Package ‘ivtools’

December 11, 2018

Type Package

Title Instrumental Variables

Version 2.1.0

Date 2018-12-01

Author Arvid Sjolander, Elisabeth Dahlqwist, Torben Martinussen

Maintainer Arvid Sjolander <arvid.sjolander@ki.se>

Description Contains tools for instrumental variables estimation. Currently, non-parametric bounds, two-stage estimation and G-estimation are implemented. Balke, A. and Pearl, J. (1997) <doi:10.2307/2965583>, Vansteelandt S., Bowden J., Babanezhad M., Goetghebeur E. (2011) <doi:10.1214/11-STS360>.

License LGPL (>= 3)

Imports stats, data.table, numDeriv, nleqslv, survival, ahaz, Rcpp

LinkingTo Rcpp

NeedsCompilation yes

RoxygenNote 6.0.1

Repository CRAN

Date/Publication 2018-12-11 15:20:03 UTC

R topics documented:

---

ah *Fitting semiparametric additive hazards regression models.*

Description

ah is a wrapper around the ahaz function in the ahaz package, with a more user-friendly and standard interface. Refer to the manual for ahaz for details.

Usage

```
ah(formula, data, weights, robust=FALSE)
```

Arguments

formula	an object of class "formula": a symbolic description of the model to be fitted.
data	a data frame containing the variables in the model.
weights	an optional vector of prior weights to be used in the fitting process.
robust	robust calculation of variance; see manual for ahaz.

Details

See manual for ahaz.

Value

An object of class "ah" is a list containing the same elements as an object of class "ahaz", plus

call	the matched call.
formula	the formula argument.
coefficients	a named vector of estimated coefficients.
vcov	the variance-covariance matrix for the estimated coefficients.
incl	the ahaz function does not allow for missing data. Thus, if data contains rows with missing data on any of the variables in the model, then these rows are excluded before calling ahaz. incl is a vector containing the rownames of those rows that are included in the analysis, that is, the rows with no missing data on any of the variables in the model.

Note

The ahaz function does not allow for ties. Thus, before calling ah any ties have to be manually broken.

Author(s)

Arvid Sjolander.

References

Lin D.Y., Ying Z. (1994). Semiparametric analysis of the additive risk model. *Biometrika* **81**(1), 61-71.

Examples

```
require(ahaz)

##This example is adapted from the example given for the ahaz function
##in the ahaz package

data(sorlie)

# Break ties
set.seed(10101)
sorlie$time <- sorlie$time+runif(nrow(sorlie))*1e-2

# Fit additive hazards model
fit <- ah(formula=Surv(time, status)~X13+X14+X15+X16+X17+X18+X19+X20+X21+X22,
  data=sorlie)
summary(fit)
```

`confint.ivmod` *Confidence interval*

Description

This is a `confint` method for class `"ivmod"`.

Usage

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

Arguments

`object`	an object of class `"ivmod"`.
`parm`	not used.
`level`	the coverage probability of the confidence intervals.
`...`	not used.

Author(s)

Arvid Sjolander.

---

estfun — *Computes the estimating function sum for* "ivmod" *objects, fitted with* estmethod="g".

Description

estfun computes the estimating function $H(\psi)$ for a "ivmod" object, fitted with estmethod="g", for a range of values of ψ. The estfun is not implemented for "ivah" objects, since G-estimation in additive hazards models is based on a recursive estimation technique, and not standard estimating equations.

Usage

```
estfun(object, lower, upper, step)
```

Arguments

object	an object of class "ivmod", fitted with estmethod="g".
lower	an optional vector of lower values for ψ. Defaults to $\psi - 0.5$.
upper	an optional vector of upper values for ψ. Defaults to $\psi + 0.5$.
step	an optional vector of steps between lower and upper. Defaults to 0.01 for each element of ψ.

Details

estfun may be useful for visual inspection of the estimating function, to make sure that a solution to the estimating equation

$$H(\psi) = 0$$

was found, see 'Examples'. For the i:th element of ψ, the estimating function sum is computed for a range of values within (lower[i], upper[i]), at the G-estimate of the remaining elements of ψ.

Value

An object of class "estfun" is a list containing

f	a named list of matricies; one matrix for each element of ψ. The first column of the i:th matrix contains the values for the i:th element of ψ at which the estimating function sum is computed, the second column contains the values of the estimating function sum.
est	the G-estimate of ψ.

Author(s)

Arvid Sjolander.

References

Burgess S, Granell R, Palmer TM, Sterne JA, Didelez V. (2014). Lack of identification in semi-parametric instrumental variable models with binary outcomes. *American Journal of Epidemiology* **180**(1), 111-119.

Vansteelandt S., Bowden J., Babanezhad M., Goetghebeur E. (2011). On instrumental variables estimation of causal odds ratios. *Statistical Science* **26**(3), 403-422.

Examples

```
set.seed(9)

##Note: the parameter values in the examples below are chosen to make
##Y0 independent of Z, which is necessary for Z to be a valid instrument.

n <- 1000
psi0 <- 0.5
psi1 <- 0.2

##---Example 1: linear model and interaction between X and L---

L <- rnorm(n)
Z <- rnorm(n, mean=L)
X <- rnorm(n, mean=Z)
m0 <- X-Z+L
Y <- rnorm(n, mean=psi0*X+psi1*X*L+m0)
data <- data.frame(L, Z, X, Y)

#G-estimation
fitZ <- glm(formula=Z~L, data=data)
fitIV <- ivglm(estmethod="g", Z="Z", X="X", Y="Y", fitZ=fitZ, data=data,
  formula=~L, link="identity")
summary(fitIV)
H <- estfun(fitIV)
plot(H)

##---Example 2: logistic model and no covariates---

Z <- rbinom(n, 1, 0.5)
X <- rbinom(n, 1, 0.7*Z+0.2*(1-Z))
m0 <- plogis(1+0.8*X-0.39*Z)
Y <- rbinom(n, 1, plogis(psi0*X+log(m0/(1-m0))))
data <- data.frame(Z, X, Y)

#G-estimation
fitY <- glm(formula=Y~X+Z+X*Z, family="binomial", data=data)
fitIV <- ivglm(estmethod="g", Z="Z", X="X", Y="Y", fitY=fitY, data=data, link="logit")
```

```
summary(fitIV)
H <- estfun(fitIV)
plot(H)
```

---

ivah *Instrumental variable estimation of the causal exposure effect in additive hazards (AH) models*

Description

ivah performs instrumental variable estimation of the causal exposure effect in AH models with individual-level data. Below, Z, X, and T are the instrument, the exposure, and the outcome, respectively. L is a vector of covariates that we wish to control for in the analysis; these would typically be confounders for the instrument and the outcome.

Usage

```
ivah(estmethod, Z, X, T, fitZ=NULL, fitX=NULL, fitT=NULL, data,
  ctrl=FALSE, clusterid=NULL, event, max.time, max.time.psi, n.sim=100, ...)
```

Arguments

estmethod	a string specifying the desired estimation method; either "ts" for two-stage estimation, or "g" for G-estimation.
Z	a string specifying the name of the instrument Z in data. This argument is not used when estmethod="ts".
X	a string specifying the name of the exposure X in data. This argument is not used when estmethod="ts".
T	a string specifying the name of the follow-up time T in data. This argument is not used when estmethod="ts".
fitZ	an object of class "glm", as returned by the glm function in the stats package. This is a fitted GLM for $E(Z\|L)$. This argument is not used when estmethod="ts". It is not required when estmethod="g" and covariates are absent.
fitX	an object of class "glm", as returned by the glm function in the stats package. This is a fitted GLM for $E(X\|L,Z)$. This argument is not used when estmethod="g".
fitT	If estmethod="ts", then this is an object of class "ah", as returned by the ah function in the ivtools package. In this case it is a fitted AH model for $\lambda(t\|L,X)$. This argument is not used when estmethod="g".
data	a data frame containing the variables in the model. The covariates, instrument, exposure and outcome can have arbitrary names, e.g. they don't need to be called L, Z, X and T.
ctrl	logical. Should the control function $R=X-\hat{X}$ be used when re-fitting fitY? This argument is not used when estmethod="g".

`clusterid` : an optional string containing the name of a cluster identification variable when data are clustered. Specifying `clusterid` corrects the standard errors but does not affect the estimates. This argument is not used when `estmethod="g"`, since correction for clustered data is currently not implemented for G-estimation.

`event` : a string specifying the name of the status indicator, 0="no event", 1="event". This argument is not used when `estmethod="ts"`.

`max.time` : optional follow-up for estimating $B(t)$ with G-estimation. Defaults to maximal observed follow-up time in `data`. This argument is not used when `estmethod="ts"`.

`max.time.psi` : optional follow-up for estimating ψ with G-estimation. Defaults to maximal observed follow-up time in `data`. This argument is not used when `estmethod="ts"`.

`n.sim` : optional number of resamplings for testing goodness-of-fit of constant effects model for G-estimation. Defaults to 100. This argument is not used when `estmethod="ts"`.

`...` : optional arguments passed on to the `nleqslv` function, which is used to solve the estimating equations when `estmethod="g"`. See the help pages for `nleqslv`. This argument is not used when `estmethod="ts"`.

Details

The `ivah` estimates different parameters, depending on whether `estmethod="ts"` or `estmethod="g"`. If `estmethod="ts"`, then `ivah` uses two-stage estimation to estimate the parameter ψ in the causal AH model

$$\lambda(t|L,Z,X)-\lambda_0(t|L,Z,X)=m^T(L)X\psi.$$

Here, $\lambda_0(t|L,Z,X)$ is counterfactual hazard function, had the exposure been set to 0. The vector function $m(L)$ contains interaction terms between L and X. These are specified implicitly through the model `fitY`. The model `fitX` is used to construct predictions $\hat{X}=\hat{E}(X|L,Z)$. These predictions are subsequently used to re-fit the model `fitY`, with X replaced with $\hat{X}$. The obtained coefficient(s) for X is the two-stage estimator $\hat{\psi}$.

If `estmethod="g"`, then `ivah` uses G-estimation to estimate the function $B(t)$ in the causal AH model

$$\lambda(t|L,Z,X)-\lambda_0(t|L,Z,X)=XdB(t).$$

It also delivers an estimate of $dB(t)$ assuming that this function is constant across time (=ψ).

Value

`ivah` returns an object of class `"ivah"`, which inherits from class `"ivmod"`. An object of class `"ivah"` is a list containing

`call` : the matched call.

`input` : `input` is a list containing all input arguments

`est` : a vector containing the estimate of ψ.

`vcov` : the variance-covariance matrix for the estimate of ψ, obtained with the sandwich formula.

`converged` : logical. Was a solution found to the estimating equations?

fitZ	the model fitZ used in the estimation process when estmethod="g". This element is NULL when estmethod="ts".
fitY	the re-fitted model fitY used in the estimation process when estmethod="ts". This element is NULL when estmethod="g".
stime	the ordered event times within (0,max.time). This element is NULL when estmethod="ts".
B	the estimate of $B(t)$. This element is NULL when estmethod="ts".
se_B	the standard error of the estimate of $B(t)$. This element is NULL when estmethod="ts".
pval_0	p-value corresponding to supremum test of the null $B(t) = 0$. This element is NULL when estmethod="ts".
eps_B	the iid-decomposition of $\sqrt{n}(\hat{B}(t)-B(t))$. This element is NULL when estmethod="ts".
pval_psi	p-value corresponding to the null $\psi = 0$. This element is NULL when estmethod="ts".
pval_GOF_sup	p-value corresponding to supremum test of the null $B(t) = \psi$. This element is NULL when estmethod="ts".
pval_GOF_CvM	as pval_GOF_sup but now based on the Cramer Von Mises test statistic. This element is NULL when estmethod="ts".
GOF.resamp	a matrix with first row the ordered jump times in (0,max.time.bet), second row the observed test process, and the remaining rows are 50 processes sampled under the null. This element is NULL when estmethod="ts".

Note

ivah allows for weights. However, these are defined implicitly through the input models. Thus, when models are used as input to ivah, these models have to be fitted with the same weights.

Left-truncation and correction of standard errors for clustered data are currerntly not implemented when estmethod="g".

Author(s)

Arvid Sjolander and Torben Martinussen.

References

Martinussen T., Vansteelandt S., Tchetgen Tchetgen E.J., Zucker D.M. (2017). Instrumental variables estimation of exposure effects on a time-to-event endpoint using structural cumulative survival models. *Epidemiology* **73**(4): 1140-1149.

Tchetgen Tchetgen E.J., Walter S., Vansteelandt S., Martinussen T., Glymour M. (2015). Instrumental variable estimation in a survival context. *Epidemiology* **26**(3): 402-410.

Examples

```
require(ahaz)

set.seed(9)
```

```
n <- 1000
psi0 <- 0.2
psi1 <- 0.0

U <- runif(n)
L <- runif(n)
Z <- rbinom(n, 1, plogis(-0.5+L))
X <- runif(n, min=Z+U, max=2+Z+U)
T <- rexp(n, rate=psi0*X+psi1*X*L+0.2*U+0.2*L)
C <- 5  #administrative censoring at t=5
d <- as.numeric(T<C)
T <- pmin(T, C)
data <- data.frame(L, Z, X, T, d)
#break ties
data$T <- data$T+rnorm(n=nrow(data), sd=0.001)

#two-stage estimation
fitX <- glm(formula=X~Z+L, data=data)
fitT <- ah(formula=Surv(T, d)~X+L+X*L, data=data)
fitIV <- ivah(estmethod="ts", fitX=fitX, fitT=fitT, data=data, ctrl=TRUE)
summary(fitIV)

#G-estimation
fitZ <- glm(formula=Z~L, family="binomial", data=data)
fitIV <- ivah(estmethod="g", Z="Z", X="X", T="T", fitZ=fitZ, data=data,
          event="d", max.time=4, max.time.psi=4, n.sim=100)
summary(fitIV)
plot(fitIV)
```

---

ivbounds — *Bounds for counterfactual outcome probabilities in instrumental variables scenarios*

Description

ivbounds computes non-parametric bounds for counterfactual outcome probabilities in instrumental variables scenarios. Let Y, X, and Z be the outcome, exposure, and instrument, respectively. Y and X must be binary, whereas Z can be either binary or ternary. Ternary instruments are common in, for instance, Mendelian randomization. Let $p(Y_x = 1)$ be the counterfactual probability of the outcome, had all subjects been exposed to level x. ivbounds computes bounds for the counterfactuals probabilities $p(Y_1 = 1)$ and $p(Y_0 = 1)$. Below, we define $p_{yx.z} = p(Y = y, X = x|Z = x)$.

Usage

```
ivbounds(data, Z, X, Y, monotonicity=FALSE, weights)
```

Arguments

- `data` — either a data frame containing the variables in the model, or a named vector `(p00.0,...,p11.1)` when Z is binary, or a named vector `(p00.0,...,p11.2)` when Z is ternary.
- `Z` — a string containing the name of the instrument Z in `data` if `data` is a data frame. In this case Z has to be coded as (0,1) when binary, and coded as (0,1,2) when ternary. `Z` is not specified if `data` is a vector of probabilities.
- `X` — a string containing the name of the exposure X in `data` if `data` is a data frame. In this case X has to be coded as (0,1). `X` is not specified if `data` is a vector of probabilities.
- `Y` — a string containing the name of the outcome Y in `data` if `data` is a data frame. In this case Y has to be coded as (0,1). `Y` is not specified if `data` is a vector of probabilities.
- `monotonicity` — logical. It is sometimes realistic to make the monotonicity assumption $z \geq z' \Rightarrow X_z \geq X_{z'}$. Should the bounds be computed under this assumption?
- `weights` — an optional vector of 'prior weights' to be used in the fitting process. Should be NULL or a numeric vector. Only applicable if `data` is a data frame.

Details

`ivbounds` uses linear programming techniques to bound the counterfactual probabilities $p(Y_1 = 1)$ and $p(Y_0 = 1)$. Bounds for a causal effect, defined as a contrast between these, are obtained by plugging in the bounds for $p(Y_1 = 1)$ and $p(Y_0 = 1)$ into the contrast. For instance, bounds for the causal risk difference $p(Y_1 = 1) - p(Y_0 = 1)$ are obtained as $[min\{p(Y_1 = 1)\} - max\{p(Y_0 = 1)\}, max\{p(Y_1 = 1)\} - min\{p(Y_0 = 1)\}]$. In addition to the bounds, `ivbounds` evaluates the IV inequality

$$\max_x \sum_y \max_z p_{yx.z} \leq 1.$$

Value

An object of class `"ivbounds"` is a list containing

- `call` — the matched call.
- `p0` — a named vector with elements `"min"` and `"max"`, containing the lower and upper bounds for $p(Y_0 = 1)$, respectively.
- `p1` — a named vector with elements `"min"` and `"max"`, containing the lower and upper bounds for $p(Y_1 = 1)$, respectively.
- `IVinequality` — logical. Does the IV inequality hold?
- `conditions` — a character vector containing the violated condiations, if `IVinequality=FALSE`.

Author(s)

Arvid Sjolander.

References

Balke, A. and Pearl, J. (1997). *Bounds on treatment effects from studies with imperfect compliance*. *Journal of the American Statistical Association* **92**(439), 1171-1176.

Examples

```
##Vitamin A example from Balke and Pearl (1997).
n000 <- 74
n001 <- 34
n010 <- 0
n011 <- 12
n100 <- 11514
n101 <- 2385
n110 <- 0
n111 <- 9663
n0 <- n000+n010+n100+n110
n1 <- n001+n011+n101+n111

#with data frame...
data <- data.frame(Y=c(0,0,0,0,1,1,1,1), X=c(0,0,1,1,0,0,1,1),
  Z=c(0,1,0,1,0,1,0,1))
n <- c(n000, n001, n010, n011, n100, n101, n110, n111)
b <- ivbounds(data=data, Z="Z", X="X", Y="Y", weights=n)
summary(b)

#...or with vector of probabilities
p <- n/rep(c(n0, n1), 4)
names(p) <- c("p00.0", "p00.1", "p01.0", "p01.1",
  "p10.0", "p10.1", "p11.0", "p11.1")
b <- ivbounds(data=p)
summary(b)
```

`ivcoxph` *Instrumental variable estimation of the causal exposure effect in Cox proportional hazards (PH) models*

Description

`ivcoxph` performs instrumental variable estimation of the causal exposure effect in Cox PH models with individual-level data. Below, Z, X, and T are the instrument, the exposure, and the outcome, respectively. L is a vector of covariates that we wish to control for in the analysis; these would typically be confounders for the instrument and the outcome.

Usage

```
ivcoxph(estmethod, Z, X, T, fitZ=NULL, fitX=NULL, fitT, data,
  formula=~1, ctrl=FALSE, clusterid=NULL, t=NULL, ...)
```

Arguments

estmethod	a string specifying the desired estimation method; either "ts" for two-stage estimation, or "g" for G-estimation.
Z	a string specifying the name of the instrument Z in data. This argument is not used when estmethod="ts".
X	a string specifying the name of the exposure X in data. This argument is not used when estmethod="ts".
T	a string specifying the name of the follow-up time T in data. This argument is not used when estmethod="ts".
fitZ	an object of class "glm", as returned by the glm function in the stats package. This is a fitted GLM for $E(Z\|L)$. This argument is not used when estmethod="ts". It is not required when estmethod="g" and covariates are absent.
fitX	an object of class "glm", as returned by the glm function in the stats package. This is a fitted GLM for $E(X\|L,Z)$. This argument is not used when estmethod="g".
fitT	If estmethod="ts", then this is an object of class "coxph", as returned by the coxph function in the survival package. In this case it is a fitted Cox PH model for $\lambda(t\|L,X)$. If estmethod="g", then this is either an object of class "coxph" or an object of class "survfit", as returned by the coxph function in the survival package. In this case it is a fitted Cox PH model for $\lambda(t\|L,Z,X)$ or a non-parametric model for $S(t\|L,Z,X)$, respectively.
data	a data frame containing the variables in the model. The covariates, instrument, exposure and outcome can have arbitrary names, e.g. they don't need to be called L, Z, X and T.
formula	an object of class "formula", with no left-hand side. This specifies the causal interaction terms $m(L)$; see 'Details'. Defaults to ~1, i.e. main effect only. This argument is not used when estmethod="ts".
ctrl	logical. Should the control function $R=X-\hat{X}$ be used when re-fitting fitY? This argument is not used when estmethod="g".
clusterid	an optional string containing the name of a cluster identification variable when data are clustered. Specifying clusterid corrects the standard errors but does not affect the estimates.
t	a numeric scalar specifying the time point at which to solve the estimating equation when estmethod="g"; see 'Details'. This argument is not used when estmethod="ts". If not supplied, then the estimating equation is solved at the optimal value of t, defined as the value that minimizes $trace\{var(\hat{\psi})\}$; see Martinussen et al (2017).
...	optional arguments passed on to the nleqslv function, which is used to solve the estimating equations when estmethod="g". See the help pages for nleqslv. This argument is not used when estmethod="ts".

Details

`ivcoxph` estimates the parameter ψ in the causal Cox PH model

$$\log\{\lambda(t|L,Z,X)\} - \log\{\lambda_0(t|L,Z,X)\} = m^T(L)X\psi.$$

Here, $\lambda_0(t|L,Z,X)$ is counterfactual hazard function, had the exposure been set to 0. The vector function $m(L)$ contains interaction terms between L and X. If `estmethod="ts"`, then these are specified implicitly through the model `fitY`. If `estmethod="g"`, then these are specified explicitly through the `formula` argument.

If `estmethod="ts"`, then two-stage estimation of ψ is performed. In this case, the model `fitX` is used to construct predictions $\hat{X} = \hat{E}(X|L,Z)$. These predictions are subsequently used to re-fit the model `fitY`, with X replaced with $\hat{X}$. The obtained coefficient(s) for X is the two-stage estimator $\hat{\psi}$.

If `estmethod="g"`, then G-estimation of ψ is performed. In this case, the estimator is obtained as the solution to the estimating equation

$$H(\psi) = \sum_{i=1}^{n} m(L_i)\{Z_i - \hat{E}(Z|L_i)\}h_i(\psi;t) = 0,$$

where

$$h_i(\psi;t) = \hat{S}(t|L_i,Z_i,X_i)^{\exp\{-m^T(L_i)\psi X_i\}}.$$

The equation is solved at the value of t specified by the argument `t`. $\hat{S}(t|L_i,Z_i,X_i)$ is an estimate of $S(t|L_i,Z_i,X_i)$ obtained from the model `fitY`.

Value

`ivcoxph` returns an object of class `"ivcoxph"`, which inherits from class `"ivmod"`. An object of class `"ivcoxph"` is a list containing

`call`	the matched call.
`input`	`input` is a list containing all input arguments
`est`	a vector containing the estimate of ψ.
`vcov`	the variance-covariance matrix for the estimate of ψ, obtained with the sandwich formula.
`estfunall`	a matrix of all subject-specific contributions to the estimating functions used in the estimation process. One row for each subject, one column for each parameter. If `estmethod="ts"`, then the first columns correspond to the parameters estimated by `fitX`, and the last columns correspond to the parameters estimated by the re-fitted model `fitY`. If `estmethod="g"`, then the first columns correspond to ψ, and the remaining columns correspond to the parameters estimated by `fitZ` and `fitY`.
`d.estfun`	the jacobian matrix of `colSums(estfun)`.
`converged`	logical. Was a solution found to the estimating equations?
`fitZ`	the model `fitZ` used in the estimation process when `estmethod="g"`. This element is NULL when `estmethod="ts"`.

`fitY` the re-fitted model `fitY` used in the estimation process when `estmethod="ts"`. This element is NULL when `estmethod="g"`.

`t` the value of `t` used in the estimation process. This element is NULL when `estmethod="ts"`.

Note

`ivglm` allows for weights. However, these are defined implicitly through the input models. Thus, when models are used as input to `ivglm`, these models have to be fitted with the same weights. When `estmethod="g"` the weights are taken from `fitZ`, so this must be specified.

Author(s)

Arvid Sjolander.

References

Martinussen T., Sorensen D.D., Vansteelandt S. (2017). Instrumental variables estimation under a structural Cox model. *Biostatistics* <doi:10.1093/biostatistics/kxx057>.

Tchetgen Tchetgen E.J., Walter S., Vansteelandt S., Martinussen T., Glymour M. (2015). Instrumental variable estimation in a survival context. *Epidemiology* **26**(3): 402-410.

Examples

```
require(survival)

set.seed(9)

##Note: the parameter values in the examples below are chosen to make
##Y0 independent of Z, which is necessary for Z to be a valid instrument.

n <- 10000
psi0 <- 0.5
Z <- rbinom(n, 1, 0.5)
X <- rbinom(n, 1, 0.7*Z+0.2*(1-Z))
m0 <- exp(0.8*X-0.41*Z) #T0 independent of Z at t=1
T <- rexp(n, rate=exp(psi0*X+log(m0)))
C <- rexp(n, rate=exp(psi0*X+log(m0))) #50% censoring
d <- as.numeric(T<C)
T <- pmin(T, C)
data <- data.frame(Z, X, T, d)

#two-stage estimation
fitX <- glm(formula=X~Z, data=data)
fitT <- coxph(formula=Surv(T, d)~X, data=data)
fitIV <- ivcoxph(estmethod="ts", fitX=fitX, fitT=fitT, data=data, ctrl=TRUE)
summary(fitIV)

#G-estimation with non-parametric model for S(t|L,Z,X)
fitT <- survfit(Surv(T, d)~X+Z, data=data)
```

```
fitIV <- ivcoxph(estmethod="g", Z="Z", X="X", T="T", fitT=fitT, data=data, t=1)
summary(fitIV)

#G-estimation with Cox model for \lambda(t|L,Z,X)
fitT <- coxph(Surv(T, d)~X+X+X*Z, data=data)
fitIV <- ivcoxph(estmethod="g", Z="Z", X="X", T="T", fitT=fitT, data=data, t=1)
summary(fitIV)
```

`ivglm` *Instrumental variable estimation of the causal exposure effect in generalized linear models*

Description

`ivglm` performs instrumental variable estimation of the causal exposure effect in generalized linear models with individual-level data. Below, Z, X, and Y are the instrument, the exposure, and the outcome, respectively. L is a vector of covariates that we wish to control for in the analysis; these would typically be confounders for the instrument and the outcome.

Usage

```
ivglm(estmethod, Z, X, Y, fitZ=NULL, fitX=NULL, fitY=NULL, data,
  formula=~1, ctrl=FALSE, clusterid=NULL, link, ...)
```

Arguments

`estmethod`	a string specifying the desired estimation method; either `"ts"` for two-stage estimation, or `"g"` for G-estimation.
`Z`	a string specifying the name of the instrument Z in `data`. This argument is not used when `estmethod="ts"`.
`X`	a string specifying the name of the exposure X in `data`. This argument is not used when `estmethod="ts"`.
`Y`	a string specifying the name of the outcome Y in `data`. This argument is not used when `estmethod="ts"`.
`fitZ`	an object of class `"glm"`, as returned by the `glm` function in the `stats` package. This is a fitted GLM for $E(Z\|L)$. This argument is not used when `estmethod="ts"`. It is not required when `estmethod="g"` and covariates are absent.
`fitX`	an object of class `"glm"`, as returned by the `glm` function in the `stats` package. This is a fitted GLM for $E(X\|L,Z)$. This argument is not used when `estmethod="g"`.
`fitY`	an object of class `"glm"`, as returned by the `glm` function in the `stats` package. This is a fitted GLM for $E(Y\|L,Z,X)$. This argument is not used when `estmethod="g"` and `link="identity"` or `link="log"`.

`data`	a data frame containing the variables in the model. The covariates, instrument, exposure and outcome can have arbitrary names, e.g. they don't need to be called `L`, `Z`, `X` and `Y`.
`formula`	an object of class `"formula"`, with no left-hand side. This specifies the causal interaction terms $m(L)$; see 'Details'. Defaults to `~1`, i.e. main effect only. This argument is not used when `estmethod="ts"`.
`ctrl`	logical. Should the control function $R = X - \hat{X}$ be used when re-fitting `fitY`? This argument is not used when `estmethod="g"`.
`clusterid`	an optional string containing the name of a cluster identification variable when data are clustered. Specifying `clusterid` corrects the standard errors but does not affect the estimates.
`link`	a string specifying the link function for the causal generalized linear model; see 'Details'. Either `"identity"`, `"log"`, or `"logit"`.
`...`	optional arguments passed on to the `nleqslv` function, which is used to solve the estimating equations when `estmethod="g"`. See the help pages for `nleqslv`. This argument is not used when `estmethod="ts"`.

Details

`ivglm` estimates the parameter ψ in the causal generalized linear model

$$\eta\{E(Y|L,Z,X)\} - \eta\{E(Y_0|L,Z,X)\} = m^T(L)X\psi.$$

Here, $E(Y_0|L,Z,X)$ is counterfactual mean of the outcome, had the exposure been set to 0. The link function η is either the identity, log or logit link, as specified by the `link` argument. The vector function $m(L)$ contains interaction terms between L and X. If `estmethod="ts"`, then these are specified implicitly through the model `fitY`. If `estmethod="g"`, then these are specified explicitly through the `formula` argument.

If `estmethod="ts"`, then two-stage estimation of ψ is performed. In this case, the model `fitX` is used to construct predictions $\hat{X} = \hat{E}(X|L,Z)$. These predictions are subsequently used to re-fit the model `fitY`, with X replaced with $\hat{X}$. The obtained coefficient(s) for X is the two-stage estimator $\hat{\psi}$.

If `estmethod="g"`, then G-estimation of ψ is performed. In this case, the estimator is obtained as the solution to the estimating equation

$$H_(\psi) = \sum_{i=1}^{n} m(L_i)\{Z_i - \hat{E}(Z|L_i)\}h_i(\psi) = 0.$$

In this equation, $\hat{E}(Z|L_i)$ is an estimate of $E(Z|L_i)$ obtained from the model `fitZ`. In the absence of L, $E(Z|L_i)$ reduces to $E(Z)$, which is estimated as $\sum_{i=1}^{n} Z_i/n$. In this case, `fitZ` may be omitted. The function $h_i(\psi)$ is defined as

$$h_i(\psi) = Y_i - m^T(L_i)\psi X_i$$

when `link="identity"`,

$$h_i(\psi) = Y_i \exp\{-m^T(L_i)\psi X_i\}$$

when `link="log"`, and

$$h_i(\psi) = \text{expit}\{\hat{E}(Y|L_i, Z_i, X_i) - m^T(L_i)\psi X_i\}$$

when `link="logit"`. In the latter, $\hat{E}(Y|L_i, Z_i, X_i)$ is an estimate of $E(Y|L_i, Z_i, X_i)$ obtained from the model `fitY`.

Value

`ivglm` returns an object of class `"ivglm"`, which inherits from class `"ivmod"`. An object of class `"ivglm"` is a list containing

- `call` the matched call.
- `input` `input` is a list containing all input arguments
- `est` a vector containing the estimate of ψ.
- `vcov` the variance-covariance matrix for the estimate of ψ, obtained with the sandwich formula.
- `estfunall` a matrix of all subject-specific contributions to the estimating functions used in the estimation process. One row for each subject, one column for each parameter. If `estmethod="ts"`, then the first columns correspond to the parameters estimated by `fitX`, and the last columns correspond to the parameters estimated by the re-fitted model `fitY`. If `estmethod="g"`, then the first columns correspond to ψ, and the remaining columns correspond to the parameters estimated by `fitZ` and (if `link="logit"`) `fitY`.
- `d.estfun` the jacobian matrix of `colSums(estfunall)`.
- `converged` logical. Was a solution found to the estimating equations?
- `fitZ` the model `fitZ` used in the estimation process when `estmethod="g"`. This element is NULL when `estmethod="ts"`.
- `fitY` the re-fitted model `fitY` used in the estimation process when `estmethod="ts"`. This element is NULL when `estmethod="g"`.

Note

`ivglm` allows for weights. However, these are defined implicitly through the input models. Thus, when models are used as input to `ivglm`, these models have to be fitted with the same weights. When `estmethod="g"` the weights are taken from `fitZ`, so this must be specified.

Author(s)

Arvid Sjolander.

References

Bowden J., Vansteelandt S. (2011). Mendelian randomization analysis of case-control data using structural mean models. *Statistics in Medicine* **30**(6), 678-694.

Vansteelandt S., Bowden J., Babanezhad M., Goetghebeur E. (2011). On instrumental variables estimation of causal odds ratios. *Statistical Science* **26**(3), 403-422.

Examples

```
set.seed(9)

##Note: the parameter values in the examples below are chosen to make
##Y0 independent of Z, which is necessary for Z to be a valid instrument.

n <- 1000
psi0 <- 0.5
psi1 <- 0.2

##---Example 1: linear model and interaction between X and L---

L <- rnorm(n)
Z <- rnorm(n, mean=L)
X <- rnorm(n, mean=Z)
m0 <- X-Z+L
Y <- rnorm(n, mean=psi0*X+psi1*X*L+m0)
data <- data.frame(L, Z, X, Y)

#two-stage estimation
fitX <- glm(formula=X~Z, data=data)
fitY <- glm(formula=Y~X+L+X*L, data=data)
fitIV <- ivglm(estmethod="ts", fitX=fitX, fitY=fitY, data=data, ctrl=TRUE)
summary(fitIV)

#G-estimation
fitZ <- glm(formula=Z~L, data=data)
fitIV <- ivglm(estmethod="g", Z="Z", X="X", Y="Y", fitZ=fitZ, data=data,
  formula=~L, link="identity")
summary(fitIV)

##---Example 2: logistic model and no covariates---

Z <- rbinom(n, 1, 0.5)
X <- rbinom(n, 1, 0.7*Z+0.2*(1-Z))
m0 <- plogis(1+0.8*X-0.39*Z)
Y <- rbinom(n, 1, plogis(psi0*X+log(m0/(1-m0))))
data <- data.frame(Z, X, Y)

#two-stage estimation
fitX <- glm(formula=X~Z, family="binomial", data=data)
fitY <- glm(formula=Y~X, family="binomial", data=data)
fitIV <- ivglm(estmethod="ts", fitX=fitX, fitY=fitY, data=data, ctrl=TRUE)
summary(fitIV)

#G-estimation
fitY <- glm(formula=Y~X+Z+X*Z, family="binomial", data=data)
fitIV <- ivglm(estmethod="g", Z="Z", X="X", Y="Y", fitY=fitY, data=data, link="logit")
summary(fitIV)
```

`plot.estfun` *Plots sums of estimating functions.*

Description

This is a `plot` method for class `"estfun"`.

Usage

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

Arguments

`x`	an object of class `"estfun"`, fitted with `method="g"`.
`...`	additional arguments to `plot`.

Author(s)

Arvid Sjolander.

Examples

```
##See documentation for estfun.
```

`plot.ivah` *Plots result of G-estimation in causal AH model.*

Description

This is a `plot` method for class `"ivah"`. It only supports objects fitted with `estmethod="g"`.

Usage

```
## S3 method for class 'ivah'
plot(x, gof=FALSE, CI.level=0.95, ...)
```

Arguments

`x`	an object of class `"ivah"`, fitted with `estmethod="g"`.
`gof`	should we plot the goodness-of-fit process? If not, then $B(t)$ is plotted together with confidence intervals.
`CI.level`	level for the confidence intervals.
`...`	not used.

Author(s)

Arvid Sjolander and Torben Martinussen.

Examples

```
##See documentation for ivah.
```

---

print.ivmod	*Prints output of instrumental variable estimation*

Description

This is a `print` method for class `"ivmod"`.

Usage

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

Arguments

`x`	an object of class `"ivmod"`.
`digits`	the number of significant digits to use.
`...`	not used.

Author(s)

Arvid Sjolander

Examples

```
##See documentation for ivglm, ivcoxph and ivah.
```

print.summary.ivbounds
Prints summary of instrumental variable bounds

Description

This is a `print` method for class `"summary.ivbounds"`.

Usage

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

Arguments

`x`	an object of class `"summary.ivbounds"`.
`digits`	the number of significant digits to use.
`...`	not used.

Author(s)

Arvid Sjolander

Examples

```
##See documentation for ivbounds.
```

print.summary.ivmod
Prints summary of instrumental variable estimation

Description

This is a `print` method for class `"summary.ivmod"`.

Usage

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

Arguments

`x`	an object of class "`summary.ivmod`".
`digits`	the number of significant digits to use.
`signif.stars`	logical. If TRUE, "significance stars" are printed for each coefficient.
`...`	not used.

Author(s)

Arvid Sjolander

Examples

```
##See documentation for ivglm, ivcoxph and ivah.
```

`summary.ivbounds` *Summarizes instrumental variable estimation*

Description

This is a `summary` method for class "`ivbounds`".

Usage

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

Arguments

`object`	an object of class "`ivbounds`".
`...`	not used.

Details

Provides the lower and and upper bounds for

$$p_0 = p(Y_0 = 1)$$

$$p_1 = p(Y_1 = 1)$$

$$\text{CRD} = p_1 - p_0$$

$$\text{CRR} = p_1/p_0$$

$$\text{COR} = \frac{p_1/(1-p_1)}{p_0/(1-p_0)}$$

Author(s)

Arvid Sjolander

Examples

```
##See documentation for ivbounds.
```

`summary.ivmod` *Summarizes instrumental variable estimation*

Description

This is a `summary` method for class `"ivmod"`.

Usage

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

Arguments

`object`	an object of class `"ivmod"`.
`...`	not used.

Author(s)

Arvid Sjolander

Examples

```
##See documentation for ivglm, ivcoxph and ivah.
```

VitD *Data from a cohort study on Vitamin D and mortality.*

Description

This dataset originates from a real cohort study on Viatmin D and mortailty, described by Martinussen et al (2017). However, to allow public availability the data were slightly mutialted before inclusion in the `ivtools` package.

Usage

```
data(VitD)
```

Format

The dataset contains the following variables:

age age at baseline.

filaggrin binary indicator of whether the subject has mutations in the filaggrin gene.

vitd vitamin D level at baseline, measured as serum 25-OH-D (nmol/L).

time follow-up time.

death indicator of whether the subject died during follow-up.

References

Martinussen T., Sorensen D.D., Vansteelandt S. (2017). Instrumental variables estimation under a structural Cox model. *Biostatistics* <doi:10.1093/biostatistics/kxx057>.

Index