Package ‘hdm’

February 14, 2024

Type Package
Title High-Dimensional Metrics
Version 0.3.2
Date 2024-02-09
Depends R (>= 3.0.0)
Description Implementation of selected high-dimensional statistical and econometric methods for estimation and inference. Efficient estimators and uniformly valid confidence intervals for various low-dimensional causal/structural parameters are provided which appear in high-dimensional approximately sparse models. Including functions for fitting heteroscedastic robust Lasso regressions with non-Gaussian errors and for instrumental variable (IV) and treatment effect estimation in a high-dimensional setting. Moreover, the methods enable valid post-selection inference and rely on a theoretically grounded, data-driven choice of the penalty.
License MIT + file LICENSE
LazyData TRUE
Imports MASS, glmnet, ggplot2, checkmate, Formula, methods
Suggests testthat, knitr, rmarkdown, formatR, xtable, mvtnorm, markdown
VignetteBuilder knitr
Encoding UTF-8
RoxygenNote 7.3.1
NeedsCompilation no
Author Martin Spindler [cre, aut], Victor Chernozhukov [aut], Christian Hansen [aut], Philipp Bach [ctb]
Maintainer Martin Spindler <martin.spindler@gmx.de>
Repository CRAN
Date/Publication 2024-02-14 21:20:02 UTC
**R topics documented:**

hdm-package .................................................... 2
AJR ................................................................. 3
BLP ................................................................. 4
coeff.rlassoEffects ............................................ 5
coeff.rlassoIV ..................................................... 6
coeff.rlassoIVselectX ........................................... 7
coeff.rlassoIVselectZ ........................................... 8
cps2012 ............................................................. 9
EminentDomain .................................................... 10
Growth Data ...................................................... 12
lambdaCalculation ............................................... 13
LassoShooting.fit ............................................... 14
pension ............................................................ 15
predict.rlassologit ............................................. 16
print.rlasso ....................................................... 17
print.rlassoEffects ............................................. 18
print.rlassoIV .................................................... 19
print.rlassoIVselectX ........................................... 19
print.rlassoIVselectZ ........................................... 20
print.rlassologitEffects ....................................... 21
print.rlassoTE ................................................... 22
print.tls .......................................................... 22
print_coef ......................................................... 23
p_adjust .......................................................... 24
rlasso ............................................................. 26
rlassoATE ......................................................... 29
rlassoEffects ..................................................... 32
rlassoIV ........................................................... 34
rlassoIVselectX .................................................. 36
rlassoIVselectZ .................................................. 37
rlassologit ........................................................ 39
rlassologitEffects .............................................. 41
summary.rlassoEffects ......................................... 43
tls ................................................................. 44

**Index**

<table>
<thead>
<tr>
<th>hdm-package</th>
<th>hdm: High-Dimensional Metrics</th>
</tr>
</thead>
</table>

**Description**

This package implements methods for estimation and inference in a high-dimensional setting.

**Details**
This package provides efficient estimators and uniformly valid confidence intervals for various low-dimensional causal/structural parameters appearing in high-dimensional approximately sparse models. The package includes functions for fitting heteroskedastic robust Lasso regressions with non-Gaussian errors and for instrumental variable (IV) and treatment effect estimation in a high-dimensional setting. Moreover, the methods enable valid post-selection inference. Moreover, a theoretically grounded, data-driven choice of the penalty level is provided.

**Author(s)**

**Maintainer:** Martin Spindler <martin.spindler@gmx.de>  
Authors:  
- Victor Chernozhukov  
- Christian Hansen  

Other contributors:  
- Philipp Bach <philipp.bach@uni-hamburg.de> [contributor]

---

**AJR data set**

**Description**

Dataset on settler mortality.

**Format**

- **Mort** Settler mortality  
- **logMort** logarithm of Mort  
- **Latitude** Latitude  
- **Latitude2** Latitude^2  
- **Africa** Africa  
- **Asia** Asia  
- **Namer** North America  
- **Samer** South America  
- **Neo** Neo-Europes  
- **GDP** GDP  
- **Exprop** Average protection against expropriation risk
Details

Data set was analysed in Acemoglu et al. (2001). A detailed description of the data can be found at https://economics.mit.edu/people/faculty/daron-acemoglu/data-archive

References


Examples

data(AJR)

<table>
<thead>
<tr>
<th>BLP</th>
<th>BLP data set</th>
</tr>
</thead>
</table>

Description

Automobile data set from the US.

Format

- **model.name**  model name  
- **model.id**  model id  
- **firm.id**  firm id  
- **cdid**  cdid  
- **id**  id  
- **price**  log price  
- **mpg**  miles per gallon  
- **mpd**  miles per dollar  
- **hpwt**  horse power per weight  
- **air**  air conditioning (binary variable)  
- **space**  size of the car  
- **share**  market share  
- **outshr**  share s0  
- **y**  outcome variable defined as log(share) - log(outshr)  
- **trend**  time trend  

Details

Data set was analysed in Berry, Levinsohn and Pakes (1995). The data stem from annual issues of the Automotive News Market Data Book. The data set inlcudes information on all models marketed during the the period beginning 1971 and ending in 1990 cotaining 2217 model/years from 997 distinct models. A detailed description is given in BLP (1995, 868–871). The internal function constructIV constructs instrumental variables along the lines described and used in BLP (1995).
References


Examples

data(BLP)

---

**Description**

Method to extract coefficients from objects of class `rlassoEffects`

**Usage**

```r
## S3 method for class 'rlassoEffects'
coef(
  object,
  complete = TRUE,
  selection.matrix = FALSE,
  include.targets = FALSE,
  ...)
```

**Arguments**

- `object`: an object of class `rlassoEffects`, usually a result of a call `rlassoEffect` or `rlassoEffects`.
- `complete`: general option of the function `coef`.
- `selection.matrix`: if TRUE, a selection matrix is returned that indicates the selected variables from each auxiliary regression. Default is set to FALSE.
- `include.targets`: if FALSE (by default) only the selected control variables are listed in the `selection.matrix`. If set to TRUE, the selection matrix will also indicate the selection of the target coefficients that are specified in the `rlassoEffects` call.
- `...`: further arguments passed to functions `coef` or `print`.

**Details**

Printing coefficients and selection matrix for S3 object `rlassoEffects`. Interpretation of entries in the selection matrix

- "-" indicates a target variable,
• "x" indicates that a variable has been selected with rlassoEffects (coefficient is different from zero),
• "." indicates that a variable has been de-selected with rlassoEffects (coefficient is zero).

Examples

```r
library(hdm)
set.seed(1)
n = 100 # sample size
p = 100 # number of variables
s = 7 # number of non-zero variables
X = matrix(rnorm(n*p), ncol=p)
colnames(X) <- paste("X", 1:p, sep="")
beta = c(rep(3,s), rep(0,p-s))
y = 1 + X%*%beta + rnorm(n)
data = data.frame(cbind(y,X))
colnames(data)[1] <- "y"
lasso.effect = rlassoEffects(X, y, index=c(1,2,3,50),
                          method = "double selection")
coef(lasso.effect) # standard use of coef() - without selection matrix
# with selection matrix
coef(lasso.effect, selection.matrix = TRUE)
# prettier output with print_coef (identical options as coef())
print_coef(lasso.effect, selection.matrix = TRUE)
```

**coef.rlassoIV**

**Coefficients from S3 objects rlassoIV**

**Description**

Method to extract coefficients from objects of class `rlassoIV`.

**Usage**

```r
## S3 method for class 'rlassoIV'
coef(object, complete = TRUE, selection.matrix = FALSE, ...)
```

**Arguments**

- `object` an object of class `rlassoIV`, usually a result of a call `rlassoIV` with options `select.X=TRUE` and `select.Z=TRUE`.
- `complete` general option of the function `coef`.
- `selection.matrix` if `TRUE`, a selection matrix is returned that indicates the selected variables from each first stage regression. Default is set to `FALSE`. See section on details for more information.
- `...` further arguments passed to function `coef`.
Details

Printing coefficients and selection matrix for S3 object rlassoIV. "x" indicates that a variable has been selected, i.e., the corresponding estimated coefficient is different from zero. The very last column collects all variables that have been selected in at least one of the lasso regressions represented in the selection.matrix. rlassoIV performs three lasso regression steps. A first stage lasso regression of the endogenous treatment variable \( d \) on the instruments \( z \) and exogenous covariates \( x \), a lasso regression of \( y \) on the exogenous variables \( x \), and a lasso regression of the instrumented treatment variable, i.e., a regression of the predicted values of \( d \), on controls \( x \).

Value

Coefficients obtained from rlassoIV by default. If option selection.matrix is TRUE, a list is returned with final coefficients, a matrix selection.matrix, and a matrix selection.matrixZ: selection.matrix contains the selection index for the lasso regression of \( y \) on \( x \) (first column) and the lasso regression of the predicted values of \( d \) on \( x \) together with the union of these indizes. selection.matrixZ contains the selection index from the first-stage lasso regression of \( d \) on \( z \) and \( x \).

Examples

```r
## Not run:
data(EminentDomain)
z <- EminentDomain$logGDP$z  # instruments
x <- EminentDomain$logGDP$x  # exogenous variables
y <- EminentDomain$logGDP$y  # outcome varialbe
d <- EminentDomain$logGDP$d  # treatment / endogenous variable
lasso.IV = rlassoIV(x=x, d=d, y=y, z=z, select.X=TRUE, select.Z=TRUE)
coef(lasso.IV)  # default behavior
coef(lasso.IV, selection.matrix = T)  # print selection matrix

## End(Not run)
```
Arguments

object  an object of class \texttt{rlassoIVselectX}, usually a result of a call \texttt{rlassoIVselectX} or \texttt{rlassoIV} with options \texttt{select.X=TRUE} and \texttt{select.Z=FALSE}.

complete  general option of the function \texttt{coef}.

selection.matrix  if TRUE, a selection matrix is returned that indicates the selected variables from each regression. Default is set to FALSE. See section on details for more information.

...  further arguments passed to functions \texttt{coef}.

Details

Printing coefficients and selection matrix for S3 object \texttt{rlassoIVselectX}. The first column of the selection matrix reports the selection index for the lasso regression of \( y \) on \( x \) in the specified \texttt{rlassoIVselectX} command. "x" indicates that a variable has been selected, i.e., the corresponding estimated coefficient is different from zero. The second column contains the selection index for the lasso regression of \( d \) on \( x \) and the remaining columns the index of selected variables \( x \) for the instruments \( z \). The very last column collects all variables that have been selected in at least one of the lasso regressions.

Examples

```r
## Not run:
library(hdm)
data(AJR); y = AJR$GDP; d = AJR$Exprop; z = AJR$logMort
x = model.matrix(~ -1 + (Latitude + Latitude2 + Africa + Asia + Namer + Samer)^2, data=AJR)
AJR.Xselect = rlassoIV(GDP ~ Exprop + (Latitude + Latitude2 + Africa + Asia + Namer + Samer)^2 | logMort + (Latitude + Latitude2 + Africa + Asia + Namer + Samer)^2, data=AJR, select.X=TRUE, select.Z=FALSE)
coef(AJR.Xselect)  # Default behavior
coef(AJR.Xselect, selection.matrix = TRUE)  # print selection matrix
```

```r
## End(Not run)
```
Arguments

object
   an object of class rlassoIVselectZ, usually a result of a call rlassoIVselectZ
   or rlassoIV with options select.X=FALSE and select.Z=TRUE.

complete
   general option of the function coef.

selection.matrix
   if TRUE, a selection matrix is returned that indicates the selected variables from
   each first stage regression. Default is set to FALSE. See section on details for
   more information.

... further arguments passed to functions coef.

Details

Printing coefficients and selection matrix for S3 object rlassoIVselectZ. The columns of the se-
lection matrix report the selection index for the first stage lasso regressions as specified rlassoIVselectZ
command, i.e., the selected variables for each of the endogenous variables. "x" indicates that a vari-
able has been selected, i.e., the corresponding estimated coefficient is different from zero. The very
last column collects all variables that have been selected in at least one of the lasso regressions.

Examples

## Not run:
lasso.IV.Z = rlassoIVselectZ(x=x, d=d, y=y, z=z)
data(EminentDomain)
z <- EminentDomain$logGDP$z # instruments
x <- EminentDomain$logGDP$x # exogenous variables
y <- EminentDomain$logGDP$y # outcome variable
d <- EminentDomain$logGDP$d # treatment / endogenous variable
lasso.IV.Z = rlassoIVselectZ(x=x, d=d, y=y, z=z)
coef(lasso.IV.Z) # Default behavior
coef(lasso.IV.Z, selection.matrix = T)
## End(Not run)

cps2012

cps2012 data set

Description

Census data from the US for the year 2012.

Format

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lnw</td>
<td>log of hourly wage (annual earnings / annual hours)</td>
</tr>
<tr>
<td>female</td>
<td>female indicator</td>
</tr>
<tr>
<td>married status</td>
<td>six indicators: widowed, divorced, separated, nevermarried, and married (omitted)</td>
</tr>
<tr>
<td>education attainment</td>
<td>six indicators: hsd08, hsd911, hsg, cg, ad, and sc (omitted)</td>
</tr>
</tbody>
</table>
**region indicators** four indicators: mw, so, we, and ne (omitted)

**potential experience** \( \max(0, \text{age} - \text{years of education} - 7) \): exp1, exp2 (divided by 100), exp3 (divided by 1000), exp4 (divided by 10000)

**weight** March Supplement sampling weight

**year** CPS year

**Details**

The CPS is a monthly U.S. household survey conducted jointly by the U.S. Census Bureau and the Bureau of Labor Statistics. The data comprise the year 2012. This data set was used in Mulligan and Rubinstein (2008). The sample comprises white non-hipanic, ages 25-54, working full time full year (35+ hours per week at least 50 weeks), exclude living in group quarters, self-employed, military, agricultural, and private household sector, allocated earning, inconsistent report on earnings and employment, missing data.

**References**


**Examples**

data(BLP)

---

**EminentDomain**

Eminent Domain data set

**Description**

Dataset on judicial eminent domain decisions.

**Format**

- \( y \) economic outcome variable
- \( x \) set of exogenous variables
- \( d \) eminent domain decisions
- \( z \) set of potential instruments

**Details**

Data set was analyzed in Belloni et al. (2012). They estimate the effect of judicial eminent domain decisions on economic outcomes with instrumental variables (IV) in a setting high a large set of potential IVs. A detailed description of the data can be found at [https://www.econometricsociety.org/publications/econometrica/2012/11/01/sparse-models-and-methods-optimal-instruments-application](https://www.econometricsociety.org/publications/econometrica/2012/11/01/sparse-models-and-methods-optimal-instruments-application). The data set contains four “sub-data sets” which differ mainly in the dependent variables: repeat-sales FHFA/OFHEO house price index for metro (FHFA) and non-metro (NM) area, the Case-Shiller home price index (CS), and state-level GDP from the Bureau of Economic Analysis - all
transformed with the logarithm. The structure of each subdata set is given above. In the data set
the following variables and name conventions are used: "numpanelskx_..." is the number of panels
with at least k members with the characteristic following the "_". The probability controls (names
start with "F_prob_") follow a similar naming convention and give the probability of observing a
panel with characteristic given following second "_" given the characteristics of the pool of judges
available to be assigned to the case.

Characteristics in the data for the control variables or instruments:

- **noreligion** judge reports no religious affiliation
- **jd_public** judge’s law degree is from a public university
- **dem** judge reports being a democrat
- **female** judge is female
- **nonwhite** judge is nonwhite (and not black)
- **black** judge is black
- **jewish** judge is Jewish
- **catholic** judge is Catholic
- **mainline** baseline religion
- **protestant** belongs to a protestant church
- **evangelical** belongs to an evangelical church
- **instate_ba** judge’s undergraduate degree was obtained within state
- **ba_public** judge’s undergraduate degree was obtained at a public university
- **elev** judge was elevated from a district court
- **year** year dummy (reference category is one year before the earliest year in the data set (excluded))
- **circuit** dummy for the circuit level (reference category excluded)
- **missing_cy_12** a dummy for whether there were no cases in that circuit-year
- **numcasecat_12** the number of takings appellate decisions

References

optimal instruments with an application to eminent domain. *Econometrica* 80 (6), 2369–2429.

Examples

data(EminentDomain)
Description

Data set of growth compiled by Barro Lee.

Format

Dataframe with the following variables:

- **outcome**: dependent variable: national growth rates in GDP per capita for the periods 1965-1975 and 1975-1985
- **x**: covariates which might influence growth

Details

The data set contains growth data of Barro-Lee. The Barro Lee data consists of a panel of 138 countries for the period 1960 to 1985. The dependent variable is national growth rates in GDP per capita for the periods 1965-1975 and 1975-1985. The growth rate in GDP over a period from \( t_1 \) to \( t_2 \) is commonly defined as \( \log(GDP_{t_1}/GDP_{t_2}) \). The number of covariates is \( p=62 \). The number of complete observations is 90.

Source

The full data set and further details can be found at [https://www2.nber.org/pub/barro.lee/](https://www2.nber.org/pub/barro.lee/) and [https://www.bristol.ac.uk//Depts//Economics//Growth//barlee.htm](https://www.bristol.ac.uk//Depts//Economics//Growth//barlee.htm).

References

R.J. Barro, J.W. Lee (1994). Data set for a panel of 139 countries. NBER.

Examples

data(GrowthData)
lambdaCalculation  

Function for Calculation of the penalty parameter

Description

This function implements different methods for calculation of the penalization parameter $\lambda$. Further details can be found under rlasso.

Usage

lambdaCalculation(
  penalty = list(homoscedastic = FALSE, X.dependent.lambda = FALSE, lambda.start = NULL, 
    c = 1.1, gamma = 0.1),
  y = NULL,
  x = NULL
)

Arguments

- **penalty** list with options for the calculation of the penalty.
  - c and gamma constants for the penalty with default $c=1.1$ and $\gamma=0.1$
  - homoscedastic logical, if homoscedastic errors are considered (default FALSE). Option none is described below.
  - X.dependent.lambda if independent or dependent design matrix $X$ is assumed for calculation of the parameter $\lambda$
  - numSim number of simulations for the X-dependent methods
  - lambda.start initial penalization value, compulsory for method "none"
- **y** residual which is used for calculation of the variance or the data-dependent loadings
- **x** matrix of regressor variables

Value

The functions returns a list with the penalty $\lambda$ which is the product of $\lambda_0$ and $\Upsilon_0$. $\Upsilon_0$ denotes either the variance (independent case) or the data-dependent loadings for the regressors. method gives the selected method for the calculation.


LassoShooting.fit  

Shooting Lasso

**Description**

Implementation of the Shooting Lasso (Fu, 1998) with variable dependent penalization weights.

**Usage**

```
LassoShooting.fit(
  x,
  y,
  lambda,
  control = list(maxIter = 1000, optTol = 10^(-5), zeroThreshold = 10^(-6)),
  XX = NULL,
  Xy = NULL,
  beta.start = NULL
)
```

**Arguments**

- **x**  
  matrix of regressor variables (n times p where n denotes the number of observations and p the number of regressors)

- **y**  
  dependent variable (vector or matrix)

- **lambda**  
  vector of length p of penalization parameters for each regressor

- **control**  
  list with control parameters: maxIter maximal number of iterations, optTol tolerance for parameter precision, zeroThreshold threshold applied to the estimated coefficients for numerical issues.

- **XX**  
  optional, precalculated matrix $t(X) \ast X$

- **Xy**  
  optional, precalculated matrix $t(X) \ast y$

- **beta.start**  
  start value for beta

**Details**

The function implements the Shooting Lasso (Fu, 1998) with variable dependent penalization. The arguments XX and Xy are optional and allow to use precalculated matrices which might improve performance.

**Value**

- **coefficients**  
  estimated coefficients by the Shooting Lasso Algorithm

- **coef.list**  
  matrix of coefficients from each iteration

- **num.it**  
  number of iterations run
References


---

**Pension 401(k) data set**

**Description**

Data set on financial wealth and 401(k) plan participation

**Format**

Dataframe with the following variables (amongst others):

- p401 participation in 401(k)
- e401 eligibility for 401(k)
- a401 401(k) assets
- tw total wealth (in US $)
- tfa financial assets (in US $)
- net_tfa net financial assets (in US $)
- nifa non-401k financial assets (in US $)
- net_nifa net non-401k financial assets
- net_n401 net non-401(k) assets (in US $)
- ira individual retirement account (IRA)
- inc income (in US $)
- age age
- fsize family size
- marr married
- pira participation in IRA
- db defined benefit pension
- hown home owner
- educ education (in years)
- male male
- twoearn two earners
- nohs, hs, smcol, col dummies for education: no high-school, high-school, some college, college
- hmort home mortage (in US $)
- hequity home equity (in US $)
- hval home value (in US $)
Details

The sample is drawn from the 1991 Survey of Income and Program Participation (SIPP) and consists of 9,915 observations. The observational units are household reference persons aged 25-64 and spouse if present. Households are included in the sample if at least one person is employed and no one is self-employed. The data set was analyzed in Chernozhukov and Hansen (2004) and Belloni et al. (2014) where further details can be found. They examine the effects of 401(k) plans on wealth using data from the Survey of Income and Program Participation using 401(k) eligibility as an instrument for 401(k) participation.

References


Examples

data(pension)

predict.rlassologit  Methods for S3 object rlassologit

Description

Objects of class rlassologit are constructed by rlassologit. print.rlassologit prints and displays some information about fitted rlassologit objects. summary.rlassologit summarizes information of a fitted rlassologit object. predict.rlassologit predicts values based on a rlassologit object. model.matrix.rlassologit constructs the model matrix of a lasso object.

Usage

```r
## S3 method for class 'rlassologit'
predict(object, newdata = NULL, type = "response", ...)

## S3 method for class 'rlassologit'
model.matrix(object, ...)

## S3 method for class 'rlassologit'
print(x, all = TRUE, digits = max(3L, getOption("digits") - 3L), ...)

## S3 method for class 'rlassologit'
summary(object, all = TRUE, digits = max(3L, getOption("digits") - 3L), ...)
```
print.rlasso

Arguments

- **object**: an object of class rlassologit
- **newdata**: new data set for prediction
- **type**: type of prediction required. The default (`'response'`) is on the scale of the response variable; the alternative `'link'` is on the scale of the linear predictors.
- **...**: arguments passed to the print function and other methods
- **x**: an object of class rlassologit
- **all**: logical, indicates if coefficients of all variables (TRUE) should be displayed or only the non-zero ones (FALSE)
- **digits**: significant digits in printout

Description

Objects of class rlasso are constructed by rlasso. print.rlasso prints and displays some information about fitted rlasso objects. summary.rlasso summarizes information of a fitted rlasso object. predict.rlasso predicts values based on a rlasso object. model.matrix.rlasso constructs the model matrix of a rlasso object.

Usage

```r
## S3 method for class 'rlasso'
print(x, all = TRUE, digits = max(3L, getOption("digits") - 3L), ...)

## S3 method for class 'rlasso'
summary(object, all = TRUE, digits = max(3L, getOption("digits") - 3L), ...)

## S3 method for class 'rlasso'
model.matrix(object, ...)

## S3 method for class 'rlasso'
predict(object, newdata = NULL, ...)
```

Arguments

- **x**: an object of class rlasso
- **all**: logical, indicates if coefficients of all variables (TRUE) should be displayed or only the non-zero ones (FALSE)
- **digits**: significant digits in printout
- **...**: arguments passed to the print function and other methods
- **object**: an object of class rlasso
- **newdata**: new data set for prediction. An optional data frame in which to look for variables with which to predict. If omitted, the fitted values are returned.
print.rlassoEffects  

Methods for S3 object rlassoEffects

Description

Objects of class rlassoEffects are constructed by rlassoEffects. print.rlassoEffects prints and displays some information about fitted rlassoEffect objects. summary.rlassoEffects summarizes information of a fitted rlassoEffect object and is described at summary.rlassoEffects. confint.rlassoEffects extracts the confidence intervals. plot.rlassoEffects plots the estimates with confidence intervals.

Usage

## S3 method for class 'rlassoEffects'
print(x, digits = max(3L, getOption("digits") - 3L), ...)

## S3 method for class 'rlassoEffects'
confint(object, parm, level = 0.95, joint = FALSE, ...)

## S3 method for class 'rlassoEffects'
plot(
  x,
  joint = FALSE,
  level = 0.95,
  main = "",
  xlab = "coef",
  ylab = "",
  xlim = NULL,
  ...
)

Arguments

x  an object of class rlassoEffects
digits significant digits in printout
... arguments passed to the print function and other methods.
object an object of class rlassoEffects
parm a specification of which parameters are to be given confidence intervals among the variables for which inference was done, either a vector of numbers or a vector of names. If missing, all parameters are considered.
level confidence level required
joint logical, if TRUE joint confidence intervals are calculated.
main an overall title for the plot
xlab a title for the x axis
print.rlassoIV

Description

Objects of class rlassoIV are constructed by rlassoIV. print.rlassoIV prints and displays some information about fitted rlassoIV objects. summary.rlassoIV summarizes information of a fitted rlassoIV object. confint.rlassoIV extracts the confidence intervals.

Usage

```r
## S3 method for class 'rlassoIV'
print(x, digits = max(3L, getOption("digits") - 3L), ...)

## S3 method for class 'rlassoIV'
summary(object, digits = max(3L, getOption("digits") - 3L), ...)

## S3 method for class 'rlassoIV'
confint(object, parm, level = 0.95, ...)
```

Arguments

- `x` an object of class rlassoIV
- `digits` significant digits in printout
- `...` arguments passed to the print function and other methods
- `object` An object of class rlassoIV
- `parm` a specification of which parameters are to be given confidence intervals, either a vector of numbers or a vector of names. If missing, all parameters are considered.
- `level` confidence level required.

print.rlassoIVselectX

Description

Objects of class rlassoIVselectX are constructed by rlassoIVselectX. print.rlassoIVselectX prints and displays some information about fitted rlassoIVselectX objects. summary.rlassoIVselectX summarizes information of a fitted rlassoIVselectX object. confint.rlassoIVselectX extracts the confidence intervals.
print.rlassoIVselectZ

Usage

## S3 method for class 'rlassoIVselectX'
print(x, digits = max(3L,getOption("digits") - 3L), ...)

## S3 method for class 'rlassoIVselectX'
summary(object, digits = max(3L,getOption("digits") - 3L), ...)

## S3 method for class 'rlassoIVselectX'
confint(object, parm, level = 0.95, ...)

Arguments

x an object of class rlassoIVselectX
digits significant digits in printout
... arguments passed to the print function and other methods
object an object of class rlassoIVselectX
parm a specification of which parameters are to be given confidence intervals, either a vector of numbers or a vector of names. If missing, all parameters are considered.
level the confidence level required.

Description

Objects of class rlassoIVselectZ are constructed by rlassoIVselectZ. print.rlassoIVselectZ prints and displays some information about fitted rlassoIVselectZ objects. summary.rlassoIVselectZ summarizes information of a fitted rlassoIVselectZ object. confint.rlassoIVselectZ extracts the confidence intervals.

Usage

## S3 method for class 'rlassoIVselectZ'
print(x, digits = max(3L,getOption("digits") - 3L), ...)

## S3 method for class 'rlassoIVselectZ'
summary(object, digits = max(3L,getOption("digits") - 3L), ...)

## S3 method for class 'rlassoIVselectZ'
confint(object, parm, level = 0.95, ...)
Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>an object of class rlassoIVselectZ</td>
</tr>
<tr>
<td>digits</td>
<td>significant digits in printout</td>
</tr>
<tr>
<td>...</td>
<td>arguments passed to the print function and other methods</td>
</tr>
<tr>
<td>object</td>
<td>an object of class rlassoIVselectZ</td>
</tr>
<tr>
<td>parm</td>
<td>a specification of which parameters are to be given confidence intervals, either a vector of numbers or a vector of names. If missing, all parameters are considered.</td>
</tr>
<tr>
<td>level</td>
<td>confidence level required.</td>
</tr>
</tbody>
</table>

Methods for S3 object rlassologitEffects

Description

Objects of class rlassologitEffects are constructed by rlassologitEffects or rlassologitEffect. print.rlassologitEffects prints and displays some information about fitted rlassologitEffect objects. summary.rlassologitEffects summarizes information of a fitted rlassologitEffects object. confint.rlassologitEffects extracts the confidence intervals.

Usage

```r
## S3 method for class 'rlassologitEffects'
print(x, digits = max(3L, getOption("digits") - 3L), ...)

## S3 method for class 'rlassologitEffects'
summary(object, digits = max(3L, getOption("digits") - 3L), ...)

## S3 method for class 'rlassologitEffects'
confint(object, parm, level = 0.95, joint = FALSE, ...)
```

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>an object of class rlassologitEffects</td>
</tr>
<tr>
<td>digits</td>
<td>number of significant digits in printout</td>
</tr>
<tr>
<td>...</td>
<td>arguments passed to the print function and other methods</td>
</tr>
<tr>
<td>object</td>
<td>an object of class rlassologitEffects</td>
</tr>
<tr>
<td>parm</td>
<td>a specification of which parameters are to be given confidence intervals, either a vector of numbers or a vector of names. If missing, all parameters are considered.</td>
</tr>
<tr>
<td>level</td>
<td>confidence level required.</td>
</tr>
<tr>
<td>joint</td>
<td>logical, if joint confidence intervals should be calculated</td>
</tr>
</tbody>
</table>
Methods for S3 object rlassoTE

Description

Objects of class rlassoTE are constructed by rlassoATE, rlassoATET, rlassoLATE, rlassoLATET. print.rlassoTE prints and displays some information about fitted rlassoTE objects. summary.rlassoTE summarizes information of a fitted rlassoTE object. confint.rlassoTE extracts the confidence intervals.

Usage

## S3 method for class 'rlassoTE'
print(x, digits = max(3L, getOption("digits") - 3L), ...)

## S3 method for class 'rlassoTE'
summary(object, digits = max(3L, getOption("digits") - 3L), ...)

## S3 method for class 'rlassoTE'
confint(object, parm, level = 0.95, ...)

Arguments

x an object of class rlassoTE
digits number of significant digits in printout
... arguments passed to the print function and other methods
object an object of class rlassoTE
parm a specification of which parameters are to be given confidence intervals, either a vector of numbers or a vector of names. If missing, all parameters are considered.
level confidence level required.

Methods for S3 object tsls

Description

Objects of class tsls are constructed by tsls. print.tsls prints and displays some information about fitted tsls objects. summary.tsls summarizes information of a fitted tsls object.
print_coef

Usage

## S3 method for class 'tsls'
print(x, digits = max(3L, getOption("digits") - 3L), ...)

## S3 method for class 'tsls'
summary(object, digits = max(3L, getOption("digits") - 3L), ...)

Arguments

x an object of class tsls
digits significant digits in printout
... arguments passed to the print function and other methods
object an object of class tsls

print_coef Printing coefficients from S3 objects rlassoEffects

Description

Printing coefficients for class rlassoEffects

Usage

print_coef(x, ...)

## S3 method for class 'rlassoEffects'
print_coef(
  x,
  complete = TRUE,
  selection.matrix = FALSE,
  include.targets = TRUE,
  ...
)

Arguments

x an object of class rlassoEffects, usually a result of a call rlassoEffect or rlassoEffects.
... further arguments passed to functions coef or print.
complete general option of the function coef.
selection.matrix if TRUE, a selection matrix is returned that indicates the selected variables from each auxiliary regression. Default is set to FALSE.
include.targets if FALSE (by default) only the selected control variables are listed in the selection.matrix. If set to TRUE, the selection matrix will also indicate the selection of the target coefficients that are specified in the rlassoEffects call.
Details

Printing coefficients and selection matrix for S3 object rlassoEffects

Examples

```r
library(hdm)
set.seed(1)
n = 100 # sample size
p = 100 # number of variables
s = 7 # number of non-zero variables
X = matrix(rnorm(n*p), ncol=p)
colnames(X) <- paste("X", 1:p, sep="")
beta = c(rep(3, s), rep(0, p-s))
y = 1 + X%*%beta + rnorm(n)
data = data.frame(cbind(y, X))
colnames(data)[1] <- "y"
lasso_effect = rlassoEffects(X, y, index=c(1,2,3,50),
method = "double selection")
# without target coefficient estimates
print_coef(lasso_effect, selection.matrix = TRUE)
# with target coefficient estimates
print_coef(lasso_effect, selection.matrix = TRUE, targets = TRUE)
```

Description

Multiple hypotheses testing adjustment of p-values from a high-dimensional linear model.

Usage

```r
p_adjust(x, ...)

## S3 method for class 'rlassoEffects'
p_adjust(x, method = "RW", B = 1000, ...)

## S3 method for class 'lm'
p_adjust(x, method = "RW", B = 1000, test.index = NULL, ...)
```

Arguments

- `x`: an object of S3 class rlassoEffects or lm.
- `...`: further arguments passed on to methods.
- `method`: the method of p-value adjustment for multiple testing. Romano-Wolf stepdown ('RW') is chosen by default.
- `B`: number of bootstrap repetitions (default 1000).
The `p_adjust` function adjusts p-values for multiple testing. It is used to adjust the p-values obtained from statistical tests to account for the increased risk of false positives when multiple hypotheses are tested. The function can be used for various types of models such as `rlassoEffects` and `lm`.

**Details**

Multiple testing adjustment is performed for S3 objects of class `rlassoEffects` and `lm`. Implemented methods for multiple testing adjustment are Romano-Wolf stepdown 'RW' (default) and the adjustment methods available in the `p.adjust` function of the `stats` package, including the Bonferroni, Bonferroni-Holm, and Benjamini-Hochberg corrections, see `p.adjust.methods`.

Objects of class `rlassoEffects` are constructed by `rlassoEffects`.

**Value**

A matrix with the estimated coefficients and the p-values that are adjusted according to the specified method.

**Methods (by class)**

- `p_adjust(rlassoEffects)`: `rlassoEffects`
- `p_adjust(lm)`: `lm`

**References**


**Examples**

```r
library(hdm);
s.set.seed(1)
n = 100 # sample size
p = 25 # number of variables
s = 3 # number of non-zero variables
X = matrix(rnorm(n*p), ncol=p)
colnames(X) <- paste("X", 1:p, sep="")
beta = c(rep(3,s), rep(0,p-s))
y = 1 + X%*%beta + rnorm(n)
data = data.frame(cbind(y,X))
colnames(data)[1] <- "y"
lasso.effect = rlassoEffects(X, y, index=c(1:20))
pvals.lasso.effect = p_adjust(lasso.effect, method = "RW", B = 1000)
ols = lm(y ~ -1 + X, data)
pvals.ols = p_adjust(ols, method = "RW", B = 1000)
pvals.ols = p_adjust(ols, method = "RW", B = 1000, test.index = c(1,2,5))
```
pvals.ols = p_adjust(ols, method = "RW", B = 1000, test.index = c(rep(TRUE, 5), rep(FALSE, p-5)))

**rlasso**

*rlasso: Function for Lasso estimation under homoscedastic and heteroscedastic non-Gaussian disturbances*

**Description**

The function estimates the coefficients of a Lasso regression with data-driven penalty under homoscedasticity and heteroscedasticity with non-Gaussian noise and X-dependent or X-independent design. The method of the data-driven penalty can be chosen. The object which is returned is of the S3 class rlasso.

**Usage**

rlasso(x, ...)

```r
rlasso(formula, data = NULL, post = TRUE, intercept = TRUE, model = TRUE, penalty = list(homoscedastic = FALSE, X.dependent.lambda = FALSE, lambda.start = NULL, c = 1.1, gamma = 0.1/log(n)), control = list(numIter = 15, tol = 10^-5, threshold = NULL), ...)
```

```
rlasso(character, x, data = NULL, post = TRUE, intercept = TRUE, model = TRUE, penalty = list(homoscedastic = FALSE, X.dependent.lambda = FALSE, lambda.start = NULL, c = 1.1, gamma = 0.1/log(n)), control = list(numIter = 15, tol = 10^-5, threshold = NULL), ...)
```

```
# Default S3 method:
ralss(y, x, ...)
```

post = TRUE,
intercept = TRUE,
model = TRUE,
penalty = list(homoscedastic = FALSE, X.dependent.lambda = FALSE, lambda.start = NULL,
c = 1.1, gamma = 0.1/log(n)),
control = list(numIter = 15, tol = 10^-5, threshold = NULL),
...)

Arguments

x regressors (vector, matrix or object can be coerced to matrix)
... further arguments (only for consistent definition of methods)
formula an object of class "formula" (or one that can be coerced to that class): a symbolic
description of the model to be fitted in the form y~x
data an optional data frame, list or environment (or object coercible by as.data.frame
to a data frame) containing the variables in the model. If not found in data, the
variables are taken from environment(formula), typically the environment from
which rlasso is called.
post logical. If TRUE, post-Lasso estimation is conducted.
intercept logical. If TRUE, intercept is included which is not penalized.
model logical. If TRUE (default), model matrix is returned.
penalty list with options for the calculation of the penalty.
  • c and gamma constants for the penalty with default c=1.1 and gamma=0.1
  • homoscedastic logical, if homoscedastic errors are considered (default FALSE). Option none is described below.
  • X.dependent.lambda logical, TRUE, if the penalization parameter depends
    on the design of the matrix x. FALSE, if independent of the design matrix
    (default).
  • numSim number of simulations for the dependent methods, default=5000
  • lambda.start initial penalization value, compulsory for method "none"
control list with control values. numIter number of iterations for the algorithm for the
  estimation of the variance and data-driven penalty, ie. loadings, tol tolerance
  for improvement of the estimated variances. threshold is applied to the final
  estimated lasso coefficients. Absolute values below the threshold are set to zero.
y dependent variable (vector, matrix or object can be coerced to matrix)

Details

The function estimates the coefficients of a Lasso regression with data-driven penalty under homoscedasticity / heteroscedasticity and non-Gaussian noise. The options homoscedastic is a logical with FALSE by default. Moreover, for the calculation of the penalty parameter it can be chosen, if the penalization parameter depends on the design matrix (X.dependent.lambda=TRUE) or independent (default, X.dependent.lambda=FALSE). The default value of the constant c is 1.1 in the post-Lasso case and 0.5 in the Lasso case. A special option is to set homoscedastic to
none and to supply a values \texttt{lambda.start}. Then this value is used as penalty parameter with independent design and heteroscedastic errors to weight the regressors. For details of the implementation of the Algorithm for estimation of the data-driven penalty, in particular the regressor-independent loadings, we refer to Appendix A in Belloni et al. (2012). When the option "none" is chosen for homoscedastic (together with \texttt{lambda.start}), lambda is set to \texttt{lambda.start} and the regressor-independent loadings und heteroscedasticity are used. The options "X-dependent" and "X-independent" under homoscedasticity are described in Belloni et al. (2013).

The option \texttt{post=TRUE} conducts post-lasso estimation, i.e. a refit of the model with the selected variables.

\textbf{Value}

\texttt{rlasso} returns an object of class \texttt{rlasso}. An object of class "rlasso" is a list containing at least the following components:

- coefficients parameter estimates
- beta parameter estimates (named vector of coefficients without intercept)
- intercept value of the intercept
- index index of selected variables (logical vector)
- lambda data-driven penalty term for each variable, product of lambda0 (the penalization parameter) and the loadings
- lambda0 penalty term
- loadings loading for each regressor
- residuals residuals, response minus fitted values
- sigma root of the variance of the residuals
- iter number of iterations
- call function call
- options options
- model model matrix (if \texttt{model = TRUE} in function call)

\textbf{References}


\textbf{Examples}

```r
set.seed(1)
n = 100  # sample size
p = 100  # number of variables
s = 3    # number of variables with non-zero coefficients
X = Xnames = matrix(rnorm(n*p), ncol=p)
colnames(Xnames) <- paste("V", 1:p, sep="")
```
beta = c(rep(5,s), rep(0,p-s))
Y = X%*%beta + rnorm(n)
reg.lasso <- rlasso(Y~Xnames)
Xnew = matrix(rnorm(n*p), ncol=p) # new X
colnames(Xnew) <- paste("V", 1:p, sep="")
Ynew = Xnew%*%beta + rnorm(n) # new Y
yhat = predict(reg.lasso, newdata = Xnew)

---

rlassoATE

Functions for estimation of treatment effects

Description

This class of functions estimates the average treatment effect (ATE), the ATE of the treated (ATET), the local average treatment effects (LATE) and the LATE of the treated (LATET). The estimation methods rely on immunized / orthogonal moment conditions which guarantee valid post-selection inference in a high-dimensional setting. Further details can be found in Belloni et al. (2014).

Usage

rlassoATE(x, ...)

## Default S3 method:
rlassoATE(x, d, y, bootstrap = "none", nRep = 500, ...)

## S3 method for class 'formula'
rlassoATE(formula, data, bootstrap = "none", nRep = 500, ...)

rlassoATET(x, ...)

## Default S3 method:
rlassoATET(x, d, y, bootstrap = "none", nRep = 500, ...)

## S3 method for class 'formula'
rlassoATET(formula, data, bootstrap = "none", nRep = 500, ...)

rlassoLATE(x, ...)

## Default S3 method:
rlassoLATE(
x,
d,
y,
z,
bootstrap = "none",
nRep = 500,
post = TRUE,
intercept = TRUE,
always_takers = TRUE,
ever_takers = TRUE,
...
)

## S3 method for class 'formula'
rlassoLATE(
  formula,
data,
bootstrap = "none",
nRep = 500,
post = TRUE,
intercept = TRUE,
always_takers = TRUE,
ever_takers = TRUE,
...
)

rlassoLATET(x, ...)

## Default S3 method:
rlassoLATET(
x,
d,
y,
z,
bootstrap = "none",
nRep = 500,
post = TRUE,
intercept = TRUE,
always_takers = TRUE,
...
)

## S3 method for class 'formula'
rlassoLATET(
  formula,
data,
bootstrap = "none",
nRep = 500,
post = TRUE,
intercept = TRUE,
always_takers = TRUE,
...
)
Arguments

- **x**: exogenous variables
- **...**: arguments passed, e.g. intercept and post
- **d**: treatment variable (binary)
- **y**: outcome variable / dependent variable
- **bootstrap**: bootstrap method which should be employed: 'none', 'Bayes', 'normal', 'wild'
- **nRep**: number of replications for the bootstrap
- **formula**: An object of class Formula of the form "$ y \sim x + d | x $" with y the outcome variable, d treatment variable, and x exogenous variables.
- **data**: An optional data frame, list or environment (or object coercible by as.data.frame to a data frame) containing the variables in the model. If not found in data, the variables are taken from environment(formula), typically the environment from which rlassoATE is called.
- **z**: instrumental variables (binary)
- **post**: logical. If TRUE, post-lasso estimation is conducted.
- **intercept**: logical. If TRUE, intercept is included which is not
- **always_takers**: option to adapt to cases with (default) and without always-takers. If FALSE, the estimator is adapted to a setting without always-takers.
- **never_takers**: option to adapt to cases with (default) and without never-takers. If FALSE, the estimator is adapted to a setting without never-takers.

Details

Details can be found in Belloni et al. (2014).

Value

Functions return an object of class rlassoTE with estimated effects, standard errors and individual effects in the form of a list.

References

Description
Estimation and inference of (low-dimensional) target coefficients in a high-dimensional linear model.

Usage
rlassoEffects(x, ...)

## Default S3 method:
rlassoEffects(
x,
y,
index = c(1:ncol(x)),
method = "partialling out",
I3 = NULL,
post = TRUE,
...
)

## S3 method for class 'formula'
rlassoEffects(
formula,
data,
I,
method = "partialling out",
included = NULL,
post = TRUE,
...
)

rlassoEffect(x, y, d, method = "double selection", I3 = NULL, post = TRUE, ...)

Arguments
- x: matrix of regressor variables serving as controls and potential treatments. For rlassoEffect it contains only controls, for rlassoEffects both controls and potential treatments. For rlassoEffects it must have at least two columns.
- ...: parameters passed to the rlasso function.
- y: outcome variable (vector or matrix)
- index: vector of integers, logicals or variables names indicating the position (column) of variables (integer case), logical vector of length of the variables (TRUE or FALSE) or the variable names of x which should be used for inference / as treatment variables.
method

For the 'double selection'-method the logical vector I3 has same length as the number of variables in x; indicates if variables (TRUE) should be included in any case to the model and they are exempt from selection. These variables should not be included in the index; hence the intersection with index must be the empty set. In the case of partialling out it is ignored.

post

logical, if post Lasso is conducted with default TRUE.

formula

An element of class formula specifying the linear model.

data

an optional data frame, list or environment (or object coercible by as.data.frame to a data frame) containing the variables in the model. If not found in data, the variables are taken from environment(formula), typically the environment from which the function is called.

I

An one-sided formula specifying the variables for which inference is conducted.

included

One-sided formula of variables which should be included in any case (only for method="double selection").

d

variable for which inference is conducted (treatment variable)

Details

The functions estimates (low-dimensional) target coefficients in a high-dimensional linear model. An application is e.g. estimation of a treatment effect $\alpha_0$ in a setting of high-dimensional controls. The user can choose between the so-called post-double-selection method and partialling-out. The idea of the double selection method is to select variables by Lasso regression of the outcome variable on the control variables and the treatment variable on the control variables. The final estimation is done by a regression of the outcome on the treatment effect and the union of the selected variables in the first two steps. In partialling-out first the effect of the regressors on the outcome and the treatment variable is taken out by Lasso and then a regression of the residuals is conducted. The resulting estimator for $\alpha_0$ is normal distributed which allows inference on the treatment effect. It presents a wrap function for rlassoEffect which does inference for a single variable.

Value

The function returns an object of class rlassoEffects with the following entries:

coefficients vector with estimated values of the coefficients for each selected variable
se
standard error (vector)
t t-statistic
pval p-value
samplesize sample size of the data set
index index of the variables for which inference is performed

References

Examples

library(hdm); library(ggplot2)
set.seed(1)
n = 100 # sample size
p = 100 # number of variables
s = 3 # number of non-zero variables
X = matrix(rnorm(n*p), ncol=p)
colnames(X) <- paste("X", 1:p, sep="")
beta = c(rep(3,s), rep(0,p-s))
y = 1 + X%*%beta + rnorm(n)
data = data.frame(cbind(y, X))
colnames(data)[1] <- "y"
fm = paste("y ~", paste(colnames(X), collapse="+"))
fm = as.formula(fm)
lasso.effect = rlassoEffects(X, y, index=c(1,2,3,50))
lasso.effect = rlassoEffects(fm, I = ~ X1 + X2 + X3 + X50, data=data)
print(lasso.effect)
summary(lasso.effect)
confint(lasso.effect)
plot(lasso.effect)

---

rlassoIV

Post-Selection and Post-Regularization Inference in Linear Models with Many Controls and Instruments

Description

The function estimates a treatment effect in a setting with very many controls and very many instruments (even larger than the sample size).

Usage

rlassoIV(x, ...)

## Default S3 method:
rlassoIV(x, d, y, z, select.Z = TRUE, select.X = TRUE, post = TRUE, ...)

## S3 method for class 'formula'
rlassoIV(formula, data, select.Z = TRUE, select.X = TRUE, post = TRUE, ...)

rlassoIVmult(x, d, y, z, select.Z = TRUE, select.X = TRUE, ...)

Arguments

x          matrix of exogenous variables
...
arguments passed to the function rlasso
d          endogenous variable
yalssIV

y
outcome / dependent variable (vector or matrix)
z
matrix of instrumental variables
select.Z
logical, indicating selection on the instruments.
select.X
logical, indicating selection on the exogenous variables.
post
logical, whether post-Lasso should be conducted (default=TRUE)
formula
An object of class Formula of the form "y ~ x + d | x + z" with y the outcome variable, d endogenous variable, z instrumental variables, and x exogenous variables.
data
an optional data frame, list or environment (or object coercible by as.data.frame to a data frame) containing the variables in the model. If not found in data, the variables are taken from environment(formula), typically the environment from which rlassoIV is called.

Details
The implementation for selection on x and z follows the procedure described in Chernozhukov et al. (2015) and is built on 'triple selection' to achieve an orthogonal moment function. The function returns an object of S3 class rlassoIV. Moreover, it is wrap function for the case that selection should be done only with the instruments Z (rlassoIVselectZ) or with the control variables X (rlassoIVselectX) or without selection (tsls). Exogenous variables x are automatically used as instruments and added to the instrument set z.

Value
an object of class rlassoIV containing at least the following components:

coefficients
estimated parameter value
se
variance-covariance matrix

References

Examples
### Not run:
data(EminentDomain)
z <- EminentDomain$logGDP$z # instruments
x <- EminentDomain$logGDP$x # exogenous variables
y <- EminentDomain$logGDP$y # outcome variable
d <- EminentDomain$logGDP$d # treatment / endogenous variable
lasso.IV.Z = rlassoIV(x=x, d=d, y=y, z=z, select.X=FALSE, select.Z=TRUE)
summary(lasso.IV.Z)
confint(lasso.IV.Z)
### End(Not run)
rlassoIVselectX

Instrumental Variable Estimation with Selection on the exogenous Variables by Lasso

Description

This function estimates the coefficient of an endogenous variable by employing Instrument Variables in a setting where the exogenous variables are high-dimensional and hence selection on the exogenous variables is required. The function returns an element of class rlassoIVselectX.

Usage

\[
\text{rlassoIVselectX}(x, \ldots)
\]

## Default S3 method:
\[
rlassoIVselectX(x, d, y, z, \text{post} = \text{TRUE}, \ldots)
\]

## S3 method for class 'formula'
\[
rlassoIVselectX(\text{formula}, \text{data}, \text{post} = \text{TRUE}, \ldots)
\]

Arguments

- **x**: exogenous variables in the structural equation (matrix)
- **...**: arguments passed to the function rlasso
- **d**: endogenous variables in the structural equation (vector or matrix)
- **y**: outcome or dependent variable in the structural equation (vector or matrix)
- **z**: set of potential instruments for the endogenous variables.
- **post**: logical. If TRUE, post-lasso estimation is conducted.
- **formula**: An object of class Formula of the form " y ~ x + d | x + z" with y the outcome variable, d endogenous variable, z instrumental variables, and x exogenous variables.
- **data**: An optional data frame, list or environment (or object coercible by as.data.frame to a data frame) containing the variables in the model. If not found in data, the variables are taken from environment(formula), typically the environment from which rlassoIVselectX is called.

Details

The implementation is a special case of of Chernozhukov et al. (2015). The option post=TRUE conducts post-lasso estimation for the Lasso estimations, i.e. a refit of the model with the selected variables. Exogenous variables x are automatically used as instruments and added to the instrument set z.
Value

An object of class rlassoIVselectX containing at least the following components:

- coefficients: estimated parameter vector
- vcov: variance-covariance matrix
- residuals: residuals
- samplesize: sample size

References


Examples

```r
library(hdm)
data(AJR); y = AJR$GDP; d = AJR$Exprop; z = AJR$logMort
x = model.matrix(~ -1 + (Latitude + Latitude2 + Africa + Asia + Namer + Samer)^2, data=AJR)
dim(x)
#AJR.Xselect = rlassoIV(x=x, d=d, y=y, z=z, select.X=TRUE, select.Z=FALSE)
AJR.Xselect = rlassoIV(GDP ~ Exprop + (Latitude + Latitude2 + Africa + Asia + Namer + Samer)^2 | logMort + (Latitude + Latitude2 + Africa + Asia + Namer + Samer)^2, data=AJR, select.X=TRUE, select.Z=FALSE)
summary(AJR.Xselect)
confint(AJR.Xselect)
```

---

rlassoIVselectZ  
Instrumental Variable Estimation with Lasso

Description

This function selects the instrumental variables in the first stage by Lasso. First stage predictions are then used in the second stage as optimal instruments to estimate the parameter vector. The function returns an element of class rlassoIVselectZ

Usage

```r
rlassoIVselectZ(x, ...)
```

## Default S3 method:
rlassoIVselectZ(x, d, y, z, post = TRUE, intercept = TRUE, ...)

## S3 method for class 'formula'
rlassoIVselectZ(formula, data, post = TRUE, intercept = TRUE, ...)

Arguments

- **x**: exogenous variables in the structural equation (matrix)
- **...**: arguments passed to the function `rlasso`.
- **d**: endogenous variables in the structural equation (vector or matrix)
- **y**: outcome or dependent variable in the structural equation (vector or matrix)
- **z**: set of potential instruments for the endogenous variables. Exogenous variables serve as their own instruments.
- **post**: logical. If TRUE, post-lasso estimation is conducted.
- **intercept**: logical. If TRUE, intercept is included in the second stage equation.
- **formula**: An object of class `Formula` of the form "y ~ x + d | x + z" with y the outcome variable, d endogenous variable, z instrumental variables, and x exogenous variables.
- **data**: An optional data frame, list or environment (or object coercible by `as.data.frame` to a data frame) containing the variables in the model. If not found in data, the variables are taken from environment(formula), typically the environment from which `rlassoIVselectZ` is called.

Details

The implementation follows the procedure described in Belloni et al. (2012). Option `post=TRUE` conducts post-lasso estimation, i.e. a refit of the model with the selected variables, to estimate the optimal instruments. The parameter vector of the structural equation is then fitted by two-stage least square (tssl) estimation.

Value

An object of class `rlassoIVselectZ` containing at least the following components:

- **coefficients**: estimated parameter vector
- **vcov**: variance-covariance matrix
- **residuals**: residuals
- **samplesize**: sample size
- **selection.matrix**: matrix of selected variables in the first stage for each endogenous variable

References

rlassologit

rlassologit: Function for logistic Lasso estimation

Description
The function estimates the coefficients of a logistic Lasso regression with data-driven penalty. The method of the data-driven penalty can be chosen. The object which is returned is of the S3 class rlassologit

Usage
rlassologit(x, ...)

## S3 method for class 'formula'
rlassologit(
  formula,
  data = NULL,
  post = TRUE,
  intercept = TRUE,
  model = TRUE,
  penalty = list(lambda = NULL, c = 1.1, gamma = 0.1/log(n)),
  control = list(threshold = NULL),
  ...
)

## S3 method for class 'character'
rlassologit(
  x,
  data = NULL,
  post = TRUE,
  intercept = TRUE,
  model = TRUE,
  penalty = list(lambda = NULL, c = 1.1, gamma = 0.1/log(n)),
  control = list(threshold = NULL),
  ...
)

## Default S3 method:
rlassologit(
  x,
  y,
  post = TRUE,
  intercept = TRUE,
  model = TRUE,
  penalty = list(lambda = NULL, c = 1.1, gamma = 0.1/log(n)),
  control = list(threshold = NULL),
  ...
)
Arguments

- **x**: regressors (matrix)
- **formula**: an object of class 'formula' (or one that can be coerced to that class): a symbolic description of the model to be fitted in the form \( y \sim x \).
- **data**: an optional data frame, list or environment.
- **post**: logical. If TRUE, post-lasso estimation is conducted.
- **intercept**: logical. If TRUE, intercept is included which is not penalized.
- **model**: logical. If TRUE (default), model matrix is returned.
- **penalty**: list with options for the calculation of the penalty. \( c \) and \( \gamma \) constants for the penalty.
- **control**: list with control values. \( \text{threshold} \) is applied to the final estimated lasso coefficients. Absolute values below the threshold are set to zero.
- **y**: dependent variable (vector or matrix)

Details

The function estimates the coefficients of a Logistic Lasso regression with data-driven penalty. The option \( \text{post}=\text{TRUE} \) conducts post-lasso estimation, i.e. a refit of the model with the selected variables.

Value

`rlassologit` returns an object of class `rlassologit`. An object of class `rlassologit` is a list containing at least the following components:

- **coefficients**: parameter estimates
- **beta**: parameter estimates (without intercept)
- **intercept**: value of intercept
- **index**: index of selected variables (logicals)
- **lambda**: penalty term
- **residuals**: residuals
- **sigma**: root of the variance of the residuals
- **call**: function call
- **options**: options

References

Examples

```r
## Not run:
library(hdm)
## DGP
set.seed(2)
n <- 250
p <- 100
px <- 10
X <- matrix(rnorm(n*p), ncol=p)
beta <- c(rep(2,px), rep(0,p-px))
intercept <- 1
P <- exp(intercept + X %*% beta)/(1+exp(intercept + X %*% beta))
y <- rbinom(length(y), size=1, prob=P)
## fit rlassologit object
rlassologit.reg <- rlassologit(y~X)
## methods
summary(rlassologit.reg, all=F)
print(rlassologit.reg)
predict(rlassologit.reg, type='response')
X3 <- matrix(rnorm(n*p), ncol=p)
predict(rlassologit.reg, newdata=X3)
## End(Not run)
```

rlassologitEffects  rigorous Lasso for Logistic Models: Inference

Description

The function estimates (low-dimensional) target coefficients in a high-dimensional logistic model.

Usage

```r
rlassologitEffects(x, ...)  

## Default S3 method:  
rlassologitEffects(x, y, index = c(1:ncol(x)), I3 = NULL, post = TRUE, ...)

## S3 method for class 'formula'  
rlassologitEffects(formula, data, I, included = NULL, post = TRUE, ...)

rlassologitEffect(x, y, d, I3 = NULL, post = TRUE)
```

Arguments

- `x`  
  matrix of regressor variables serving as controls and potential treatments. For `rlassologitEffect` it contains only controls, for `rlassologitEffects` both controls and potential treatments. For `rlassologitEffects` it must have at least two columns.
... additional parameters
y outcome variable
index vector of integers, logical or names indicating the position (column) or name of variables of x which should be used as treatment variables.
I3 logical vector with same length as the number of controls; indicates if variables (TRUE) should be included in any case.
post logical. If TRUE, post-Lasso estimation is conducted.
formula An element of class formula specifying the linear model.
data an optional data frame, list or environment (or object coercible by as.data.frame to a data frame) containing the variables in the model. If not found in data, the variables are taken from environment(formula), typically the environment from which the function is called.
I An one-sided formula specifying the variables for which inference is conducted.
included One-sided formula of variables which should be included in any case.
d variable for which inference is conducted (treatment variable)

Details

The functions estimates (low-dimensional) target coefficients in a high-dimensional logistic model. An application is e.g. estimation of a treatment effect $\alpha_0$ in a setting of high-dimensional controls. The function is a wrap function for rlassologitEffect which does inference for only one variable (d).

Value

The function returns an object of class rlassologitEffects with the following entries:

coefficients estimated value of the coefficients
se standard errors
t t-statistics
pval p-values
samplesize sample size of the data set
I index of variables of the union of the lasso regressions

References

A. Belloni, V. Chernozhukov, Y. Wei (2013). Honest confidence regions for a regression parameter in logistic regression with a large number of controls. cemmap working paper CWP67/13.

Examples

```r
## Not run:
library(hdm)
## DGP
set.seed(2)
n <- 250
```
p <- 100
px <- 10
X <- matrix(rnorm(n*p), ncol=p)
colnames(X) = paste("V", 1:p, sep="")
beta <- c(rep(2,px), rep(0,p-px))
intercept <- 1
P <- exp(intercept + X %*% beta)/(1+exp(intercept + X %*% beta))
y <- rbinom(n, size=1, prob=P)
xd <- X[,2:50]
d <- X[,1]
logit.effect <- rlassologitEffect(x=xd, d=d, y=y)
logit.effects <- rlassologitEffects(X,y, index=c(1,2,40))
logit.effects.f <- rlassologitEffects(y ~ X, I ~ V1 + V2)
## End(Not run)

**summary.rlassoEffects**  
*Summarizing rlassoEffects fits*

### Description

Summary method for class `rlassoEffects`

### Usage

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

```
## S3 method for class 'summary.rlassoEffects'
print(x, digits = max(3L, getOption("digits") - 3L), ...)
```

### Arguments

- **object**: an object of class `rlassoEffects`, usually a result of a call to `rlassoEffects`
- **...**: further arguments passed to or from other methods.
- **x**: an object of class `summary.rlassoEffects`, usually a result of a call or `summary.rlassoEffects`
- **digits**: the number of significant digits to use when printing.

### Details

Summary of objects of class `rlassoEffects`
Two-Stage Least Squares Estimation (TSLS)

**Description**

The function does Two-Stage Least Squares Estimation (TSLS).

**Usage**

```r
tsls(x, ...)
```

```r
## Default S3 method:
## tsls(x, d, y, z, intercept = TRUE, homoscedastic = TRUE, ...)
```

```r
## S3 method for class 'formula'
## tsls(formula, data, intercept = TRUE, homoscedastic = TRUE, ...)
```

**Arguments**

- `x`: exogenous variables
- `...`: further arguments (only for consistent definition of methods)
- `d`: endogenous variables
- `y`: outcome variable
- `z`: instruments
- `intercept`: logical, if intercept should be included
- `homoscedastic`: logical, if homoscedastic (TRUE, default) or heteroscedastic errors (FALSE) should be calculated.
- `formula`: An object of class `Formula` of the form " `y ~ x + d | x + z` " with `y` the outcome variable, `d` endogenous variable, `z` instrumental variables, and `x` exogenous variables.
- `data`: An optional data frame, list or environment (or object coercible by `as.data.frame` to a data frame) containing the variables in the model. If not found in data, the variables are taken from environment(formula), typically the environment from which `tsls` is called.

**Details**

The function computes tsls estimate (coefficients) and variance-covariance-matrix assuming homoskedasticity for outcome variable `y` where `d` are endogenous variables in structural equation, `x` are exogenous variables in structural equation and `z` are instruments. It returns an object of class `tols` for which the methods print and summary are provided.
Value

The function returns a list with the following elements:

- **coefficients**: coefficients
- **vcov**: variance-covariance matrix
- **residuals**: outcome minus predicted values
- **call**: function call
- **samplesize**: sample size
- **se**: standard error
Index

* 401(k)
  pension, 15
* GDP
  Growth Data, 12
* Growth
  Growth Data, 12
* datasets
  AJR, 3
  BLP, 4
  cps2012, 9
  EminentDomain, 10
  Growth Data, 12
  pension, 15
* lasso
  rlassologit, 39
* logistic
  rlassologit, 39
* pension
  pension, 15
* regression
  rlassologit, 39

AJR, 3
ATE (rlassoATE), 29
ate (rlassoATE), 29
ATET (rlassoATE), 29
atet (rlassoATE), 29

BLP, 4

coef.rlassoEffects, 5
coef.rlassoIV, 6
coef.rlassoIVselectX, 7
coef.rlassoIVselectZ, 8
confint.rlassoEffects
  (print.rlassoEffects), 18
confint.rlassoIV (print.rlassoIV), 19
confint.rlassoIVselectX
  (print.rlassoIVselectX), 19
confint.rlassoIVselectZ
  (print.rlassoIVselectZ), 20
confint.rlassologitEffects
  (print.rlassologitEffects), 21
confint.rlassoTE (print.rlassoTE), 22
cps2012, 9
data (pension), 15
EminentDomain, 10
Example (Growth Data), 12
GDP (Growth Data), 12
Growth (Growth Data), 12
Growth Data, 12
GrowthData (Growth Data), 12
hdm (hdm-package), 2
hdm-package, 2
lambdaCalculation, 13
LassoShooting.fit, 14
LATE (rlassoATE), 29
late (rlassoATE), 29
LATET (rlassoATE), 29
latet (rlassoATE), 29
lm, 25

methods.rlasso (print.rlasso), 17
methods.rlassoEffects
  (print.rlassoEffects), 18
methods.rlassoIV (print.rlassoIV), 19
methods.rlassoIVselectX
  (print.rlassoIVselectX), 19
methods.rlassoIVselectZ
  (print.rlassoIVselectZ), 20
methods.rlassologit
  (predict.rlassologit), 16
methods.rlassologitEffects
  (print.rlassologitEffects), 21
methods.rlassoTE (print.rlassoTE), 22
methods tsls (print.tsls), 22
model.matrix rlasso (print.rlasso), 17
model.matrix rlassologit (predict.rlassologit), 16
p.adjust.methods, 25
p_adjust, 24
pension, 15
plans (pension), 15
plot.rlassoEffects (print.rlassoEffects), 18
predict.rlasso (print.rlasso), 17
predict.rlassologit, 16
print.rlasso, 17
print.rlassoEffects, 18
print.rlassoIV, 19
print.rlassoIVselectX, 19
print.rlassoIVselectZ, 20
print.rlassologit (predict.rlassologit), 16
print.rlassologitEffects, 21
print.rlassoTE, 22
print.summary.rlassoEffects (summary.rlassoEffects), 43
print.tsls, 22
print_coef, 23
rlasso, 13, 26
rlassoATE, 29
rlassoATE (rlassoATE), 29
rlassoEffect (rlassoEffects), 32
rlassoEffects, 25, 32
rlassoIV, 34
rlassoIVmult (rlassoIV), 34
rlassoIVselectX, 36
rlassoIVselectZ, 37
rlassoLATE (rlassoATE), 29
rlassoLATE (rlassoATE), 29
rlassologit, 39
rlassologitEffect (rlassologitEffects), 41
rlassologitEffects, 41
summary.rlasso (print.rlasso), 17
summary.rlassoEffects, 18, 43
summary.rlassoIV (print.rlassoIV), 19
summary.rlassoIVselectX (print.rlassoIVselectX), 19
summary.rlassoIVselectZ (print.rlassoIVselectZ), 20
summary.rlassologit (predict.rlassologit), 16
summary.rlassologitEffects (print.rlassologitEffects), 21
summary.rlassoTE (print.rlassoTE), 22
summary.tsls (print.tsls), 22
tsls, 44
wealth (pension), 15