Package ‘ivregEX’

July 19, 2017

Type Package
Title Create Independent Evidence in IV Analyses and Do Sensitivity Analysis in Regression and IV Analysis
Version 1.0
Date 2017-07-18
Author Bikram Karmakar
Maintainer Bikram Karmakar <bikramk@wharton.upenn.edu>
Imports AER,Formula
Depends lmtest,sandwich
Description Allows you to create an evidence factor (EX analysis) in an instrumental variables regression model. Additionally, performs Sensitivity analysis for OLS analysis, 2SLS analysis and EX analysis with interpretable plotting and printing features.
License GPL (>= 2)
NeedsCompilation no
Repository CRAN
Date/Publication 2017-07-19 21:46:49 UTC

R topics documented:

ivregEX-package ........................................... 2
coefftestIV ............................................. 4
get_nmc .................................................... 5
ivreg.EX ..................................................... 6
ivreg.fit.EX ............................................... 8
plot.ivregDecision ....................................... 10
plot.sensIVregEX ......................................... 11
plot.si ...................................................... 12
plot_confregion ......................................... 13
print.ivregDecision ..................................... 14
print.si .................................................... 15
sens2SLS .................................................. 16
ivregEX-package

Create Independent Evidence in IV Analyses and Do Sensitivity Analysis in Regression and IV Analysis

Description

Allows you to create an evidence factor (EX analysis) in an instrumental variables regression model. Additionally, performs Sensitivity analysis for OLS analysis, 2SLS analysis and EX analysis with interpretable plotting and printing features.

Details

Index of help topics:

- coefestIV: Coefficient test of output of ivreg.EX.
- get_nrc: Find the grid structure in plotting
- ivreg.EX: Constructing independent evidence (EX estimates) in an IV regression
- ivreg.fit.EX: Fits the Two Stage Least Squared Estimates and EX estimates
- ivregEX-package: Create Independent Evidence in IV Analyses and Do Sensitivity Analysis in Regression and IV Analysis
- plot.ivregDecision: Plotting an object of class 'ivregDecision' or 'ivregDecisionList'
- plot.sensIVregEX: Plotting of Sensitivity Analyses
- plot.si: Plot Insensitive Region
- plot_confregion: Plot Confidence Region
- print.ivregDecision: Printing object of class "ivregDecision"
- print.si: Printing object of class "si"
- sens2SLS: Sensitivity Interval for Two Stage Least Squared Regression
- sensEX: Sensitivity Interval for Secondary EX Estimator
- sensIvreg: Sensitivity analysis in instrumental variables regression model
- sensOLS: Sensitivity Interval for Least Squared Estimator
- senslm: Sensitivity analysis of a least squared regression estimator
Author(s)

Bikram Karmakar
Maintainer: Bikram Karmakar <bikramk@wharton.upenn.edu>

References

~~ Literature or other references for background information ~~

Examples

data("CigarettesSW", package = "AER")
CigarettesSW$rprice <- with(CigarettesSW, price/cpi)
CigarettesSW$rincome <- with(CigarettesSW, income/population/cpi)
CigarettesSW$tdiff <- with(CigarettesSW, (taxs - tax)/cpi)

#---------------------#

lmfit <- lm(log(packs ~ log(rprice) + log(rincome), data = CigarettesSW, subset = year == "1995")
z <- senslm(lmfit, variable = 'log(rprice)')
plot(z)

#---------------------#

library(AER)
fm1 <- ivreg(log(packs) ~ log(rprice) + log(rincome) | tdiff + log(rincome),
data = CigarettesSW, subset = year == "1995", x=TRUE)
z <- sensivreg(fm1, variable = 'log(rprice)')
plot(z)

#---------------------#

fm2 <- ivreg.EX(log(packs) ~ log(rprice) + log(rincome) | tdiff + log(rincome),
data = CigarettesSW, subset = year == "1995")
fm3 <- iervative(log(packs) ~ log(rprice) + log(rincome), data = CigarettesSW,
subset = year == "1995")
res = coeftestIV(fm2, lmfit, 'log(rprice)')
z <- sensivreg(fm2, res, 'log(rprice)', effect = c(-.1, 0, .15, .17))
plot(z)

#---------------------#

x = CigarettesSW[CigarettesSW[, 'year'] == "1995", c('rprice', 'rincome')]
y = CigarettesSW[CigarettesSW[, 'year'] == "1995", c('packs')]
z = CigarettesSW[CigarettesSW[, 'year'] == "1995", c('tdiff', 'rincome')]
### coeftestIV

**Coefficient test of output of ivreg.EX.**

**Description**

For a given set of endogenous variables in an instrumental variables model coeftestIV provides you the coefficients matrix with estimates, std errs, t-ratios and p-values for both homoscedastic and heteroscedastic errors. Also, clustered standard error can be calculated.

**Usage**

```r
coeftestIV(ivregfit, lmfit, vars, cls = NULL, vcov)
```

**Arguments**

- **ivregfit**: An object of class 'ivregEX', an output of ivreg.EX or ivreg.fit.Ex function.
- **lmfit**: An object of class 'lm', an output of lmfit or lm function.
- **vars**: A vector of endogenous variables of the instrumental variables model for which the standard error calculation and significance test to be performed.
- **cls**: A numeric vector specifying the stratification strata structure of the model. Default setting of NULL assumes no stratification.
- **vcov**: A specification of the covariance matrix of the estimated coefficients. This can be specified as a matrix or as a function yielding a matrix when applied to x. Only used for the computation under heteroscedastic errors. Default is vcovHC.

**Details**

For details on vcov see the documentation of coeftest in "lmtest" package.

**Value**

An object of class 'coeftest'. A matrix of coefficient estimates, standard errors, t-ratios and p-values.

**Author(s)**

Bikram Karmakar
get_nrnc

See Also

coeftest

Examples

data("CigarettesSW", package = "AER")
CigarettesSW$rprice <- with(CigarettesSW, price/cpi)
CigarettesSW$rincome <- with(CigarettesSW, income/population/cpi)
CigarettesSW$tddiff <- with(CigarettesSW, (taxs - tax)/cpi)

fm2 <- ivreg.EX(log(packs) ~ log(rprice) + log(rincome) | tddiff + log(rincome),
data = CigarettesSW, subset = year == "1995")

lmfit <- lm(log(packs) ~ log(rprice) + log(rincome), data = CigarettesSW,
subset = year == "1995")

res = coeftestIV(fm2, lmfit, 'log(rprice)')

get_nrnc(m = 10, nc = -1, choices = 2:5)

Arguments

m Number of plots, a positive integer.
nc A preferred choice of number of columns. If set to -1, it is calculated based on
choices.
choices Various choices for nc.
Value
Two numbers for nr and nc in mrow argument of par().

Note
This is used in plotting results of sensitivity analyses, i.e., outputs of sensivreg or senslm.

Author(s)
Bikram Karmakar

Examples
## Not run:
get_nrnc(6)
get_nrnc(5, 3)
## End(Not run)

ivreg.EX  
Constructing independent evidence (EX estimates) in an IV regression

Description
This function uses the same arguments as ivreg and additionally outputs the EX analysis.

Usage
ivreg.EX(formula, instruments, data, subset, na.action, weights, offset, contrasts = NULL, model = TRUE, ...)

Arguments

formula, instruments
formula specification(s) of the regression relationship and the instruments. Either instruments is missing and formula has three parts as in y ~ x1 + x2 | z1 + z2 + z3 (recommended) or formula is y ~ x1 + x2 and instruments is a one-sided formula ~ z1 + z2 + z3 (only for backward compatibility).

data
an optional data frame containing the variables in the model. By default the variables are taken from the environment of the formula.

subset
an optional vector specifying a subset of observations to be used in fitting the model.

na.action
a function that indicates what should happen when the data contain NAs. The default is set by the na.action option.

weights
an optional vector of weights to be used in the fitting process.

offset
an optional offset that can be used to specify an a priori known component to be included during fitting.
ivreg.EX returns an object of class "ivregEX" (also of class "ivreg"), with the following components:

- **coefficients**: parameter estimates.
- **residuals**: a vector of residuals.
- **fitted.values**: a vector of predicted means.
- **weights**: either the vector of weights used (if any) or NULL (if none).
- **offset**: either the offset used (if any) or NULL (if none).
- **n**: number of observations.
- **nobs**: number of observations with non-zero weights.
- **rank**: the numeric rank of the fitted linear model.
- **df.residual**: residual degrees of freedom for fitted model.
- **cov.unscaled**: unscaled covariance matrix for the coefficients.
- **sigma**: residual standard error.
- **call**: the original function call.
- **formula**: the model formula.
- **terms**: a list with elements "regressors" and "instruments" containing the terms objects for the respective components.
- **levels**: levels of the categorical regressors.
- **contrasts**: the contrasts used for categorical regressors.
- **model**: the full model frame (if model = TRUE).
- **y**: the response vector.
- **x**: a list with elements "regressors", "instruments", "projected", containing the model matrices from the respective components. "projected" is the matrix of regressors projected on the image of the instruments.

See Also

ivreg.coeftestIV
Examples

```r
data("CigarettesSW", package = "AER")
CigarettesSW$rprice <- with(CigarettesSW, price/cpi)
CigarettesSW$rincome <- with(CigarettesSW, income/population/cpi)
CigarettesSW$tdiff <- with(CigarettesSW, (taxs - tax)/cpi)

fm2 <- ivreg.fit.EX(log(packs) ~ log(rprice) + log(rincome) | tdiff + log(rincome),
data = CigarettesSW, subset = year == "1995")
lmfit <- lm(log(packs) ~ log(rprice) + log(rincome), data = CigarettesSW, subset = year == "1995")
coeftestIV(fm2, lmfit, 'log(rprice)')
```

---

**ivreg.fit.EX**  
*Fits the Two Stage Least Squared Estimates and EX estimates*

### Description

Calculates the instrumental variables model when the output, regressors and instrument matrices are explicitly provided.

### Usage

```r
ivreg.fit.EX(x, y, z, weights, offset, ...)
```

### Arguments

- `x` regressor matrix.
- `y` vector with dependent variable.
- `z` instruments matrix.
- `weights` an optional vector of weights to be used in the fitting process.
- `offset` an optional offset that can be used to specify an a priori known component to be included during fitting.
- `...` further arguments passed to `lm.fit` or `link{stats}{lm.wfit}`, respectively.

### Details

See documentation of `ivreg.fit` for further operational details.

Statistical inference and sensitivity analysis based on the output of `ivreg.fit.Ex` can be performed using `coeftestIV` and `sens Ivreg` functions.
Value

`ivreg.fit.EX` returns an object of class `ivregEX` list with the following components:

- `coefficients`: parameter estimates.
- `residuals`: a vector of residuals.
- `fitted.values`: a vector of predicted means.
- `weights`: either the vector of weights used (if any) or NULL (if none).
- `offset`: either the offset used (if any) or NULL (if none).
- `estfun`: a matrix containing the empirical estimating functions.
- `n`: number of observations.
- `nobs`: number of observations with non-zero weights.
- `rank`: the numeric rank of the fitted linear model.
- `df.residual`: residual degrees of freedom for fitted model.
- `cov.unscaled`: unscaled covariance matrix for the coefficients.
- `sigma`: residual standard error.
- `y`: the response vector.
- `x`: a list with elements "regressors", "instruments", "projected", containing the model matrices from the respective components. "projected" is the matrix of regressors projected on the image of the instruments.
- `coefficientsEX`: parameter estimates of the EX analysis.
- `residualsEX`: vector of residuals in EX regression.
- `fitted.valuesEX`: vector of fitted values in EX regression.
- `vhat`: residuals of first stage of two stage least squared analysis.
- `vars`: A list of variable names. List has three vectors for three kinds of variables: 'exogenous', 'endogenous' and 'instruments'.

Author(s)

Bikram Karmakar

See Also

`ivreg.EX`, `coeftestIV`, `sensIvreg`, `ivreg.fit`

Examples

data("CigarettesSW", package = "AER")
CigarettesSW$rprice <- with(CigarettesSW, price/cpi)
CigarettesSW$rincome <- with(CigarettesSW, income/population/cpi)
CigarettesSW$tdiff <- with(CigarettesSW, (taxs - tax)/cpi)

dx = CigarettesSW[CigarettesSW[, 'year'] == "1995", c('rprice', 'rincome')]
y = CigarettesSW[CigarettesSW[, 'year'] == "1995", c('packs')]
z = CigarettesSW[CigarettesSW[, 'year'] == "1995", c('tdiff', 'rincome')]
fm3 <- ivreg.fit.EX(as.matrix(x), y, z)

---

### plot.ivregDecision

**Plotting an object of class 'ivregDecision' or 'ivregDecisionList'**

#### Description

An object of class `ivregDecisionList`, which is a list of objects of class `ivregDecision`, is output of `sensIvreg.ivregEX`. This function gives you freedom to stylize a gray-scale plot of a sensitivity analysis.

#### Usage

```r
## S3 method for class 'ivregDecision'
plot(x, xaxes = TRUE, yaxes = TRUE, xaxes.label = TRUE, yaxes.label = TRUE, legend = TRUE, ...)

## S3 method for class 'ivregDecisionList'
plot(x, xaxes = TRUE, yaxes = TRUE, xaxes.label = TRUE, yaxes.label = TRUE, legend = TRUE, subtitle = TRUE, nc = ..., ...)
```

#### Arguments

- **x**: An object of class `ivregDecision` or `ivregDecisionList`. A decision matrix (or list of matrices) for given treatment effect(s) for different values of the sensitivity parameters.
- **xaxes, yaxes, xaxes.label, yaxes.label**: Whether x-axis, y-axis and their labels to be added to the plots.
- **legend**: Whether a legend will be added to the plot.
- **subtitle**: Logical for whether a subtitle to be added in a plot indicating the effect size of a `ivregDecisionList` object.
- **nc**: Optional integer for the number of columns in the plot.
- **...**: Optional arguments for R plotting function.

#### Author(s)

Bikram Karmakar

#### See Also

- `plot.sensIvregEX`
plot.sensIvregEX

Examples

data("CigarettesSW", package = "AER")
CigarettesSW$rprice <- with(CigarettesSW, price/cpi)
CigarettesSW$rincome <- with(CigarettesSW, income/population/cpi)
CigarettesSW$tdiff <- with(CigarettesSW, (taxs - tax)/cpi)

fm2 <- ivreg.EX(log(packs) ~ log(rprice) + log(rincome) | tdiff + log(rincome),
data = CigarettesSW, subset = year == "1995")
lmfit <- lm(log(packs) ~ log(rprice) + log(rincome), data = CigarettesSW,
subset = year == "1995")
res = coeftestIV(fm2, lmfit, 'log(rprice)')

z <- sensivreg(fm2, res, 'log(rprice)', delta1 = seq(-.51, .51, .01),
effect = c(-.1, 0, .1, .15, .17))
plot(z$Z[[1]])
plot(z$Z, nc = 3)

plot.sensIvregEX  Plotting of Sensitivity Analyses

Description

Based on the output of sensIvreg creates plots of sensitive regions.

Usage

## S3 method for class 'sensIvregEX'
plot(x, effect = 0, ...)

Arguments

x An object of class 'sensIvregEX', output of sensIvreg.
effect Effect size (optional). Default value is 0.
... Optional arguments for R plotting features.

Note

This functionally creates two sets of plots, one a greyscale plot of the sensitivity analyses in the secondary instrumental variables analysis with independent EX analysis. Additionally it creates a plot of individual sensitivity regions for each analyses for the given effect size.

Author(s)

Bikram Karmakar
See Also

plot.ivregDecision.

Examples

data("CigarettesSW", package = "AER")
CigarettesSW$rprice <- with(CigarettesSW, price/cpi)
CigarettesSW$rincome <- with(CigarettesSW, income/population/cpi)
CigarettesSW$tdiff <- with(CigarettesSW, (taxs - tax)/cpi)

fm2 <- ivreg.EX(log(packs) ~ log(rprice) + log(rincome) | tdiff + log(rincome),
data = CigarettesSW, subset = year == "1995")
lmfit <- lm(log(packs) ~ log(rprice) + log(rincome), data = CigarettesSW, subset = year == "1995")
res = coefTestIV(fm2, lmfit, 'log(rprice)')
z <- sensivreg(fm2, res, 'log(rprice)', effect = c(-1, .0, .1, .15, .17), show = FALSE)
plot(z, nc = 3)

plot.si

Plot Insensitive Region

Description

This function provides a visual representation of the sensitivity intervals.

Usage

## S3 method for class 'si'
plot(x, effect = 0, ...)

Arguments

x An object of class 'si'; output of senslm and sensIvreg.ivreg.
effect Optional value of the null treatment effect.
... Optional arguments for the R plotting function.

Note

This function is called as a part of plotting object of class 'sensIvregEX'.

Author(s)

Bikram Karmakar
Examples

data("CigarettesSW", package = "AER")
CigarettesSW$rprice <- with(CigarettesSW, price/cpi)
CigarettesSW$rincome <- with(CigarettesSW, income/population/cpi)
CigarettesSW$tdiff <- with(CigarettesSW, (taxs - tax)/cpi)

library(AER)

lmfit <- lm(log(packs) ~ log(rprice) + log(rincome), data = CigarettesSW, subset = year == "1995")
z <- senslm(lmfit, variable = 'log(rprice)')
plot(z)

plot_confregion

Description

This function provides a visual representation of the confidence regions capturing the three analyses, OLS, 2SLS and EX.

Usage

plot_confregion(est, se, t, roundto = 3, xlims, ylims = c(0, .7), ypos = .35, xaxis = T, legend = T, xlab = 'Treatment Effect', xlab.cex = 1.3, oma = c(4.5, .1, 2, 0), ...)

Arguments

est          A numeric vector of length 3, for coefficient estimates of the three analyses - OLS, 2SLS and EX respectively.
se           A numeric vector of length 3, for standard errors of the three analyses - OLS, 2SLS and EX respectively.
t           A number to be used to calculate individual confidence interval by 'est ± t*se'. Default value used is 2. For different level of confidence corresponding value has to be provided.
roundto      The number of decimal digit precision.
xlims        Optional numeric array of size 2 for the plotting region of the treatment effect plotted.
ylims, ypos  Optional arguments. Can be used for plotting aesthetics.
xaxis, legend Logical arguments for whether the treatment effect axis to be plotted and legends shown.
xlab          A logical argument or a character vector for the xaxis label.
`print.ivregDecision`

```
xlab.cex    Size of xaxis label.
ona         Outer margin dimensions.
...         Other graphics parameters which are passed to the main plotting function.
```

**Note**

This function plots the confidence region of the treatment effect. The confidence region is calculated from the coefficient estimates and standard errors of the OLS, 2SLS and EX. This confidence region has overall level of confidence $(1 - \alpha) \times 100\%$ when each individual analyses are all at the same level of significance. Argument `t` should be used to specify the corresponding quantile.

Resultant confidence region has a four pieces: region where we have no evidence, region where only OLS shows evidence, region where OLS and one of other two estimators provides evidence, and the region where both the IV based estimators show evidence.

This plotting function creates a new plotting device for its plotting purpose.

**Author(s)**

Bikram Karmakar

**Examples**

```r
est = c(1.6858, 2.8912, 1.1435)
se = c(0.5029, 0.9785, 0.5377)

plot_confregion(est, se, roundto = 3, ypos = .35, xlab.cex = 1.2)
```

---

print.ivregDecision  *Printing object of class "ivregDecision"*

**Description**

"ivregDecisionList", a list of objects of class "ivregDecision", is part of an output of sensitivity analysis using function `sensIVreg`.

**Usage**

```r
## S3 method for class 'ivregDecision'
print(x, ...)
```
Arguments

x  object of class "ivregDecision".
... Ignore.

Note

Trivially applies to object of class "ivregDecisionList"

Author(s)

Bikram Karmakar

See Also

plotNivregdecision

Description

Printing feature of sensitivity intervals. Object of class "si" is found in output of functions sensivreg and senslm.

Usage

## S3 method for class 'si'
print(x, effect = 0, ...)

Arguments

x  object of class "si".
effect Null treatment effect.
... Ignore.

Author(s)

Bikram Karmakar

See Also

plot.si
This function performs the background sensitivity interval calculation for the 2SLS estimator in an instrumental variables regression model.

Usage

sens2SLS(delta2, df, est, se, thaty, rho2, alpha = 0.05, talpha)

Arguments

delta2: A numeric vector of values of the second sensitivity parameter. Default value seq(-.1, .1, .01). Only used for 2SLS estimator.
df, est, se: Degrees of freedom of the model; 2SLS estimate and its standard error.
thaty: t-statistic of the treatment in the regression of the instruments on the treatment.
rho2: squared partial correlation of g(instruments) and the treatment.
alpha: Significance level. Default at 0.05.
talpha: If the interval is to be adjusted for model selection etc, the multiplicative factor to be used. Optional.

Details

See the description of sensIVreg.

Value

Returns an object of class "si".
An object if class "si" is essentially a matrix of two columns for the the upper and lower sensitivity limits corresponding to the sensitivity parameters.

Note

Do not use this function by itself, use sensIVreg function.

Author(s)

Bikram Karmakar

See Also

sensIVreg, plot.si
Description
This function performs the background sensitivity interval calculation for the EX estimator in an instrumental variables regression model.

Usage
sensex(delta1, df, est, se, rhoYX1.X1sq, alpha = 0.05, talpha)

Arguments
delta1  A numeric vector of values of the first sensitivity parameter. Default value seq(-.1, .1, .01). Not used for 2SLS estimator.
df, est, se Degrees of freedom of the model; 2SLS estimate and its standard error.
rhoYX1.X1sq squared partial correlation of the instruments and the treatment.
alpha Significance level. Default at 0.05.
talpha If the interval is to be adjusted for model selection etc, the multiplicative factor to be used. Optional.

Details
See the description of sensivreg.

Value
Returns an object of class "si".
An object if class "si" is essentially a matrix of two columns for the the upper and lower sensitivity limits corresponding to the sensitivity parameters.

Note
Do not use this function by itself, use sensIVreg function.

Author(s)
Bikram Karmakar

See Also
sensIVreg, plot.si
sensIvreg

Sensitivity analysis in instrumental variables regression model

Description

This function performs sensitivity analysis for two stage least squared estimator. When output of OLS and EX analyses are also supplied as an object of class "ivregEX" all three sensitivity analyses for the three estimates are performed.

Usage

```r
# S3 method for class 'ivregEX'
sensIvreg(ivregfit, coeftestIVres, variable,
  delta1 = seq(-0.1, 0.1, 0.01), delta2 = seq(-0.1, 0.1, 0.01),
  effect = 0, g = sqrt, errType = c("homoscedastic", "heteroscedastic"),
  alpha = 0.05, show = FALSE, ...)
```

```r
# S3 method for class 'ivreg'
sensIvreg(ivregfit, coeftestIVres, variable,
  delta2 = seq(-.1, .1, .01),
  g = sqrt, errType = c("homoscedastic", "heteroscedastic"),
  alpha = 0.05, ...)
```

Arguments

- `ivregfit` An object of class `ivreg` or `ivregEX` respectively. `ivregEX` object is an output of `ivreg.EX` or `ivreg.fit.EX`. `ivreg` object is an output of `ivreg`.
- `coeftestIVres` An object of class "coeftest". Must be supplied if `ivregfitEX` as the output of `coeftestIV`. Optional for only a two stage least squared estimator.
- `variable` The endogenous variable of interest in the instrumental variables regression model.
- `delta1` A numeric vector of values of the first sensitivity parameter. Default value `seq(-.1, .1, .01)`. Not used for 2SLS estimator.
- `delta2` A numeric vector of values of the second sensitivity parameter. Default value `seq(-.1, .1, .01)`. Only used for 2SLS estimator.
- `effect` A numeric vector of null treat effects of interest. Default 0.
- `g` A function describing the mechanism by which the assumptions of an instrument may be violated. Default `sqrt`.
- `errType` What kind of error structure to be assumed: "homoscedastic" or "heteroscedastic".
- `alpha` Significance level. Default at 0.05.
- `show` Whether result is to be printed.
- `...` Optional. May be ignored. used in `sensOLS, sens2SLS` or `sensex`
**Details**

delta1 and delta2 are the two sensitivity parameters. delta1 is the partial correlation of the unobservable and the treatment. It measures the amount of violation in the assumption of no-unmeasured confounding. delta2 is the partial correlation of the outcome and g(instruments) measuring the amount of violation in the exogeneity and no direct effect assumption of the instruments.

Only the 2SLS analysis uses delta2, OLS and EX analysis uses delta1 to perform sensitivity analysis.

**Value**

Returns an object of class "sensIvregEX" if input is of type ivregEX. If the input is of type ivreg, returns an object of class "si".

An object if class "si" is essentially a matrix of two columns for the the upper and lower sensitivity limits corresponding to the sensitivity parameters.

An object of class "sensIvregEX" consists of an object of class "ivregDecisionList" and three objects of class "si" for the three estimators. "ivregDecisionList" is a list of overall sensitivity analyses (based on primary OLS and secondary 2SLS and EX analysis. Each list element correspond to the deferent effect values and tells you the sensitivity of the problem for all possible pairs of (delta1, delta2).

**Author(s)**

Bikram Karmakar

**See Also**

senslm, plot.sensIvregEX, plot.si

**Examples**

data("CigarettesSW", package = "AER")
CigarettesSW$rprice <- with(CigarettesSW, price/cpi)
CigarettesSW$rincome <- with(CigarettesSW, income/population/cpi)
CigarettesSW$tdiff <- with(CigarettesSW, (taxs - tax)/cpi)

library(AER)

fm1 <- ivreg(log(packs) ~ log(rprice) + log(rincome) | tdiff + log(rincome),
data = CigarettesSW, subset = year == "1995", x=TRUE)
z <- sensIvreg(fm1, variable = 'log(rprice)')

fm2 <- ivreg.EX(log(packs) ~ log(rprice) + log(rincome) | tdiff + log(rincome),
data = CigarettesSW, subset = year == "1995")
```r
lmfit <- lm(log(packs) ~ log(rprice) + log(rincome), data = CigarettesSW, subset = year == "1995")
res = coeftestIV(fm2, lmfit, 'log(rprice)')

z <- sensIVreg(fm2, res, 'log(rprice)', effect = c(-.1, 0, .1, .15, .17))
```

---

**senslm**

*Sensitivity analysis of a least squared regression estimator*

**Description**

Performs a sensitivity analysis of the least squared regression

**Usage**

```r
senslm(lmfit, coefmat, variable, delta1 = seq(-0.1, 0.1, 0.01), alpha = 0.05, ...)
```

**Arguments**

- `lmfit`: An object of class "lm".
- `coefmat`: Optional matrix of class "coeftest" an output of coeftest.
- `variable`: Treatment variable of interest.
- `delta1`: A numeric vector of values of the first sensitivity parameter. Default value `seq(-.1, .1, .01)`.
- `alpha`: Significance level. Default at 0.05.
- `...`: May be ignored, used in sensOLS.

**Details**

delta1 is the partial correlation of the unobservable and the treatment. It measures the amount of violation in the assumption of no-unmeasured confounding.
Value

Returns an object of class "si".
An object of class "si" is essentially a matrix of two columns for the the upper and lower sensitivity limits corresponding to the sensitivity parameters.

Author(s)

Bikram Karmakar

See Also

sensIVreg, plot.sensIVregEX, plot.si

Examples

data("CigarettesSW", package = "AER")
CigarettesSW$rprice <- with(CigarettesSW, price/cpi)
CigarettesSW$rincome <- with(CigarettesSW, income/population/cpi)
CigarettesSW$tdiff <- with(CigarettesSW, (taxs - tax)/cpi)

lmfit <- lm(log(packs) ~ log(rprice) + log(rincome), data = CigarettesSW, subset = year == "1995")
z <- senslm(lmfit, variable = 'log(rprice)')

---

sensOLS

Sensitivity Interval for Least Squared Estimator

Description

This function performs the background sensitivity interval calculation for the OLS estimator in a regression model.

Usage

sensOLS(delta1, df, est, se, alpha = 0.05, talpha)

Arguments

delta1
A numeric vector of values of the first sensitivity parameter. Default value seq(-.1, .1, .01). Not used for 2SLS estimator.
df, est, se
Degrees of freedom of the model; 2SLS estimate and its standard error.
alpha
Significance level. Default at 0.05.
talpha
If the interval is to be adjusted for model selection etc, the multiplicative factor to be used. Optional.
Details

See the description of `senslm`.

Value

Returns an object of class "si".

An object if class "si" is essentially a matrix of two columns for the upper and lower sensitivity limits corresponding to the sensitivity parameters.

Note

Do not use this function by itself, use `senslm` function.

Author(s)

Bikram Karmakar

See Also

`senslm, plot.si`
Index

*Topic **Confidence Region**
  plot_confregion, 13

*Topic **IV Model**
  ivreg.EX, 6
  ivreg.fit.EX, 8
  sensIVreg, 18

*Topic **Plotting**
  get_nrnc, 5
  plot.ivregDecision, 10
  plot.sensIVregEX, 11
  plot.si, 12
  plot_confregion, 13

*Topic **Sensitivity Analysis**
  plot.ivregDecision, 10
  plot.sensIVregEX, 11
  print.ivregDecision, 14
  print.si, 15
  sens2SLS, 16
  sensEX, 17
  sensIVreg, 18
  senslm, 20
  sensOLS, 21

*Topic **Sensitivity Interval**
  plot.sensIVregEX, 11
  plot.si, 12
  print.si, 15
  sens2SLS, 16
  sensEX, 17
  sensIVreg, 18
  senslm, 20
  sensOLS, 21

*Topic **htest**
  coeftestIV, 4

*Topic **sandwich method**
  coeftestIV, 4
  coeftest, 5, 20
  coeftestIV, 4, 7, 9
  get_nrnc, 5
  ivreg, 6, 7, 18
  ivreg.EX, 6, 9, 18
  ivreg.fit, 8, 9
  ivreg.fit.EX, 7, 8, 18
  ivregEX (ivregEX-package), 2
  ivregEX-package, 2
  lm.fit, 8
  model.matrix.default, 7
  plot.ivregDecision, 10, 12, 15
  plot.ivregDecisionList
    (plot.ivregDecision), 10
  plot.sensIVregEX, 10, 11, 19, 21
  plot.si, 12, 15–17, 19, 21, 22
  plot_confregion, 13
  print.ivregDecision, 14
  print.si, 15
  sens2SLS, 16, 18
  sensEX, 17, 18
  sensIVreg, 9, 16, 17, 18, 21
  senslm, 19, 20, 22
  sensOLS, 18, 20, 21