for defining a subpopulation for which the treatment effect is estimated as a function of the subpopulation's distribution of x. Default is NULL (estimation of the average treatment effect in the total population). normalized: If set to TRUE, then the inverse probability-based weights are normalized such that they add up to 1 within treatment groups. Default is TRUE trim: Trimming rule for discarding observations with treatment propensity scores that are smaller than trim or larger than 1-trim (to avoid too small denominators in weighting by the inverse of the propensity scores). Default is 0.01.

squared_parameters List of input parameters to test identification using the squared deviation: zeta_sigma: standard deviation of the normal distributed errors to avoid degenerated limit distribution. Default is min(0.05,500/n).

bootstrap_parameters List of input parameters to test identification using the DR score and sample splitting to detect heterogeneity (if bootstrap=TRUE): B: number of bootstrap samples to be used in the multiplier bootstrap. Default is 2000. importance: upper quantile of covariates in terms of their predictive importance for heterogeneity in the DR score function according to a random forest (with 200 trees). The data are split into subsets based on the median values of these predictive covariates (entering the upper quantile). Default is 0.95. alpha: level of the statistical test. Default is 0.1. share: share of observations used to detect heterogeneity in the DR score function by the random forest (while the remaining observations are used for hypothesis testing). Default is 0.5.

Details

Testing the identification of causal effects of a treatment d on an outcome y in observational data using a supposed instrument z and controlling for observed covariates x.

Value

An identificationDML object contains different parameters, at least the two following:

effect: estimate of the target parameter(s).

pval: p-value(s) of the identification test.

References

Huber, M., & Kueck, J. (2022): Testing the identification of causal effects in observational data. arXiv:2203.15890.

Examples

```
# Two examples with simulated data
## Not run:
set.seed(777)
n <- 20000  # sample size
p <- 50    # number of covariates
s <- 5  # sparsity (relevant covariates)
alpha <- 0.1    # level

control violation of identification
delta <- 2    # effect of unobservable in outcome on index of treatment - either 0 or 2
gamma <- 0   # direct effect of the instrument on outcome  - either 0 or 0.1

DGP - general
xcorr <- 1   # if 1, then non-zero covariance between regressors
if (xcorr == 0) {
 sigmax <- diag(1,p)}       # covariate matrix at baseline
if (xcorr != 0){
 sigmax = matrix(NA,p,p)
for (i in 1:p){
   for (j in 1:p){
     sigmax[i,j] = 0.5^(abs(i-j))
   }
 }}
sparse = FALSE # if FALSE, an approximate sparse setting is considered
beta = rep(0,p)
if (sparse == TRUE){
 for (j in 1:s){ beta[j] <- 1} }
if (sparse != TRUE){
 for (j in 1:p) beta[j] <- (1/j)}
noise_U <- 0.1 # control signal-to-noise
noise_V <- 0.1
noise_W <- 0.25
x <- (rmvnorm(n,rep(0,p),sigmax))
w <- rnorm(n,0,sd=noise_W)
z <- 1*(rnorm(n)>0)
d <- (x%*%beta+z+w+rnorm(n,0,sd=noise_V)>0)*1         # treatment equation

DGP 1 - effect homogeneity

y <- x%*%beta+d+gamma*z+delta*w+rnorm(n,0,sd=noise_U)

output1 <- identificationDML(y = y, d=d, x=x, z=z, score = "DR", bootstrap = FALSE,
ztreat = 1, zcontrol = 0 , seed = 123, MLmethod ="lasso", k = 3,
DR_parameters = list(s = NULL , normalized = TRUE, trim = 0.01))
output1$pval
output2 <- identificationDML(y=y, d=d, x=x, z=z, score = "squared", bootstrap = FALSE,
ztreat = 1, zcontrol =0 , seed = 123, MLmethod ="lasso", k = 3)
output2$pval
output3 <- identificationDML(y=y, d=d, x=x, z=z, score = "squared", bootstrap = TRUE,
ztreat = 1, zcontrol =0 , seed = 123, MLmethod ="lasso", k = 3,
DR_parameters = list(s = NULL , normalized = TRUE, trim = 0.005),
```

```
bootstrap_parameters = list(B = 2000, importance = 0.95, alpha = 0.1, share = 0.5))
output3$pval

DGP 2 - effect heterogeneity

y = x%*%beta+d+gamma*z*x[,1]+gamma*z*x[,2]+delta*w*x[,1]+delta*w*x[,2]+rnorm(n/2,0,sd=noise_U)

output1 <- identificationDML(y = y, d=d, x=x, z=z, score = "DR", bootstrap = FALSE,
ztreat = 1, zcontrol = 0 , seed = 123, MLmethod ="lasso", k = 3,
DR_parameters = list(s = NULL , normalized = TRUE, trim = 0.01))
output1$pval
output2 <- identificationDML(y=y, d=d, x=x, z=z, score = "squared", bootstrap = FALSE,
ztreat = 1, zcontrol =0 , seed = 123, MLmethod ="lasso", k = 3)
output2$pval
output3 <- identificationDML(y=y, d=d, x=x, z=z, score = "DR", bootstrap = TRUE,
ztreat = 1, zcontrol =0 , seed = 123, MLmethod ="lasso", k = 3,
DR_parameters = list(s = NULL , normalized = TRUE, trim = 0.005),
bootstrap_parameters = list(B = 2000, importance = 0.95, alpha = 0.1, share = 0.5))
output3$pval

## End(Not run)
```

`india` *India's National Health Insurance Program (RSBY)*

Description

A dataset which is a subset of the data from the randomized evaluation of the India's National Health Insurance Program (RSBY).

Usage

```
india
```

Format

A data frame with 11'089 rows and 7 variables:

X individual identification

village_id village identification

DistrictId district identification

treat treatment status (insurance)

mech treatment assignment mechanism

enrolled enrolled

EXPhosp_1 hospital expenditure

References

Imai, Kosuke; Jiang, Zhichao; Malani, Anup, 2020, "Replication Data for: Causal Inference with Interference and Noncompliance in Two-Stage Randomized Experiments.", https://doi.org/10.7910/DVN/N7D9LS, Harvard Dataverse, V1

Examples

```
## Not run:
require(devtools)                        # load devtools package
install_github("szonszein/interference") # install interference package
library(interference)                    # load interference package
data(india)                              # load data
attach(india)                            # attach data
india=na.omit(india)                     # drop observations with missings
group=india$village_id                   # cluster id
group_tr=india$mech                      # indicator high treatment proportion
indiv_tr=india$treat                     # individual treatment (insurance)
obs_outcome=india$EXPhosp_1              # outcome  (hospital expenditure)
dat=data.frame(group,group_tr,indiv_tr,obs_outcome) # generate data frame
estimates_hierarchical(dat)
## End(Not run)             # run estimation
```

---

ivnr *Instrument-based treatment evaluation under endogeneity and non-response bias*

Description

Non- and semiparaemtric treatment effect estimation under treatment endogeneity and selective non-response in the outcome based on a binary instrument for the treatment and a continous instrument for response.

Usage

```
ivnr(
  y,
  d,
  r,
  z1,
  z2,
  x = NULL,
  xpar = NULL,
  ruleofthumb = 1,
  wgtfct = 2,
  rtype = "ll",
  numresprob = 20,
  boot = 499,
  estlate = TRUE,
```

```
  trim = 0.01
)
```

Arguments

y	Dependent variable.
d	Treatment, must be binary and must not contain missings.
r	Response, must be a binary indicator for whether the outcome is observed.
z1	Binary instrument for the treatment, must not contain missings.
z2	Continuous instrument for response, must not contain missings.
x	A data frame of covariates to be included in the nonparametric estimation, must not contain missings. Factors and ordered varaibles must be appropriately defined as such by `factor()` and `ordered()`. Default is `NULL` (no covariates included). Covariates are only considered if both `x` and `xpar` are not `NULL`.
xpar	Covariates to be included in the semiparametric estimation, must not contain missings. Default is `NULL` (no covariates included). Covariates are only considered if both `x` and `xpar` are not `NULL`.
ruleofthumb	If 1, bandwidth selection in any kernel function is based on the Silverman (1986) rule of thumb. Otherwise, least squares cross-validation is used. Default is 1.
wgtfct	Weighting function to be used in effect estimation. If set to 1, equation (18) in Fricke et al (2020) is used as weight. If set to 2, equation (19) in Fricke et al (2020) is used as weight. If set to 3, the median of LATEs across values of response probabilities `numresprob` is used. Default is 2.
rtype	Regression type used for continuous outcomes in the kernel regressions. Either `"ll"` for local linear or `"lc"` for local constant regression. Default is `"ll"`.
numresprob	number of response probabilities at which the effects are evaluated. An equidistant grid is constructed based on the number provided. Default is 20.
boot	Number of bootstrap replications for estimating standard errors of the effects. Default is 499.
estlate	If set to `TRUE` the local average treatment effect on compliers (LATE) is estimated, otherwise the average treatment effect (ATE) is estimated. Default is `TRUE`.
trim	Trimming rule for too extreme denominators in the weighting functions or inverses of products of conditional treatment probabilities. Values below `trim` are set to `trim` to avoid values that are too close to zero in any denominator. Default is 0.01.

Details

Non- and semiparametric treatment effect estimation under treatment endogeneity and selective non-response in the outcome based on a binary instrument for the treatment and a continuous instrument for response. The effects are estimated both semi-parametrically (using probit and OLS for the estimation of plug-in parameters like conditional probabilities and outcomes) and fully non-parametrically (based on kernel regression for any conditional probability/mean). Besides the instrument-based estimates, results are also presented under a missing-at-random assumption (MAR) when not using the instrument `z2` for response (but only `z1` for the treatment). See Fricke et al. (2020) for further details.

Value

A ivnr object contains one output component:

output: The first row provides the effect estimates under non- and semi-parametric estimation using both instruments, see "nonpara (L)ATE IV" and "semipara (L)ATE IV" as well as under a missing-at-random assumption for response when using only the first instrument for the treatment, see "nonpara (L)ATE MAR" and "semipara (L)ATE MAR". The second row provides the standard errors based on bootstrapping the effects. The third row provides the p-values based on the t-statistics.

References

Fricke, H., Frölich, M., Huber, M., Lechner, M. (2020): "Endogeneity and non-response bias in treatment evaluation - nonparametric identification of causal effects by instruments", Journal of Applied Econometrics, forthcoming.

Examples

```
# A little example with simulated data (1000 observations)
## Not run:
n=1000          # sample size
e<-(rmvnorm(n,rep(0,3), matrix(c(1,0.5,0.5,  0.5,1,0.5,  0.5,0.5,1),3,3)))
# correlated error term of treatment, response, and outcome equation
x=runif(n,-0.5,0.5)           # observed confounder
z1<-(-0.25*x+rnorm(n)>0)*1    # binary instrument for treatment
z2<- -0.25*x+rnorm(n)         # continuous instrument for selection
d<-(z1-0.25*x+e[,1]>0)*1      # treatment equation
 y_star<- -0.25*x+d+e[,2]     # latent outcome
 r<-(-0.25*x+z2+d+e[,3]>0)*1  # response equation
 y=y_star                     # observed outcome
 y[r==0]=0                    # nonobserved outcomes are set to zero
 # The true treatment effect is 1
 ivnr(y=y,d=d,r=r,z1=z1,z2=z2,x=x,xpar=x,numresprob=4,boot=39)
## End(Not run)
```

JC *Job Corps data*

Description

A dataset from the U.S. Job Corps experimental study with information on the participation of disadvantaged youths in (academic and vocational) training in the first and second year after program assignment.

Usage

```
JC
```

Format

A data frame with 9240 rows and 46 variables:

assignment 1=randomly assigned to Job Corps, 0=randomized out of Job Corps

female 1=female, 0=male

age age in years at assignment

white 1=white, 0=non-white

black 1=black, 0=non-black

hispanic 1=hispanic, 0=non-hispanic

educ years of education at assignment

educmis 1=education missing at assignment

geddegree 1=has a GED degree at assignment

hsdegree 1=has a high school degree at assignment

english 1=English mother tongue

cohabmarried 1=cohabiting or married at assignment

haschild 1=has at least one child, 0=no children at assignment

everwkd 1=has ever worked at assignment, 0=has never worked at assignment

mwearn average weekly gross earnings at assignment

hhsize household size at assignment

hhsizemis 1=household size missing

educmum mother's years of education at assignment

educmummis 1=mother's years of education missing

educdad father's years of education at assignment

educdadmis 1=father's years of education missing

welfarechild welfare receipt during childhood in categories from 1 to 4 (measured at assignment)

welfarechildmis 1=missing welfare receipt during childhood

health general health at assignment from 1 (excellent) to 4 (poor)

healthmis 1=missing health at assignment

smoke extent of smoking at assignment in categories from 1 to 4

smokemis 1=extent of smoking missing

alcohol extent of alcohol consumption at assignment in categories from 1 to 4

alcoholmis 1=extent of alcohol consumption missing

everwkdy1 1=has ever worked one year after assignment, 0=has never worked one year after assignment

earnq4 weekly earnings in fourth quarter after assignment

earnq4mis 1=missing weekly earnings in fourth quarter after assignment

pworky1 proportion of weeks employed in first year after assignment

pworky1mis 1=missing proportion of weeks employed in first year after assignment

health12 general health 12 months after assignment from 1 (excellent) to 4 (poor)

health12mis 1=missing general health 12 months after assignment

trainy1 1=enrolled in education and/or vocational training in the first year after assignment, 0=no education or training in the first year after assignment

trainy2 1=enrolled in education and/or vocational training in the second year after assignment, 0=no education or training in the second year after assignment

pworky2 proportion of weeks employed in second year after assignment

pworky3 proportion of weeks employed in third year after assignment

pworky4 proportion of weeks employed in fourth year after assignment

earny2 weekly earnings in second year after assignment

earny3 weekly earnings in third year after assignment

earny4 weekly earnings in fourth year after assignment

health30 general health 30 months after assignment from 1 (excellent) to 4 (poor)

health48 general health 48 months after assignment from 1 (excellent) to 4 (poor)

References

Schochet, P. Z., Burghardt, J., Glazerman, S. (2001): "National Job Corps study: The impacts of Job Corps on participants' employment and related outcomes", Mathematica Policy Research, Washington, DC.

Examples

```
## Not run:
data(JC)
# Dynamic treatment effect evaluation of training in 1st and 2nd year
# define covariates at assignment (x0) and after one year (x1)
x0=JC[,2:29]; x1=JC[,30:36]
# define treatment (training) in first year (d1) and second year (d2)
d1=JC[,37]; d2=JC[,38]
# define outcome (weekly earnings in fourth year after assignment)
y2=JC[,44]
# assess dynamic treatment effects (training in 1st+2nd year vs. no training)
output=dyntreatDML(y2=y2, d1=d1, d2=d2, x0=x0, x1=x1)
cat("dynamic ATE: ",round(c(output$effect),3),", standard error: ",
    round(c(output$se),3), ", p-value: ",round(c(output$pval),3))
## End(Not run)
```

`labormarket` *Temporary Work Agency (TWA) Assignments and Permanent Employment in Sicily*

Description

A dataset containing 1120 individuals from Sicily, an Italian region in which a labor market intervention through a temporary work agency (TWA) was offered. It includes 229 individuals assigned to TWA jobs in the first 6 months of 2001 and 891 controls aged 18-40, who were in the labor force but not in stable employment on 1 January 2001, and who were not on a TWA assignment during the first half of that year.

Usage

```
labormarket
```

Format

A data frame with 1120 rows and 32 variables:

age age at time 2

blu1 blue-collar at time 1

child number of children at time 1

ct dummy Catania

dist distance from nearest agency

emp1 employed at time 1

hour1 weekly hours of work at time 1

hour3 weekly hours of work at time 3

id identification code

italy italian nationality

male male gender

me dummy Messina

nofl1 out of labor force at time 1

nyu1 fraction of school-to-work without employment

out3 employed at time 3

pa dummy Palermo

single non married

tp dummy Trapani

train1 received professional training before treatment

treat temp at time 2 - treatment status

unemp1 unemployed at time 1

wage1 monthly wage at time 1

wage2 monthly wage at time 2

wage3 monthly wage at time 3

yu1 years without employment at time 1

white1 white collar at time 1

unemp3 unemployed at time 3

nofl3 out of labor force at time 3

edu0 low education

edu1 medium education

edu2 high education

edu education level

References

Ichino, A., Mealli, F., and Nannicini, T. (2008): "From Temporary Help Jobs to Permanent Employment: What Can We Learn from Matching Estimators and their Sensitivity?", Journal of Applied Econometrics, 23(3), 305-327.

`lateweight` *Local average treatment effect estimation based on inverse probability weighting*

Description

Instrumental variable-based evaluation of local average treatment effects using weighting by the inverse of the instrument propensity score.

Usage

```
lateweight(
  y,
  d,
  z,
  x,
  LATT = FALSE,
  trim = 0.05,
  logit = FALSE,
  boot = 1999,
  cluster = NULL
)
```

Arguments

`y`	Dependent variable, must not contain missings.
`d`	Treatment, must be binary (either 1 or 0), must not contain missings.
`z`	Instrument for the endogenous treatment, must be binary (either 1 or 0), must not contain missings.
`x`	Confounders of the instrument and outcome, must not contain missings.

LATT	If FALSE, the local average treatment effect (LATE) among compliers (whose treatment reacts to the instrument) is estimated. If TRUE, the local average treatment effect on the treated compliers (LATT) is estimated. Default is FALSE.
trim	Trimming rule for discarding observations with extreme propensity scores. If LATT=FALSE, observations with Pr(Z=1|X)<trim or Pr(Z=1|X)>(1-trim) are dropped. If LATT=TRUE, observations with Pr(Z=1|X)>(1-trim) are dropped. Default is 0.05.
logit	If FALSE, probit regression is used for propensity score estimation. If TRUE, logit regression is used. Default is FALSE.
boot	Number of bootstrap replications for estimating standard errors. Default is 1999.
cluster	A cluster ID for block or cluster bootstrapping when units are clustered rather than iid. Must be numerical. Default is NULL (standard bootstrap without clustering).

Details

Estimation of local average treatment effects of a binary endogenous treatment based on a binary instrument that is conditionally valid, implying that all confounders of the instrument and the outcome are observed. Units are weighted by the inverse of their conditional instrument propensities given the observed confounders, which are estimated by probit or logit regression. Standard errors are obtained by bootstrapping the effect.

Value

A lateweight object contains 10 components, effect, se.effect, pval.effect, first, se.first, pval.first, ITT, se.ITT, pval.ITT, and ntrimmed:

effect: local average treatment effect (LATE) among compliers if LATT=FALSE or the local average treatment effect on treated compliers (LATT) if LATT=TRUE.

se.effect: bootstrap-based standard error of the effect.

pval.effect: p-value of the effect.

first: first stage estimate of the complier share if LATT=FALSE or the first stage estimate among treated if LATT=TRUE.

se.first: bootstrap-based standard error of the first stage effect.

pval.first: p-value of the first stage effect.

ITT: intention to treat effect (ITT) of z on y if LATT=FALSE or the ITT among treated if LATT=TRUE.

se.ITT: bootstrap-based standard error of the ITT.

pval.ITT: p-value of the ITT.

ntrimmed: number of discarded (trimmed) observations due to extreme propensity score values.

References

Frölich, M. (2007): "Nonparametric IV estimation of local average treatment effects with covariates", Journal of Econometrics, 139, 35-75.

Examples

```
# A little example with simulated data (10000 observations)
## Not run:
n=10000
u=rnorm(n)
x=rnorm(n)
z=(0.25*x+rnorm(n)>0)*1
d=(z+0.25*x+0.25*u+rnorm(n)>0.5)*1
y=0.5*d+0.25*x+u
# The true LATE is equal to 0.5
output=lateweight(y=y,d=d,z=z,x=x,trim=0.05,LATT=FALSE,logit=TRUE,boot=19)
cat("LATE: ",round(c(output$effect),3),", standard error: ",
             round(c(output$se.effect),3), ", p-value: ",
             round(c(output$pval.effect),3))
output$ntrimmed
## End(Not run)
```

---

medDML *Causal mediation analysis with double machine learning*

Description

Causal mediation analysis (evaluation of natural direct and indirect effects) for a binary treatment and one or several mediators using double machine learning to control for confounders based on (doubly robust) efficient score functions for potential outcomes.

Usage

```
medDML(
  y,
  d,
  m,
  x,
  k = 3,
  trim = 0.05,
  multmed = TRUE,
  fewsplits = FALSE,
  normalized = TRUE,
  MLmethod = "lasso",
  debiasfirst = TRUE
)
```

Arguments

y	Dependent variable, must not contain missings.
d	Treatment, must be binary (either 1 or 0), must not contain missings.

m	Mediator, must not contain missings. May be a scalar or a vector of binary, categorical, or continuous variables if multmed is TRUE. Must be a binary scalar if multmed is FALSE.
x	(Potential) pre-treatment confounders of the treatment, mediator, and/or outcome, must not contain missings.
k	Number of folds in k-fold cross-fitting if multmed is FALSE. k-1 folds are used for estimating the model parameters of the treatment, mediator, and outcome equations and one fold is used for predicting the efficient score functions. The roles of the folds are swapped. Default for k is 3. If multmed is TRUE, then 3-fold cross-valdiation is used, irrespective of the number provided in k (i.e. k is ignored if multmed is TRUE).
trim	Trimming rule for discarding observations with extreme conditional treatment or mediator probabilities (or products thereof). Observations with (products of) conditional probabilities that are smaller than trim in any denominator of the potential outcomes are dropped. Default is 0.05.
multmed	If set to TRUE, a representation of direct and indirect effects that avoids conditional mediator densities/probabilities is used, see Farbmacher, Huber, Langen, and Spindler (2022). This method can incorporate multiple and/or continuous mediators. If multmed is FALSE, the representation of Tchetgen Tchetgen and Shpitser (2012) is used, which involves mediator densities. In this case, the mediator must be a binary scalar. Default of multimed is TRUE.
fewsplits	If set to TRUE, the same training data are used for estimating nested models of nuisance parameters, i.e. E[Y|D=d,M,X] and E[E[Y|D=d,M,X]|D=1-d,X]. If fewsplits is FALSE, the training data are split for the sequential estimation of nested models E[Y|D=d,M,X] and E[E[Y|D=d,M,X]|D=1-d,X]. This parameter is only relevant if multmed is TRUE. Default of fewsplits is FALSE.
normalized	If set to TRUE, then the inverse probability-based weights are normalized such that they add up to 1 within treatment groups. Default is TRUE.
MLmethod	Machine learning method for estimating the nuisance parameters based on the hdm package (for lasso) or the SuperLearner package (for any other machine learner). Must be either "lasso" (default) for lasso estimation, "randomforest" for random forests, "xgboost" for xg boosting, "svm" for support vector machines, "ensemble" for using an ensemble algorithm based on all previously mentioned machine learners, or "parametric" for linear or logit regression.
debiasfirst	If set to TRUE, the nested conditional mean E[E[Y|D=d,M,X]|D=1-d,X] (for d being 1 or 0) is estimated in debiased form, using reweighted residuals of E[Y|D=d,M,X], following Qi and Zhang (2026): "Quadratic robust methods for causal mediation analysis". This is only relevant if multmed is TRUE (when nested conditional means are estimated) and otherwise ignored. Default is TRUE.

Details

Estimation of causal mechanisms (natural direct and indirect effects) of a treatment under selection on observables, assuming that all confounders of the binary treatment and the mediator, the treatment and the outcome, or the mediator and the outcome are observed and not affected by the treatment. Estimation is based on the (doubly robust) efficient score functions for potential outcomes, see Tchetgen Tchetgen and Shpitser (2012) and Farbmacher, Huber, Langen, and Spindler

(2019), as well as on double machine learning with cross-fitting, see Chernozhukov et al (2018). To this end, one part of the data is used for estimating the model parameters of the treatment, mediator, and outcome equations based on post-lasso regression, using the `rlasso` and `rlassologit` functions (for conditional means and probabilities, respectively) of the `hdm` package with default settings. The other part of the data is used for predicting the efficient score functions. The roles of the data parts are swapped and the direct and indirect effects are estimated based on averaging the predicted efficient score functions in the total sample. Standard errors are based on asymptotic approximations using the estimated variance of the (estimated) efficient score functions.

Value

A medDML object contains two components, `results` and `ntrimmed`:

`results`: a 3X6 matrix containing the effect estimates in the first row ("effects"), standard errors in the second row ("se"), and p-values in the third row ("p-value"). The first column provides the total effect, namely the average treatment effect (ATE). The second and third columns provide the direct effects under treatment and control, respectively ("dir.treat", "dir.control"). The fourth and fifth columns provide the indirect effects under treatment and control, respectively ("indir.treat", "indir.control"). The sixth column provides the estimated mean under non-treatment ("Y(0,M(0))").

`ntrimmed`: number of discarded (trimmed) observations due to extreme conditional probabilities.

References

Chernozhukov, V., Chetverikov, D., Demirer, M., Duflo, E., Hansen, C., Newey, W., Robins, J. (2018): "Double/debiased machine learning for treatment and structural parameters", The Econometrics Journal, 21, C1-C68.

Farbmacher, H., Huber, M., Laffers, L., Langen, H., and Spindler, M. (2022): "Causal mediation analysis with double machine learning", The Econometrics Journal, 25, 277-300.

Qi, Z., and Zhang, Y. (2026): "Quadratic robust methods for causal mediation analysis", arXiv preprint 2601.22592.

Tchetgen Tchetgen, E. J., and Shpitser, I. (2012): "Semiparametric theory for causal mediation analysis: efficiency bounds, multiple robustness, and sensitivity analysis", The Annals of Statistics, 40, 1816-1845.

Tibshirani, R. (1996): "Regression shrinkage and selection via the lasso", Journal of the Royal Statistical Society: Series B, 58, 267-288.

Examples

```
# A little example with simulated data (10000 observations)
## Not run:
n=10000                           # sample size
p=100                             # number of covariates
s=2                               # number of covariates that are confounders
x=matrix(rnorm(n*p),ncol=p)       # covariate matrix
beta=c(rep(0.25,s), rep(0,p-s))   # coefficients determining degree of confounding
d=(x%*%beta+rnorm(n)>0)*1         # treatment equation
m=(x%*%beta+0.5*d+rnorm(n)>0)*1   # mediator equation
y=x%*%beta+0.5*d+m+rnorm(n)       # outcome equation
# The true direct effects are equal to 0.5, the indirect effects equal to 0.19
```

```
output=medDML(y=y,d=d,m=m,x=x)
round(output$results,3)
output$ntrimmed
## End(Not run)
```

`medlateweight` *Causal mediation analysis with instruments for treatment and mediator based on weighting*

Description

Causal mediation analysis (evaluation of natural direct and indirect effects) with instruments for a binary treatment and a continuous mediator based on weighting as suggested in Frölich and Huber (2017), Theorem 1.

Usage

```
medlateweight(
  y,
  d,
  m,
  zd,
  zm,
  x,
  trim = 0.1,
  csquared = FALSE,
  boot = 1999,
  cminobs = 40,
  bwreg = NULL,
  bwm = NULL,
  logit = FALSE,
  cluster = NULL
)
```

Arguments

`y`	Dependent variable, must not contain missings.
`d`	Treatment, must be binary (either 1 or 0), must not contain missings.
`m`	Mediator(s),must be a continuous scalar, must not contain missings.
`zd`	Instrument for the treatment, must be binary (either 1 or 0), must not contain missings.
`zm`	Instrument for the mediator, must contain at least one continuous element, may be a scalar or a vector, must not contain missings. If no user-specified bandwidth is provided for the regressors when estimating the conditional cumulative distribution function F(M|Z2,X), i.e. if `bwreg=NULL`, then `zm` must be exclusively numeric.

`x` Pre-treatment confounders, may be a scalar or a vector, must not contain missings. If no user-specified bandwidth is provided for the regressors when estimating the conditional cumulative distribution function F(M|Z2,X), i.e. if `bwreg=NULL`, then `x` must be exclusively numeric.

`trim` Trimming rule for discarding observations with extreme weights. Discards observations whose relative weight would exceed the value in `trim` in the estimation of any of the potential outcomes. Default is 0.1 (i.e. a maximum weight of 10 percent per observation).

`csquared` If TRUE, then not only the control function C, but also its square is used as regressor in any estimated function that conditions on C. Default is FALSE.

`boot` Number of bootstrap replications for estimating standard errors. Default is 1999.

`cminobs` Minimum number of observations to compute the control function C, see the numerator of equation (7) in Frölich and Huber (2017). A larger value increases boundary bias when estimating the control function for lower values of M, but reduces the variance. Default is 40, but should be adapted to sample size and the number of variables in Z2 and X.

`bwreg` Bandwidths for `zm` and `x` in the estimation of the conditional cumulative distribution function F(M|Z2,X) based on the np package by Hayfield and Racine (2008). The length of the numeric vector must correspond to the joint number of elements in `zm` and `x` and will be used both in the original sample for effect estimation and in bootstrap samples to compute standard errors. If set to `NULL`, then the rule of thumb is used for bandwidth calculation, see the np package for details. In the latter case, all elements in the regressors must be numeric. Default is `NULL`.

`bwm` Bandwidth for `m` in the estimation of the conditional cumulative distribution function F(M|Z2,X) based on the np package by Hayfield and Racine (2008). Must be scalar and will be used both in the original sample for effect estimation and in bootstrap samples to compute standard errors. If set to `NULL`, then the rule of thumb is used for bandwidth calculation, see the np package for details. Default is `NULL`.

`logit` If FALSE, probit regression is used for any propensity score estimation. If TRUE, logit regression is used. Default is FALSE.

`cluster` A cluster ID for block or cluster bootstrapping when units are clustered rather than iid. Must be numerical. Default is NULL (standard bootstrap without clustering).

Details

Estimation of causal mechanisms (natural direct and indirect effects) of a binary treatment among treatment compliers based on distinct instruments for the treatment and the mediator. The treatment and its instrument are assumed to be binary, while the mediator and its instrument are assumed to be continuous, see Theorem 1 in Frölich and Huber (2017). The instruments are assumed to be conditionally valid given a set of observed confounders. A control function is used to tackle mediator endogeneity. Standard errors are obtained by bootstrapping the effects.

Value

A medlateweight object contains two components, `results` and `ntrimmed`:

`results`: a 3x7 matrix containing the effect estimates in the first row ("effects"), standard errors in the second row ("se"), and p-values in the third row ("p-value"). The first column provides the total effect, namely the local average treatment effect (LATE) on the compliers. The second and third columns provide the direct effects under treatment and control, respectively ("dir.treat", "dir.control"). The fourth and fifth columns provide the indirect effects under treatment and control, respectively ("indir.treat", "indir.control"). The sixth and seventh columns provide the parametric direct and indirect effect estimates ("dir.para", "indir.para") without intercation terms, respectively. For the parametric estimates, probit or logit specifications are used for the treatment model and OLS specifications for the mediator and outcome models.

`ntrimmed`: number of discarded (trimmed) observations due to large weights.

References

Frölich, M. and Huber, M. (2017): "Direct and indirect treatment effects: Causal chains and mediation analysis with instrumental variables", Journal of the Royal Statistical Society Series B, 79, 1645–1666.

Examples

```
# A little example with simulated data (3000 observations)
## Not run:
n=3000; sigma=matrix(c(1,0.5,0.5,0.5,1,0.5,0.5,0.5,1),3,3)
e=(rmvnorm(n,rep(0,3),sigma))
x=rnorm(n)
zd=(0.5*x+rnorm(n)>0)*1
d=(-1+0.5*x+2*zd+e[,3]>0)
zm=0.5*x+rnorm(n)
m=(0.5*x+2*zm+0.5*d+e[,2])
y=0.5*x+d+m+e[,1]
# The true direct and indirect effects on compliers are equal to 1 and 0.5, respectively
medlateweight(y,d,m,zd,zm,x,trim=0.1,csquared=FALSE,boot=19,cminobs=40,
bwreg=NULL,bwm=NULL,logit=FALSE)
## End(Not run)
```

`medqteDML` *Natural direct and indirect quantile treatment effects with double machine learning*

Description

Estimation of natural direct and indirect quantile treatment effects (QTEs) of a binary treatment operating through a scalar mediator, based on double machine learning. The method inverts post-lasso-based double/debiased machine learning estimates of the cumulative distribution functions (cdfs) of the four potential outcomes `Y(1,M(1))`, `Y(1,M(0))`, `Y(0,M(1))`, `Y(0,M(0))`, computed from the efficient influence functions of Hsu, Huber, and Yen (2026), and provides pointwise and uniform confidence bands via a multiplier bootstrap.

Usage

```
medqteDML(
  y,
  d,
  m,
  x,
  tau = seq(0.2, 0.9, 0.1),
  a = NULL,
  kfold = 5,
  trim = 0.05,
  nboot = 1000,
  alpha = 0.05,
  q = 0.1
)
```

Arguments

y	Outcome/dependent variable, must be a numeric scalar variable, must not contain missings.
d	Treatment, must be binary (either 1 or 0), must not contain missings.
m	Mediator, must be a numeric scalar variable, must not contain missings.
x	(Potential) pre-treatment confounders of the treatment, mediator, and/or outcome, must not contain missings.
tau	Vector of quantile ranks at which the QTEs are estimated, each strictly between 0 and 1. Default is `seq(0.2,0.9,0.1)`.
a	Grid of values at which the cdfs of the potential outcomes are estimated before inversion into quantiles. Default is `NULL`, in which case a grid of at least 50 points is built from empirical quantiles of `y`, spanning somewhat beyond the range of `tau`: the potential-outcome distributions are shifted relative to the pooled distribution of `y`, so a grid tied only to `tau` itself is usually too narrow and forces the quantile inversion to extrapolate. A custom `a` can be supplied for more control, but should be finer and wider than `tau`; `medqteDML` warns if the estimated cdfs do not bracket the requested `tau`.
kfold	Number of folds in the cross-fitting procedure. Default is 5.
trim	Trimming threshold applied to the estimated propensity scores `P(D=1|X)` and `P(D=1|M,X)`, which are winsorized to `[trim, 1-trim]`. Default is 0.05.
nboot	Number of multiplier bootstrap replications used for the pointwise confidence intervals and uniform confidence bands. Default is 1000. The bootstrap reuses the estimated efficient influence functions rather than refitting the nuisance parameters, so increasing `nboot` is inexpensive relative to increasing `length(a)` or `kfold`.
alpha	Significance level for the confidence intervals and bands. Default is 0.05.
q	Lower quantile used to compute the rescaled quantile spread underlying the uniform confidence bands, see Chernozhukov, Fernandez-Val, and Melly (2013). Default is 0.1, a common choice in that literature.

Details

The nuisance parameters (propensity scores P(D=1|X) and P(D=1|M,X), and the conditional cdf P(Y<=a|D,M,X) estimated by distribution regression) are estimated by post-lasso, with covariates selected via a double/triple selection step (lasso selection from the D|X, D|M,X, and 1{Y<=a}|D,M,X models, unioned) at every grid point a and every fold, following the implementation underlying the empirical application in Hsu, Huber, and Yen (2026). The estimator itself is the K-fold cross-fitting estimator of the cdf of each potential outcome based on the (triply robust) efficient influence function, see Section 2 of Hsu, Huber, and Yen (2026) and, for the underlying regression imputation approach, Algorithm 2 in Farbmacher et al. (2022). Quantiles are obtained by inverting the estimated cdfs on the grid a, after a rearrangement step (Chernozhukov, Fernandez-Val, and Galichon 2010) that enforces monotonicity. Statistical inference is based on a multiplier bootstrap that reuses the estimated efficient influence functions, following Section 2.5 of Hsu, Huber, and Yen (2026); both pointwise confidence intervals (percentile/reflection method) and uniform confidence bands (based on the bootstrap Kolmogorov-Smirnov max-t-statistic and the rescaled quantile spread) are provided.

Value

A medqteDML object contains the following components:

effects: a list of five data frames, one per estimand: total (the total effect, Y(1,M(1)) minus Y(0,M(0))); dir.treat and dir.control (the direct effects with the mediator fixed at its value under treatment, M(1), or under non-treatment, M(0), respectively – NDQTE'} and \code{NDQTE} in Hsu, Huber, and Yen 2026); and \code{indir.treat} and \code{indir.control} (the indirect effects with the treatment fixed at \code{D=1} or \code{D=0}, respectively -- \code{NIQTE} and \code{NIQTE' in Hsu, Huber, and Yen 2026). Each data frame has columns tau (the quantile rank), effect (the point estimate), se (bootstrap standard deviation), pw.lower/pw.upper (pointwise confidence interval), and unif.lower/unif.upper (uniform confidence band).

cdf: a data frame with the estimated cdfs of the four potential outcomes across the grid a.

quantiles: a data frame with the estimated quantiles Q(1,M(1)), Q(1,M(0)), Q(0,M(1)), Q(0,M(0)) at each tau.

boot: a list with the raw multiplier bootstrap draws of the five QTEs (rows are tau, columns are bootstrap replications), for users who want to construct alternative confidence measures.

ntrimmed: number of observations whose estimated propensity score P(D=1|X) or P(D=1|M,X) was winsorized (capped to trim or 1-trim) in at least one fold or grid point.

References

Chernozhukov, V., Fernandez-Val, I., and Galichon, A. (2010): "Quantile and Probability Curves Without Crossing", Econometrica, 78, 1093-1125.

Chernozhukov, V., Fernandez-Val, I., and Melly, B. (2013): "Inference on Counterfactual Distributions", Econometrica, 81, 2205-2268.

Chernozhukov, V., Chetverikov, D., Demirer, M., Duflo, E., Hansen, C., Newey, W., Robins, J. (2018): "Double/debiased machine learning for treatment and structural parameters", The Econometrics Journal, 21, C1-C68.

Farbmacher, H., Huber, M., Laffers, L., Langen, H., and Spindler, M. (2022): "Causal mediation analysis with double machine learning", The Econometrics Journal, 25, 277-300.

Hsu, Y.-C., Huber, M., and Yen, Y.-M. (2026): "Estimation of Direct and Indirect Quantile Treatment Effects with Double Machine Learning", Journal of Business & Economic Statistics, DOI: 10.1080/07350015.2026.2654889.

Examples

```
# A little example with simulated data
## Not run:
n=5000                                              # sample size
p=10                                                # number of covariates
x=matrix(rnorm(n*p),ncol=p)                         # covariate matrix
d=rbinom(n,1,plogis(0.5*x[,1]-0.3*x[,2]))           # treatment equation
m=rbinom(n,1,plogis(-0.3+0.8*d+0.4*x[,1]-0.3*x[,2])) # mediator equation
y=2*d+1.5*m-0.4*x[,1]+0.4*x[,2]+rnorm(n)            # outcome equation
# Direct effect is 2 at every quantile; indirect effect is around 0.3
output=medqteDML(y=y,d=d,m=m,x=x,tau=c(0.3,0.5,0.7))
output$effects$dir.control
output$effects$indir.treat
output$effects$dir.treat
output$effects$indir.control
output$ntrimmed

## End(Not run)
```

`medweight` *Causal mediation analysis based on inverse probability weighting with optional sample selection correction.*

Description

Causal mediation analysis (evaluation of natural direct and indirect effects) based on weighting by the inverse of treatment propensity scores as suggested in Huber (2014) and Huber and Solovyeva (2018).

Usage

```
medweight(
  y,
  d,
  m,
  x,
  w = NULL,
  s = NULL,
  z = NULL,
  selpop = FALSE,
  ATET = FALSE,
  trim = 0.05,
  logit = FALSE,
  boot = 1999,
```

```
  cluster = NULL
)
```

Arguments

y	Dependent variable, must not contain missings.
d	Treatment, must be binary (either 1 or 0), must not contain missings.
m	Mediator(s), may be a scalar or a vector, must not contain missings.
x	Pre-treatment confounders of the treatment, mediator, and/or outcome, must not contain missings.
w	Post-treatment confounders of the mediator and the outcome. Default is NULL. Must not contain missings.
s	Optional selection indicator. Must be one if y is observed (non-missing) and zero if y is not observed (missing). Default is NULL, implying that y does not contain any missings. Is ignored if w is not NULL.
z	Optional instrumental variable(s) for selection s. If NULL, outcome selection based on observables (x,d,m) - known as "missing at random" - is assumed.
selpop	Only to be used if both s and z are defined. If TRUE, the effects are estimated for the selected subpopulation with s=1 only. If FALSE, the effects are estimated for the total population.
ATET	If FALSE, the average treatment effect (ATE) and the corresponding direct and indirect effects are estimated. If TRUE, the average treatment effect on the treated (ATET) and the corresponding direct and indirect effects are estimated. Default is FALSE.
trim	Trimming rule for discarding observations with extreme propensity scores. In the absence of post-treatment confounders (w=NULL), observations with Pr(D=1|M,X)<trim or Pr(D=1|M,X)>(1-trim) are dropped. In the presence of post-treatment confounders (w is defined), observations with Pr(D=1|M,W,X)<trim or Pr(D=1|M,W,X)>(1-trim) are dropped. Default is 0.05. If s is defined (only considered if w is NULL!) and z is NULL, observations with low selection propensity scores, Pr(S=1|D,M,X)<trim, are discarded, too. If s and z are defined, the treatment propensity scores to be trimmed change to Pr(D=1|M,X,Pr(S=1|D,X,Z)).
logit	If FALSE, probit regression is used for propensity score estimation. If TRUE, logit regression is used. Default is FALSE.
boot	Number of bootstrap replications for estimating standard errors. Default is 1999.
cluster	A cluster ID for block or cluster bootstrapping when units are clustered rather than iid. Must be numerical. Default is NULL (standard bootstrap without clustering).

Details

Estimation of causal mechanisms (natural direct and indirect effects) of a binary treatment under a selection on observables assumption assuming that all confounders of the treatment and the mediator, the treatment and the outcome, or the mediator and the outcome are observed. Units are weighted by the inverse of their conditional treatment propensities given the mediator and/or

observed confounders, which are estimated by probit or logit regression. The form of weighting depends on whether the observed confounders are exclusively pre-treatment (x), or also contain post-treatment confounders of the mediator and the outcome (w). In the latter case, only partial indirect effects (from d to m to y) can be estimated that exclude any causal paths from d to w to m to y, see the discussion in Huber (2014). Standard errors are obtained by bootstrapping the effects. In the absence of post-treatment confounders (such that w is NULL), defining s allows correcting for sample selection due to missing outcomes based on the inverse of the conditional selection probability. The latter might either be related to observables, which implies a missing at random assumption, or in addition also to unobservables, if an instrument for sample selection is available. Effects are then estimated for the total population, see Huber and Solovyeva (2018) for further details.

Value

A medweight object contains two components, results and ntrimmed:

results: a 3X5 matrix containing the effect estimates in the first row ("effects"), standard errors in the second row ("se"), and p-values in the third row ("p-value"). The first column provides the total effect, namely the average treatment effect (ATE) if ATET=FALSE or the average treatment effect on the treated (ATET) if ATET=TRUE. The second and third columns provide the direct effects under treatment and control, respectively ("dir.treat", "dir.control"). See equation (6) if w=NULL (no post-treatment confounders) and equation (13) if w is defined, respectively, in Huber (2014). If w=NULL, the fourth and fifth columns provide the indirect effects under treatment and control, respectively ("indir.treat", "indir.control"), see equation (7) in Huber (2014). If w is defined, the fourth and fifth columns provide the partial indirect effects under treatment and control, respectively ("par.in.treat", "par.in.control"), see equation (14) in Huber (2014).

ntrimmed: number of discarded (trimmed) observations due to extreme propensity score values.

References

Huber, M. (2014): "Identifying causal mechanisms (primarily) based on inverse probability weighting", Journal of Applied Econometrics, 29, 920-943.

Huber, M. and Solovyeva, A. (2018): "Direct and indirect effects under sample selection and outcome attrition ", SES working paper 496, University of Fribourg.

Examples

```
# A little example with simulated data (10000 observations)
## Not run:
n=10000
x=rnorm(n)
d=(0.25*x+rnorm(n)>0)*1
w=0.2*d+0.25*x+rnorm(n)
m=0.5*w+0.5*d+0.25*x+rnorm(n)
y=0.5*d+m+w+0.25*x+rnorm(n)
# The true direct and partial indirect effects are all equal to 0.5
output=medweight(y=y,d=d,m=m,x=x,w=w,trim=0.05,ATET=FALSE,logit=TRUE,boot=19)
round(output$results,3)
output$ntrimmed
## End(Not run)
```

`medweightcont` *Causal mediation analysis with a continuous treatment based on weighting by the inverse of generalized propensity scores*

Description

Causal mediation analysis (evaluation of natural direct and indirect effects) of a continuous treatment based on weighting by the inverse of generalized propensity scores as suggested in Hsu, Huber, Lee, and Lettry (2020).

Usage

```
medweightcont(
  y,
  d,
  m,
  x,
  d0,
  d1,
  ATET = FALSE,
  trim = 0.1,
  lognorm = FALSE,
  bw = NULL,
  boot = 1999,
  cluster = NULL
)
```

Arguments

`y`	Dependent variable, must not contain missings.
`d`	Continuous treatment, must not contain missings.
`m`	Mediator(s), may be a scalar or a vector, must not contain missings.
`x`	Pre-treatment confounders of the treatment, mediator, and/or outcome, must not contain missings.
`d0`	Value of `d` under non-treatment. Effects are based on pairwise comparisons, i.e. differences in potential outcomes evaluated at `d1` and `d0`.
`d1`	Value of `d` under treatment. Effects are based on pairwise comparisons, i.e. differences in potential outcomes evaluated at `d1` and `d0`.
`ATET`	If FALSE, the average treatment effect (ATE) and the corresponding direct and indirect effects are estimated. If TRUE, the average treatment effect on the treated (ATET) and the corresponding direct and indirect effects are estimated. Default is `FALSE`.
`trim`	Trimming rule for discarding observations with too large weights in the estimation of any mean potential outcome. That is, observations with a weight>`trim` are dropped from the sample. Default is a maximum weight of 0.1 (or 10 percent) per observation.

lognorm	If FALSE, a linear model with normally distributed errors is assumed for generalized propensity score estimation. If TRUE, a lognormal model is assumed. Default is FALSE.
bw	Bandwith for the second order Epanechnikov kernel functions of the treatment. If set to NULL, bandwidth computation is based on the rule of thumb for Epanechnikov kernels, determining the bandwidth as the standard deviation of the treatment times 2.34/(n^0.25), where n is the sample size. Default is NULL.
boot	Number of bootstrap replications for estimating standard errors. Default is 1999.
cluster	A cluster ID for block or cluster bootstrapping when units are clustered rather than iid. Must be numerical. Default is NULL (standard bootstrap without clustering).

Details

Estimation of causal mechanisms (natural direct and indirect effects) of a continuous treatment under a selection on observables assumption assuming that all confounders of the treatment and the mediator, the treatment and the outcome, or the mediator and the outcome are observed. Units are weighted by the inverse of their conditional treatment densities (known as generalized propensity scores) given the mediator and/or observed confounders, which are estimated by linear or loglinear regression. Standard errors are obtained by bootstrapping the effects.

Value

A medweightcont object contains two components, results and ntrimmed:

results: a 3X5 matrix containing the effect estimates in the first row ("effects"), standard errors in the second row ("se"), and p-values in the third row ("p-value"). The first column provides the total effect, namely the average treatment effect (ATE) if ATET=FALSE or the average treatment effect on the treated (ATET), i.e. those with D=d1, if ATET=TRUE. The second and third columns provide the direct effects under treatment and control, respectively ("dir.treat", "dir.control"). The fourth and fifth columns provide the indirect effects under treatment and control, respectively ("indir.treat", "indir.control").

ntrimmed: number of discarded (trimmed) observations due to extreme propensity score values.

References

Hsu, Y.-C., Huber, M., Lee, Y.-Y., Lettry, L. (2020): "Direct and indirect effects of continuous treatments based on generalized propensity score weighting", Journal of Applied Econometrics, 35, 814-840.

Examples

```
# A little example with simulated data (10000 observations)
## Not run:
n=10000
x=runif(n=n,min=-1,max=1)
d=0.25*x+runif(n=n,min=-2,max=2)
d=d-min(d)
m=0.5*d+0.25*x+runif(n=n,min=-2,max=2)
```

```
y=0.5*d+m+0.25*x+runif(n=n,min=-2,max=2)
# The true direct and indirect effects are all equal to 0.5
output=medweightcont(y,d,m,x,d0=2,d1=3,ATET=FALSE,trim=0.1,
       lognorm=FALSE,bw=NULL,boot=19)
round(output$results,3)
output$ntrimmed
## End(Not run)
```

`paneltestDML` *paneltestDML: Overidentification test for ATET estimation in panel data*

Description

This function applies an overidentification test to assess if unconfoundedness of the treatments and conditional common trends as imposed in differences-in-differences jointly hold in panel data when evaluating the average treatment effect on the treated (ATET).

Usage

```
paneltestDML(y1, y0, d, x, trim = 0.05, MLmethod = "lasso", k = 4)
```

Arguments

`y1`	Outcome variable in the post-treatment period. Should not contain missing values.
`y0`	Outcome variable in the pre-treatment period. Should not contain missing values.
`d`	Treatment group indicator (binary). Should not contain missing values.
`x`	Covariates to be controlled for. Should not contain missing values.
`trim`	Trimming threshold for discarding observations with too extreme treatment propensity scores (smaller than trim or larger than 1-trim). Default is 0.05.
`MLmethod`	Machine learning method for estimating nuisance parameters using the `SuperLearner` package. Must be one of `"lasso"` (default), `"randomforest"`, `"xgboost"`, `"svm"`, `"ensemble"`, or `"parametric"`.
`k`	Number of folds in k-fold cross-fitting. Default is 4.

Details

The test statistic corresponds to the difference between the ATETs that are based on two distinct doubly robust score functions, namely that under unconfoundedness and that based on difference-in-differences under conditional common trends. Estimation in panel data is based on double machine learning and the function supports different machine learning methods to estimate nuisance parameters (conditional mean outcomes and propensity scores) as well as cross-fitting to mitigate overfitting.

Value

A list with the following components:

est	Test statistic.
se	Standard error.
pval	P-value.
ntrimmed	Number of trimmed or dropped observations due to propensitiy scores below the threshold trim.
pscore.xy0	Propensity score under unconfoundedness.
pscore.x	Propensity score under conditional common trends.
ATETselobs	ATET based on the selection on observables/unconfoundedness assumption.
seATETselobs	Standard error of the ATET based on the selection on observables/unconfoundedness assumption.
ATETdid	ATET based on difference-in-differences invoking the conditional common trends assumption.
seATETdid	Standard error of the ATET based on difference-in-differences invoking the conditional common trends assumption.

References

Huber, M., and Oeß, E.-M. (2026): "A joint test of unconfoundedness and common trends", Journal of Applied Econometrics.

Examples

```
## Not run:
n=1000
x=data.frame(rnorm(n), rnorm(n))
d=1*(0.5*x[,1]+rnorm(n)>0)
y0=rnorm(n)
y1=0.5*x[,1]+y0+d+rnorm(n)
# report p-value (note that unconfoundedness and common trends hold jointly)
paneltestDML(y1=y1, y0=y0, d=d, x=x)$pval

## End(Not run)
```

RDDcovar *Sharp regression discontinuity design conditional on covariates*

Description

Nonparametric (kernel regression-based) sharp regression discontinuity controlling for covariates that are permitted to jointly affect the treatment assignment and the outcome at the threshold of the running variable, see Frölich and Huber (2019).

Usage

```
RDDcovar(
  y,
  z,
  x,
  boot = 1999,
  bw0 = NULL,
  bw1 = NULL,
  regtype = "ll",
  bwz = NULL
)
```

Arguments

y	Dependent variable, must not contain missings.
z	Running variable. Must be coded such that the treatment is zero for z being smaller than zero and one for z being larger than or equal to zero. Must not contain missings.
x	Covariates, must not contain missings.
boot	Number of bootstrap replications for estimating standard errors. Default is 1999.
bw0	Bandwidth for a kernel regression of y on z and x below the threshold (for treatment equal to zero), using the Epanechnikov kernel. Default is NULL, implying that the bandwidth is estimated by least squares cross-validation.
bw1	Bandwidth for a kernel regression of y on z and x above the threshold (for treatment equal to one), using the Epanechnikov kernel. Default is NULL, implying that the bandwidth is estimated by least squares cross-validation.
regtype	Defines the type of the kernel regressions of y on z and x below and above the threshold. Must either be set to "ll" for local linear regression or to "ll" for local constant regression. Default is "ll".
bwz	Bandwidth for the (Epanechnikov) kernel function on z. Default is NULL, implying that the bandwidth is estimated by least squares cross-validation.

Details

Sharp regression discontinuity design conditional on covariates to control for observed confounders jointly affecting the treatment assignment and outcome at the threshold of the running variable as discussed in Frölich and Huber (2019). This is implemented by running kernel regressions of the outcome on the running variable and the covariates separately above and below the threshold and by applying a kernel smoother to the running variable around the threshold. The procedure permits choosing kernel bandwidths by cross-validation, even though this does in general not yield the optimal bandwidths for treatment effect estimation (checking the robustness of the results by varying the bandwidths is therefore highly recommended). Standard errors are based on bootstrapping.

Value

effect: Estimated treatment effect at the threshold.

se: Bootstrap-based standard error of the effect estimate.

pvalue: P-value based on the t-statistic.

bw0: Bandwidth for kernel regression of y on z and x below the threshold (for treatment equal to zero).

bw1: Bandwidth for kernel regression of y on z and x above the threshold (for treatment equal to one).

bwz: Bandwidth for the kernel function on z.

References

Frölich, M. and Huber, M. (2019): "Including covariates in the regression discontinuity design", Journal of Business & Economic Statistics, 37, 736-748.

Examples

```
## Not run:
# load unemployment duration data
data(ubduration)
# run sharp RDD conditional on covariates with user-defined bandwidths
output=RDDcovar(y=ubduration[,1],z=ubduration[,2],x=ubduration[,c(-1,-2)],
 bw0=c(0.17, 1, 0.01, 0.05, 0.54, 70000, 0.12, 0.91, 100000),
 bw1=c(0.59, 0.65, 0.30, 0.06, 0.81, 0.04, 0.12, 0.76, 1.03),bwz=0.2,boot=19)
cat("RDD effect estimate: ",round(c(output$effect),3),", standard error: ",
 round(c(output$se),3), ", p-value: ", round(c(output$pvalue),3))
## End(Not run)
```

---

rkd *Swedish municipalities*

Description

A dataset containing information on Swedish municipalities in the years 1996-2004

Usage

```
rkd
```

Format

A data frame with 2511 rows and 53 variables:

code Municipality code

year Year

municipality Municipality name

pop_1 Population, lagged 1 year

pop Population

partpop06 Share of population aged 0-6

partpop7_15 Share of population aged 7-15

partpop80_ Share of population aged 80+

partforeign Share of foreign born population

costequalgrants Cost-equalizing grants

popchange_10y 10 year out-migration, lagged 2 years

pers_admin Personnel, administration

pers_child Personnel, child care

pers_school Personnel, schools

pers_elder Personnel, elderly care

pers_total Personnel, total (full time equivalents pers 1,000 capita)

pers_social Personnel, social welfare

pers_tech Personnel, technical services

expenditures_total Total per capita public expenditures

wage_admin Average montly wage, administration

wage_child Average monthly wage, child care

wage_school Average monthly wage, schools

wage_elder Average monthly wage, elderly care

wage_total Average monthly wage, total

wage_social Average monthly wage, social welfare

wage_tech Average monthly wage, technical services

pers_officials Personnel, high administrative officials

pers_assistants Personnel, administrative assistants

pers_priv_school Outsourced personnel, schools

pers_priv_elder Outsourced personnel, elderly care

pers_priv_social Outsourced personnel, social welfare

pers_priv_child Outsourced personnel, child care

migrationgrant round(abs((popchange_10y+2)*100)) if popchange_10y <= -2, 0 otherwise

migpop migrationgrant*pop_1

summigpop sum(migpop) by year

sumpop sum(pop_1) by year

migrationmean summigpop/sumpop

exp_total Annual expenditures on personnel in 100SEK/capita

exp_admin Annual expenditures on personnel in 100SEK/capita

exp_child Annual expenditures on personnel in 100SEK/capita

exp_school Annual expenditures on personnel in 100SEK/capita

exp_elder Annual expenditures on personnel in 100SEK/capita

exp_social Annual expenditures on personnel in 100SEK/capita

exp_tech Annual expenditures on personnel in 100SEK/capita

expshare_total exp_total*100/expenditures_total

expshare_admin exp_admin*100/expenditures_total

expshare_child exp_child*100/expenditures_total

expshare_school exp_school*100/expenditures_total

expshare_elder exp_elder*100/expenditures_total

expshare_social exp_social*100/expenditures_total

expshare_tech exp_tech*100/expenditures_total

outmigration -popchange_10y

forcing outmigration-2 (running variable with threshold normalized to zero)

References

Lundqvist, Heléne, Dahlberg, Matz and Mörk, Eva (2014): "Stimulating Local Public Employment: Do General Grants Work?" American Economic Journal: Economic Policy, 6 (1): 167-92.

Examples

```
## Not run:
require(rdrobust)                            # load rdrobust package
require(causalweight)                        # load causalweight package
data(rkd)                                    # load rkd data
attach(rkd)                                  # attach rkd data
Y=pers_total                                 # define outcome (total personnel)
R=forcing                                    # define running variable
D=costequalgrants                            # define treatment (grants)
results=rdrobust(y=Y, x=R, fuzzy=D, deriv=1)  # run fuzzy RKD
summary(results)
## End(Not run)                             # show results
```

`swissexper` *Correspondence test in Swiss apprenticeship market*

Description

A dataset related to a field experiment (correspondence test) in the Swiss apprenticeship market 2018/2019. The experiment investigated the effects of applicant gender and parental occupation in applications to apprenticeships on callback rates (invitations to interviews, assessment centers, or trial apprenticeships)

Usage

```
swissexper
```

Format

A data frame with 2928 rows and 18 variables:

city agglomeration of apprenticeship: 1=Bern,2=Zurich,3=Basel,6=Lausanne

foundatdate date when job add was found

employees (estimated) number of employees: 1=1-20; 2=21-50; 3=51-100; 4=101-250; 5=251-500; 6=501-1000; 7=1001+

sector 1=public sector; 2=trade/wholesale; 3=manufacturing/goods; 4=services

uniqueID ID of application

sendatdate date when application was sent

job_father treatment: father's occupation: 1=professor; 2=unskilled worker; 3=intermediate commercial; 4=intermediate technical

job_mother treatment: mother's occupation: 1= primary school teacher; 2=homemaker

tier skill tier of apprenticeship: 1=lower; 2=intermediate; 3=upper

hasmoved applicant moved from different city: 1=yes; 0=no

contgender gender of contact person in company: 0=unknown; 1=female; 2=male

letterback 1: letters sent from company to applicant were returned; 0: no issues with returned letters

outcome_invite outcome: invitation to interview, assessment center, or trial apprenticeship: 1=yes; 0=no

female_appl treatment: 1=female applicant; 0=male applicant

antidiscrpolicy 1=explicit antidiscrimination policy on company's website; 0=no explicit antidiscrimination policy

outcome_interest outcome: either invitation, or asking further questions, or keeping application for further consideration

gender_neutrality 0=gender neutral job type; 1=female dominated job type; 2=male dominated type

company_activity scope of company's activity: 0=local; 1=national; 2=international

References

Fernandes, A., Huber, M., and Plaza, C. (2019): "The Effects of Gender and Parental Occupation in the Apprenticeship Market: An Experimental Evaluation", SES working paper 506, University of Fribourg.

`testmedident` *Test for identification in causal mediation and dynamic treatment models*

Description

This function tests for identification in causal mediation and dynamic treatment models based on covariates and instrumental variables using machine learning methods.

Usage

```
testmedident(
  y,
  d,
  m = NULL,
  x,
  w = NULL,
  z1,
  z2 = NULL,
  testmediator = TRUE,
  seed = 123,
  MLmethod = "lasso",
  k = 3,
  zeta_sigma = min(0.5, 500/length(y))
)
```

Arguments

`y`	Outcome variable.
`d`	Treatment variable.
`m`	Mediator variable (optional).
`x`	Baseline covariates (prior to treatment assignment).
`w`	Post-treatment covariates (prior to mediator assignment, optional).
`z1`	Instrument for the treatment.
`z2`	Instrument for the mediator (optional).
`testmediator`	Logical indicating if the mediator should be used as dependent variable (in addition to outcome `y`) when testing if the effect of treatment `d` is identified. Default is TRUE.
`seed`	Random seed for sample splitting in cross-fitting. Default is 123.
`MLmethod`	Machine learning method for estimating conditional outcome/mediator means required for testing. Default is "lasso".
`k`	Number of cross-fitting folds. Default is 3.

`zeta_sigma` Tuning parameter defining the standard deviation of a random, mean zero, and normal variable that is added to the test statistic to avoid a degenerate distribution of test statistic under the null hypothesis. `zeta_sigma` gauges the trade-off between power and size of the test. Default is the minimum of 0.5 and 500/(# of observations).

Details

This function implements a hypothesis test for identifying causal effects in mediation and dynamic treatment models involving sequential assignment of treatment and mediator variables. The test jointly verifies the exogeneity/ignorability of treatment and mediator variables conditional on covariates and the validity of (distinct) instruments for the treatment and mediator (ignorability of instrument assignment and exclusion restriction). If the null hypothesis holds, dynamic and pathwise causal effects may be identified based on the sequential exogeneity/ignorability of the treatment and the mediator given the covariates. The function employs machine learning techniques to control for covariates in a data-driven manner.

Value

A list with the following components:

`teststat`	Test statistic.
`se`	Standard error of the test statistic.
`pval`	Two-sided p-value of the test.

References

Huber, M., Kloiber, K., and Lafférs, L. (2024): "Testing identification in mediation and dynamic treatment models", arXiv preprint 2406.13826.

Examples

```
## Not run:
# Example with simulated data in which null hypothesis holds
n=2000
x=rnorm(n)
z1=rnorm(n)
z2=rnorm(n)
d=1*(0.5*x+0.5*z1+rnorm(n)>0)      # Treatment equation
m=0.5*x+0.5*d+0.5*z2+rnorm(n)      # Mediator equation
y=0.5*x+d+0.5*m+rnorm(n)           # Outcome equation
# Run test and report p-value
testmedident(y=y, d=d, m=m, x=x, z1=z1, z2=z2)$pval

## End(Not run)
```

`treatcontDML` *Continuous Treatment Effect Estimation under Selection on Observables using Double Machine Learning*

Description

This function estimates the average treatment effect (ATE, default) or average treatment effect on the treated (ATET, optional) of a continuously distributed treatment under selection on observables (unconfoundedness), using double machine learning with kernel smoothing around the treatment values of interest.

Usage

```
treatcontDML(
  y,
  d,
  x,
  dtreat,
  dcontrol,
  ATET = FALSE,
  MLmethod = "lasso",
  psmethod = 1,
  trim = 0.1,
  lognorm = FALSE,
  bw = NULL,
  bwfactor = 0.7,
  cluster = NULL,
  k = 3
)
```

Arguments

`y`	Outcome variable. Should not contain missing values.
`d`	Treatment variable. Should be continuous and not contain missing values.
`x`	Covariates. Should not contain missing values.
`dtreat`	Value of the treatment for which the mean potential outcome is evaluated as the "treatment" level.
`dcontrol`	Value of the treatment for which the mean potential outcome is evaluated as the "control" level.
`ATET`	Logical. If `FALSE` (default), the average treatment effect (ATE) in the total population is estimated. If `TRUE`, the average treatment effect on units with treatment close to `dtreat` (ATET) is estimated instead.
`MLmethod`	Machine learning method for estimating nuisance parameters using the `SuperLearner` package. Must be one of `"lasso"` (default), `"randomforest"`, `"xgboost"`, `"svm"`, `"ensemble"`, or `"parametric"`.

psmethod	Method for computing generalized propensity scores. Set to 1 for estimating conditional treatment densities using the treatment as dependent variable (imposing the parametric assumption of a Gaussian treatment distribution), or 2 for using the treatment kernel weights as dependent variable (nonparametric). Default is 1.
trim	Trimming threshold (in percent) for discarding observations whose (normalized) weight would otherwise dominate the estimate. Default is 0.1.
lognorm	Logical indicating if log-normal transformation should be applied when estimating conditional treatment densities using the treatment as dependent variable (parametric). Default is FALSE.
bw	Bandwidth for kernel density estimation. Default is NULL, implying that the bandwidth is calculated based on the rule-of-thumb.
bwfactor	Factor by which the bandwidth is multiplied. Default is 0.7 (undersmoothing).
cluster	Optional clustering variable for calculating standard errors.
k	Number of folds in k-fold cross-fitting. Default is 3.

Details

Under ATET=FALSE (default), the function estimates the ATE using a doubly robust, kernel-weighted score for continuous treatments, following Kennedy, Ma, McHugh, and Small (2017) and Colangelo and Lee (2020): for each of `dtreat` and `dcontrol`, every observation contributes both an outcome-regression prediction at that dose and a kernel/generalized-propensity-score-weighted residual correction. Under ATET=TRUE, a different doubly robust function instead makes observations near `dcontrol` to match the covariate distribution of units with treatment near `dtreat`, targeting the effect on that treated subpopulation rather than the total population.

Value

A list with the following components:

`effect`: Estimate of the average treatment effect (ATE) or average treatment effect on the treated (ATET).

`se`: Standard error of the effect estimate.

`trimmed`: Number of discarded (trimmed) observations.

`pval`: P-value.

References

Chernozhukov, V., Chetverikov, D., Demirer, M., Duflo, E., Hansen, C., Newey, W., Robins, J. (2018): "Double/debiased machine learning for treatment and structural parameters", The Econometrics Journal, 21, C1-C68.

Kennedy, E. H., Ma, Z., McHugh, M. D., Small, D. S. (2017): "Non-parametric methods for doubly robust estimation of continuous treatment effects", Journal of the Royal Statistical Society Series B, 79, 1229-1245.

Colangelo, K., Lee, Y.-Y. (2020): "Double debiased machine learning nonparametric inference with continuous treatments", arXiv preprint 2004.03036.

Examples

```
## Not run:
# Example with simulated data
n=3000
x=0.8*rnorm(n)
d=x+rnorm(n)
y=2*d+x+rnorm(n)
# true effect is 2
results=treatcontDML(y=y, d=d, dtreat=1, dcontrol=0, x=x, MLmethod="lasso")
cat("ATE: ", round(results$effect, 3), ", Standard error: ", round(results$se, 3))

## End(Not run)
```

`treatDML` *Binary or multiple discrete treatment effect evaluation with double machine learning*

Description

Treatment effect estimation for assessing the average effects of discrete (multiple or binary) treatments. Combines estimation based on (doubly robust) efficient score functions with double machine learning to control for confounders in a data-driven way.

Usage

```
treatDML(
  y,
  d,
  x,
  s = NULL,
  dtreat = 1,
  dcontrol = 0,
  trim = 0.01,
  MLmethod = "lasso",
  k = 3,
  normalized = TRUE
)
```

Arguments

`y`	Dependent variable, must not contain missings.
`d`	Treatment variable, must be discrete, must not contain missings.
`x`	Covariates, must not contain missings.
`s`	Indicator function for defining a subpopulation for whom the treatment effect is estimated as a function of the subpopulation's distribution of `x`. Default is `NULL` (estimation of the average treatment effect in the total population).

dtreat	Value of the treatment in the treatment group. Default is 1.
dcontrol	Value of the treatment in the control group. Default is 0.
trim	Trimming rule for discarding observations with treatment propensity scores that are smaller than trim or larger than 1-trim (to avoid too small denominators in weighting by the inverse of the propensity scores). Default is 0.01.
MLmethod	Machine learning method for estimating the nuisance parameters based on the SuperLearner package. Must be either "lasso" (default) for lasso estimation, "randomforest" for random forests, "xgboost" for xg boosting, "svm" for support vector machines, "ensemble" for using an ensemble algorithm based on all previously mentioned machine learners, or "parametric" for linear or logit regression.
k	Number of folds in k-fold cross-fitting. Default is 3.
normalized	If set to TRUE, then the inverse probability-based weights are normalized such that they add up to 1 within treatment groups. Default is TRUE.

Details

Estimation of the causal effects of binary or multiple discrete treatments under conditional independence, assuming that confounders jointly affecting the treatment and the outcome can be controlled for by observed covariates. Estimation is based on the (doubly robust) efficient score functions for potential outcomes in combination with double machine learning with cross-fitting, see Chernozhukov et al (2018). To this end, one part of the data is used for estimating the model parameters of the treatment and outcome equations based machine learning. The other part of the data is used for predicting the efficient score functions. The roles of the data parts are swapped (using k-fold cross-fitting) and the average treatment effect is estimated based on averaging the predicted efficient score functions in the total sample. Standard errors are based on asymptotic approximations using the estimated variance of the (estimated) efficient score functions.

Value

A treatDML object contains eight components, effect, se, pval, ntrimmed, meantreat, meancontrol, pstreat, and pscontrol:

effect: estimate of the average treatment effect.

se: standard error of the effect.

pval: p-value of the effect estimate.

ntrimmed: number of discarded (trimmed) observations due to extreme propensity scores.

meantreat: Estimate of the mean potential outcome under treatment.

meancontrol: Estimate of the mean potential outcome under control.

pstreat: P-score estimates for treatment in treatment group.

pscontrol: P-score estimates for treatment in control group.

References

Chernozhukov, V., Chetverikov, D., Demirer, M., Duflo, E., Hansen, C., Newey, W., Robins, J. (2018): "Double/debiased machine learning for treatment and structural parameters", The Econometrics Journal, 21, C1-C68.

van der Laan, M., Polley, E., Hubbard, A. (2007): "Super Learner", Statistical Applications in Genetics and Molecular Biology, 6.

Examples

```
# A little example with simulated data (2000 observations)
## Not run:
n=2000                              # sample size
p=100                               # number of covariates
s=2                                 # number of covariates that are confounders
x=matrix(rnorm(n*p),ncol=p)         # covariate matrix
beta=c(rep(0.25,s), rep(0,p-s))     # coefficients determining degree of confounding
d=(x%*%beta+rnorm(n)>0)*1           # treatment equation
y=x%*%beta+0.5*d+rnorm(n)         # outcome equation
# The true ATE is equal to 0.5
output=treatDML(y,d,x)
cat("ATE: ",round(c(output$effect),3),", standard error: ",
    round(c(output$se),3), ", p-value: ",round(c(output$pval),3))
output$ntrimmed
## End(Not run)
```

`treatselDML` *Binary or multiple treatment effect evaluation with double machine learning under sample selection/outcome attrition*

Description

Average treatment effect (ATE) estimation for assessing the average effects of discrete (multiple or binary) treatments under sample selection/outcome attrition. Combines estimation based on Neyman-orthogonal score functions with double machine learning to control for confounders in a data-driven way.

Usage

```
treatselDML(
  y,
  d,
  x,
  s,
  z = NULL,
  selected = 0,
  dtreat = 1,
  dcontrol = 0,
```

```
  trim = 0.01,
  MLmethod = "lasso",
  k = 3,
  normalized = TRUE
)
```

Arguments

y	Dependent variable, may contain missings.
d	Treatment variable, must be discrete, must not contain missings.
x	Covariates, must not contain missings.
s	Selection indicator. Must be 1 if y is observed (non-missing) and 0 if y is not observed (missing).
z	Optional instrumental variable(s) for selection s. If NULL, outcome selection based on observables (x,d) - known as "missing at random" - is assumed. If z is defined, outcome selection based on unobservables - known as "non-ignorable missingness" - is assumed. Default is NULL.
selected	Must be 1 if ATE is to be estimated for the selected population without missing outcomes. Must be 0 if the ATE is to be estimated for the total population. Default is 0 (ATE for total population). This parameter is ignored if z is NULL (under MAR, the ATE in the total population is estimated).
dtreat	Value of the treatment in the treatment group. Default is 1.
dcontrol	Value of the treatment in the control group. Default is 0.
trim	Trimming rule for discarding observations with (products of) propensity scores that are smaller than trim (to avoid too small denominators in weighting by the inverse of the propensity scores). If selected is 0 (ATE estimation for the total population), observations with products of the treatment and selection propensity scores that are smaller than trim are discarded. If selected is 1 (ATE estimation for the subpopulation with observed outcomes), observations with treatment propensity scores smaller than trim are discarded. Default for trim is 0.01.
MLmethod	Machine learning method for estimating the nuisance parameters based on the SuperLearner package. Must be either "lasso" (default) for lasso estimation, "randomforest" for random forests, "xgboost" for xg boosting, "svm" for support vector machines, "ensemble" for using an ensemble algorithm based on all previously mentioned machine learners, or "parametric" for linear or logit regression.
k	Number of folds in k-fold cross-fitting, which must not be smaller than 3. Default is 3.
normalized	If set to TRUE, then the inverse probability-based weights are normalized such that they add up to 1 within treatment groups. Default is TRUE.

Details

Estimation of the causal effects of binary or multiple discrete treatments under conditional independence, assuming that confounders jointly affecting the treatment and the outcome can be controlled

for by observed covariates, and sample selection/outcome attrition. The latter might either be related to observables, which implies a missing at random assumption, or in addition also to unobservables, if an instrument for sample selection is available. Estimation is based on Neyman-orthogonal score functions for potential outcomes in combination with double machine learning with cross-fitting, see Chernozhukov et al (2018). To this end, one part of the data is used for estimating the model parameters of the treatment and outcome equations based machine learning. The other part of the data is used for predicting the efficient score functions. The roles of the data parts are swapped (using k-fold cross-fitting) and the average treatment effect is estimated based on averaging the predicted efficient score functions in the total sample. Standard errors are based on asymptotic approximations using the estimated variance of the (estimated) efficient score functions.

Value

A `treatDML` object contains eight components, `effect`, `se`, `pval`, `ntrimmed`, `meantreat`, `meancontrol`, `pstreat`, and `pscontrol`:

`effect`: estimate of the average treatment effect.

`se`: standard error of the effect.

`pval`: p-value of the effect estimate.

`ntrimmed`: number of discarded (trimmed) observations due to extreme propensity scores.

`meantreat`: Estimate of the mean potential outcome under treatment.

`meancontrol`: Estimate of the mean potential outcome under control.

`pstreat`: P-score estimates for treatment in treatment group.

`pscontrol`: P-score estimates for treatment in control group.

References

Bia, M., Huber, M., Laffers, L. (2024): "Double machine learning for sample selection models", Journal of Business & Economic Statistics, 42, 958-969.

Chernozhukov, V., Chetverikov, D., Demirer, M., Duflo, E., Hansen, C., Newey, W., Robins, J. (2018): "Double/debiased machine learning for treatment and structural parameters", The Econometrics Journal, 21, C1-C68.

van der Laan, M., Polley, E., Hubbard, A. (2007): "Super Learner", Statistical Applications in Genetics and Molecular Biology, 6.

Examples

```
# A little example with simulated data (2000 observations)
## Not run:
n=2000                            # sample size
p=100                             # number of covariates
s=2                               # number of covariates that are confounders
sigma=matrix(c(1,0.5,0.5,1),2,2)
e=(2*rmvnorm(n,rep(0,2),sigma))
x=matrix(rnorm(n*p),ncol=p)       # covariate matrix
beta=c(rep(0.25,s), rep(0,p-s))   # coefficients determining degree of confounding
d=(x%*%beta+rnorm(n)>0)*1         # treatment equation
z=rnorm(n)
```

```
s=(x%*%beta+0.25*d+z+e[,1]>0)*1   # selection equation
y=x%*%beta+0.5*d+e[,2]            # outcome equation
 y[s==0]=0
# The true ATE is equal to 0.5
output=treatselDML(y,d,x,s,z)
cat("ATE: ",round(c(output$effect),3),", standard error: ",
    round(c(output$se),3), ", p-value: ",round(c(output$pval),3))
output$ntrimmed
## End(Not run)
```

`treatweight` *Treatment evaluation based on inverse probability weighting with optional sample selection correction.*

Description

Treatment evaluation based on inverse probability weighting with optional sample selection correction.

Usage

```
treatweight(
  y,
  d,
  x,
  s = NULL,
  z = NULL,
  selpop = FALSE,
  ATET = FALSE,
  trim = 0.05,
  logit = FALSE,
  boot = 1999,
  cluster = NULL
)
```

Arguments

y	Dependent variable.
d	Treatment, must be binary (either 1 or 0), must not contain missings.
x	Confounders of the treatment and outcome, must not contain missings.
s	Selection indicator. Must be one if y is observed (non-missing) and zero if y is not observed (missing). Default is NULL, implying that y does not contain any missings.
z	Optional instrumental variable(s) for selection s. If NULL, outcome selection based on observables (x,d) - known as "missing at random" - is assumed. If z is defined, outcome selection based on unobservables - known as "non-ignorable missingness" - is assumed. Default is NULL. If s is NULL, z is ignored.

selpop	Only to be used if both s and z are defined. If TRUE, the effect is estimated for the selected subpopulation with s=1 only. If FALSE, the effect is estimated for the total population. (note that this relies on somewhat stronger statistical assumptions). Default is FALSE. If s or z is NULL, selpop is ignored.
ATET	If FALSE, the average treatment effect (ATE) is estimated. If TRUE, the average treatment effect on the treated (ATET) is estimated. Default is FALSE.
trim	Trimming rule for discarding observations with extreme propensity scores. If ATET=FALSE, observations with Pr(D=1|X)<trim or Pr(D=1|X)>(1-trim) are dropped. If ATET=TRUE, observations with Pr(D=1|X)>(1-trim) are dropped. If s is defined and z is NULL, observations with extremely low selection propensity scores, Pr(S=1|D,X)<trim, are discarded, too. If s and z are defined, the treatment propensity scores to be trimmed change to Pr(D=1|X,Pr(S=1|D,X,Z)). If in addition selpop is FALSE, observation with Pr(S=1|D,X,Z)<trim are discarded, too. Default for trim is 0.05.
logit	If FALSE, probit regression is used for propensity score estimation. If TRUE, logit regression is used. Default is FALSE.
boot	Number of bootstrap replications for estimating standard errors. Default is 1999.
cluster	A cluster ID for block or cluster bootstrapping when units are clustered rather than iid. Must be numerical. Default is NULL (standard bootstrap without clustering).

Details

Estimation of treatment effects of a binary treatment under a selection on observables assumption assuming that all confounders of the treatment and the outcome are observed. Units are weighted by the inverse of their conditional treatment propensities given the observed confounders, which are estimated by probit or logit regression. Standard errors are obtained by bootstrapping the effect. If s is defined, the procedure allows correcting for sample selection due to missing outcomes based on the inverse of the conditional selection probability. The latter might either be related to observables, which implies a missing at random assumption, or in addition also to unobservables, if an instrument for sample selection is available. See Huber (2012, 2014) for further details.

Value

A treatweight object contains six components: effect, se, pval, y1, y0, and ntrimmed.

effect: average treatment effect (ATE) if ATET=FALSE or the average treatment effect on the treated (ATET) if ATET=TRUE.

se: bootstrap-based standard error of the effect.

pval: p-value of the effect.

y1: mean potential outcome under treatment.

y0: mean potential outcome under control.

ntrimmed: number of discarded (trimmed) observations due to extreme propensity score values.

References

Horvitz, D. G., and Thompson, D. J. (1952): "A generalization of sampling without replacement from a finite universe", Journal of the American Statistical Association, 47, 663–685.

Huber, M. (2012): "Identification of average treatment effects in social experiments under alternative forms of attrition", Journal of Educational and Behavioral Statistics, 37, 443-474.

Huber, M. (2014): "Treatment evaluation in the presence of sample selection", Econometric Reviews, 33, 869-905.

Examples

```
# A little example with simulated data (10000 observations)
## Not run:
n=10000
x=rnorm(n); d=(0.25*x+rnorm(n)>0)*1
y=0.5*d+0.25*x+rnorm(n)
# The true ATE is equal to 0.5
output=treatweight(y=y,d=d,x=x,trim=0.05,ATET=FALSE,logit=TRUE,boot=19)
cat("ATE: ",round(c(output$effect),3),", standard error: ",
    round(c(output$se),3), ", p-value: ",round(c(output$pval),3))
output$ntrimmed
## End(Not run)
# An example with non-random outcome selection and an instrument for selection
## Not run:
n=10000
sigma=matrix(c(1,0.6,0.6,1),2,2)
e=(2*rmvnorm(n,rep(0,2),sigma))
x=rnorm(n)
d=(0.5*x+rnorm(n)>0)*1
z=rnorm(n)
s=(0.25*x+0.25*d+0.5*z+e[,1]>0)*1
y=d+x+e[,2]; y[s==0]=0
# The true ATE is equal to 1
output=treatweight(y=y,d=d,x=x,s=s,z=z,selpop=FALSE,trim=0.05,ATET=FALSE,
       logit=TRUE,boot=19)
cat("ATE: ",round(c(output$effect),3),", standard error: ",
    round(c(output$se),3), ", p-value: ",round(c(output$pval),3))
output$ntrimmed
## End(Not run)
```

---

ubduration *Austrian unemployment duration data*

Description

A dataset containing unemployed females between 46 and 53 years old living in an Austrian region where an extension of the maximum duration of unemployment benefits (from 30 to 209 weeks under particular conditions) for job seekers aged 50 or older was introduced.

Usage

```
ubduration
```

Format

A data frame with 5659 rows and 10 variables:

y Outcome variable: unemployment duration of the jobseeker in weeks (registered at the unemployment office). Variable is numeric.

z Running variable: distance to the age threshold of 50 (implying an extended duration of unemployment benefits), measured in months divided by 12. Variable is numeric.

marrstatus Marital status: 0=other, 1=married, 2=single. Variable is a factor.

education Eductation: 0=low education, 1=medium education, 2=high education. Variable is ordered.

foreign Migrant status: 1=foreigner, 0=Austrian. Variable is a factor.

rr Replacement rate (of previous earnings by unemployment benefits). Variable is numeric.

lwageljob Log wage in last job. Variable is numeric.

experience Ratio of actual to potential work experience. Variable is numeric.

whitecollar 1=white collar worker, 0=blue collar worker. Variable is a factor.

industry Industry: 0=other, 1=agriculture, 2=utilities, 3=food, 4=textiles, 5=wood, 6=machines, 7=other manufacturing, 8=construction, 9=tourism, 10=traffic, 11=services. Variable is a factor.

References

Lalive, R. (2008): "How Do Extended Benefits Affect Unemployment Duration? A Regression Discontinuity Approach", Journal of Econometrics, 142, 785–806.

Frölich, M. and Huber, M. (2019): "Including covariates in the regression discontinuity design", Journal of Business & Economic Statistics, 37, 736-748.

Examples

```
## Not run:
# load unemployment duration data
data(ubduration)
# run sharp RDD conditional on covariates with user-defined bandwidths
output=RDDcovar(y=ubduration[,1],z=ubduration[,2],x=ubduration[,c(-1,-2)],
 bw0=c(0.17, 1, 0.01, 0.05, 0.54, 70000, 0.12, 0.91, 100000),
 bw1=c(0.59, 0.65, 0.30, 0.06, 0.81, 0.04, 0.12, 0.76, 1.03),bwz=0.2,boot=19)
cat("RDD effect estimate: ",round(c(output$effect),3),", standard error: ",
 round(c(output$se),3), ", p-value: ", round(c(output$pvalue),3))
## End(Not run)
```

wexpect *Wage expectations of students in Switzerland*

Description

A dataset containing information on wage expectations of 804 students at the University of Fribourg and the University of Applied Sciences in Bern in the year 2017.

Usage

```
wexpect
```

Format

A data frame with 804 rows and 39 variables:

wexpect1 wage expectations after finishing studies: 0=less than 3500 CHF gross per month; 1=3500-4000 CHF; 2=4000-4500 CHF;...; 15=10500-11000 CHF; 16=more than 11000 CHF

wexpect2 wage expectations 3 years after studying: 0=less than 3500 CHF gross per month; 1=3500-4000 CHF; 2=4000-4500 CHF;...; 15=10500-11000 CHF; 16=more than 11000 CHF

wexpect1othersex expected wage of other sex after finishing studies in percent of own expected wage

wexpect2othersex expected wage of other sex 3 years after studying in percent of own expected wage

male 1=male; 0=female

business 1=BA in business

econ 1=BA in economics

communi 1=BA in communication

businform 1=BA in business informatics

plansfull 1=plans working fulltime after studies

planseduc 1=plans obtaining further education (e.g. MA) after studies

sectorcons 1=planned sector: construction

sectortradesales 1=planned sector: trade and sales

sectortransware 1=planned sector: transport and warehousing

sectorhosprest 1=planned sector: hospitality and restaurant

sectorinfocom 1=planned sector: information and communication

sectorfininsur 1=planned sector: finance and insurance

sectorconsult 1=planned sector: consulting

sectoreduscience 1=planned sector: education and science

sectorhealthsocial 1=planned sector: health and social services

typegenstratman 1=planned job type: general or strategic management

typemarketing 1=planned job type: marketing

typecontrol 1=planned job type: controlling

typefinance 1=planned job type: finance

typesales 1=planned job type: sales

typetechengin 1=planned job type: technical/engineering

typehumanres 1=planned job type: human resources

posmanager 1=planned position: manager

age age in years

swiss 1=Swiss nationality

hassiblings 1=has one or more siblings

motherhighedu 1=mother has higher education

fatherhighedu 1=father has higher education

motherworkedfull 1=mother worked fulltime at respondent's age 4-6

motherworkedpart 1=mother worked parttime at respondent's age 4-6

matwellbeing self-assessed material wellbeing compared to average Swiss: 1=much worse; 2=worse; 3=as average Swiss; 4=better; 5=much better

homeowner 1=home ownership

treatmentinformation 1=if information on median wages in Switzerland was provided (randomized treatment)

treatmentorder 1=if order of questions on professional plans and personal information in survey has been reversed (randomized treatment), meaning that personal questions are asked first and professional ones later

References

Fernandes, A., Huber, M., and Vaccaro, G. (2020): "Gender Differences in Wage Expectations", arXiv preprint arXiv:2003.11496.

Examples

```
data(wexpect)
attach(wexpect)
# effect of randomized wage information (treatment) on wage expectations 3 years after
# studying (outcome)
treatweight(y=wexpect2,d=treatmentinformation,x=cbind(male,business,econ,communi,
businform,age,swiss,motherhighedu,fatherhighedu),boot=199)
# direct effect of gender (treatment) and indirect effect through choice of field of
# studies (mediator) on wage expectations (outcome)
medweight(y=wexpect2,d=male,m=cbind(business,econ,communi,businform),
x=cbind(treatmentinformation,age,swiss,motherhighedu,fatherhighedu),boot=199)
```

Index

∗ **datasets**
  c401k, 6
  coffeeleaflet, 10
  coupon, 12
  couponsretailer, 13
  creditcard, 15
  games, 34
  india, 39
  JC, 42
  labormarket, 44
  rkd, 64
  swissexper, 66
  ubduration, 79
  wexpect, 81

ATETDML, 2
attrlateweight, 4

c401k, 6
catehetDML, 8
coffeeleaflet, 10
coupon, 12
couponsretailer, 13
creditcard, 15

detectIV, 16
didcontDML, 19
didcontDMLpanel, 21
didDML, 24
didmedDML, 26
didmedDMLpanel, 28
didweight, 30
dyntreatDML, 31

games, 34

identificationDML, 36
india, 39
ivnr, 40

JC, 42

labormarket, 44
lateweight, 46

medDML, 48
medlateweight, 51
medqteDML, 53
medweight, 56
medweightcont, 59

paneltestDML, 61

RDDcovar, 62
rkd, 64

swissexper, 66

testmedident, 68
treatcontDML, 70
treatDML, 72
treatselDML, 74
treatweight, 77

ubduration, 79

wexpect, 81