Package ‘icsw’

October 13, 2022

Type Package

Title Inverse Compliance Score Weighting

Version 1.0.0

Date 2018-05-16

Author Peter M. Aronow <peter.aronow@yale.edu>, Dean Eckles <icsw@deaneckles.com> and Kyle Peyton <kyle.peyton@yale.edu>

Maintainer Kyle Peyton <kyle.peyton@yale.edu>

Description Provides the necessary tools to estimate average treatment effects with an instrumental variable by re-weighting observations using a model of compliance.

Depends R (>= 3.0.0)

Imports stats

Suggests rgenoud, minqa

License MIT + file LICENSE

LazyLoad yes

RoxygenNote 6.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2018-05-16 22:44:58 UTC

R topics documented:

icsw-package	*Inverse compliance score weighting*

Description

Inverse compliance score weighting is a tool for estimating average treatment effects from local average treatment effects by weighting units using inverse probabilities of complying with an encouragement (instrument).

Details

Package:	icsw
Type:	Package
Version:	1.0.0
Date:	2018-05-16
License:	MIT
LazyLoad:	yes

Author(s)

Peter M. Aronow <peter.aronow@yale.edu>; Dean Eckles <icsw@deaneckles.com>; Kyle Peyton <kyle.peyton@yale.edu>

Maintainer: Kyle Peyton <kyle.peyton@yale.edu>

References

Bethany Albertson and Adria Lawrence. (2009). After the credits roll: The long-term effects of educational television on public knowledge and attitudes. *American Politics Research*. 37(2): 275-300.

Peter M. Aronow and Allison Carnegie. (2013). Beyond LATE: Estimation of the average treatment effect with an instrumental variable. *Political Analysis*. 21.4 (2013): 492-506.

Peter M. Aronow and Allison Carnegie. (2013). Replication data for: Beyond LATE: Estimation of the average treatment effect with an instrumental variable. *Dataverse Network.* http://hdl.handle.net/1902.1/21729 (accessed May 14, 2015).

clip.small.probs	*Replace probabilities below threshold with threshold value*

Description

Return the provided vector with values smaller than the provided threshold replaced with that threshold (i.e., clip the probabilities below a certain value). If the threshold is chosen to match an empirical quantile then this can be used to implement Winsorizing probabilities from below. If no threshold is provided, the smallest value greater than zero is used.

Usage

```
clip.small.probs(x, min.prob = NULL)
```

Arguments

`x`	Vector of probabilities.
`min.prob`	Threshold. Values smaller than `min.prob` are replaced with `min.prob`. If not provided, set to the smallest value in `x` greater than 0.

Details

Used to avoid small probabilities blowing up in inverse probability weighting.

Produces warnings whenever values are actually replaced.

Value

Vector the same length as `x` with, possibly, some entries replaced.

Author(s)

Peter M. Aronow <peter.aronow@yale.edu>, Dean Eckles <icsw@deaneckles.com>, Kyle Peyton <kyle.peyton@yale.edu>

Examples

```
probs <- seq(0, .01, by = .001)
min(clip.small.probs(probs, .05))

# without min.prob, uses smallest value > 0
min(clip.small.probs(probs))
```

compliance.score — *Estimate compliance scores using covariates.*

Description

Estimate probability that units are compliers using generalized linear models for probability of being a complier or always-taker and for being an always-taker conditional on being a complier or always-taker. These compliance scores can be used in inverse probability weighting to estimate average treatment effects. In the case of one-sided non-compliance, this estimation is considerably simpler.

Usage

```
compliance.score(D, Z, W, weights = NULL,
  link = qnorm, inv.link = pnorm, genoud = TRUE,
  num.iter = ifelse(genoud, 200, 10000),
  one.sided = FALSE)
```

Arguments

`D`	Binary treatment of interest.
`Z`	Binary instrument.
`W`	Matrix of covariates for compliance model.
`weights`	Observation weights.
`link`	Link function applied for linear models. Defaults to probit link function.
`inv.link`	Inverse link function (i.e., mean function) applied for linear models. Defaults to probit mean function.
`genoud`	Whether to use global optimization via genetic optimization from package `rgenoud`. Default, and highly recommended because the likelihood need not be concave. Otherwise, use the BOBYQA algorithm for constrained optimization from package `minqa`.
`num.iter`	Number of iterations of optimization routine.
`one.sided`	Whether non-compliance is one-sided (logical). When compliance is one-sided, the previous four arguments are ignored, and the compliance scores are estimated with probit regression.

Details

A unit i is a complier if $D_{i1} > D_{i0}$, where D_{i1} and D_{i0} are the potential treatments for unit i when Z is set to 1 and 0. This is a latent (unobserved) characteristic of individual units, since each unit is only observed with one value of Z.

By default this function uses genetic optimization via `genoud` because the loss function for the complier scores is not necessarily convex.

Value

Vector of estimated probabilities of being a complier (i.e., compliance scores).

Note

Requires `rgenoud` package if genoud = TRUE. Requires `minqa` package if genoud = FALSE.

Author(s)

Peter M. Aronow <peter.aronow@yale.edu>, Dean Eckles <icsw@deaneckles.com>, Kyle Peyton <kyle.peyton@yale.edu>

References

Peter M. Aronow and Allison Carnegie. (2013). Beyond LATE: Estimation of the average treatment effect with an instrumental variable. *Political Analysis*.

See Also

Used by `icsw.tsls`.

Examples

```
# Load example dataset, see help(FoxDebate) for details
data(FoxDebate)

# Matrix of covariates
covmat <- with(FoxDebate, cbind(partyid, pnintst, watchnat, educad, readnews,
                                gender, income, white))

# Estimate compliance scores with covariates, assuming (default)
#   case of two-sided non-compliance
cscoreout <- with(FoxDebate, compliance.score(D = watchpro, Z = conditn,
                                              W = covmat))

# Extract vector of estimated compliance scores
cscore <- cscoreout$C.score
summary(cscore)
```

`FoxDebate` *The effects of watching a Fox debate on Proposition 209.*

Description

The data set (n=507) contains findings from the experiment described in Albertson and Lawrence (2009) in which a representative sample of survey respondents in Orange County, California, were randomly assigned to receive encouragement to view a Fox debate on affirmative action, which would take place on the eve of the 1996 presidential election. Shortly after the election, these respondents were reinterviewed. The postelection questionnaire asked respondents whether they viewed the debate, whether they supported a California proposition (209) to eliminate affirmative action (`support`), and how informed they felt about the proposition (`infopro`). The dataset can be used to reproduce Table 2 in Aronow and Carnegie (2013). Note that mean imputation was used to handle missing data so non-integer values are imputed. `support` and `infopro` are excepted and include missing values.

Usage

```
data(FoxDebate)
```

Format

A data frame with 507 observations on the following 11 variables:

- `partyid` An 11 point scale from “strong Republican” to “strong Democrat”.
- `pnintst` Respondent interest in politics and national affairs. Coded 1 = "very interested", 2 = "somewhat interested", 3 = "only slightly interested", 4 = "not interested at all".
- `watchnat` Frequency of national television news consumption. Coded 1 = "never"", 2 = "less than once a month"", 3 = "once a month", 4 = "several times a month", 5 = "once a week", 6 = "several times a week", 7 = "every day".
- `educad` Education level of respondent. Coded 1 = "eighth grade or less", 2 = "beyond eighth grade, not high school", 3 = "ged", 4 = "high school", 5 = "less than one year vocational school", 6 = "one to two year vocational school", 7 = "two years or more vocational school", 8 = "less than two years of college", 9 = "two or more years of college", 10 = "finished a two-year college program", 11 = "finished a four-year college program", 12 = "master degree or equivalent", 13 = "ph.d., m.d., or other advance degree".
- `readnews` How often respondent reads political news. Coded 1 = "never"", 2 = "less than once a month"", 3 = "once a month", 4 = "several times a month", 5 = "once a week", 6 = "several times a week", 7 = "every day".
- `gender` Respondent gender. Coded 1 for female and 0 for male.
- `income` Family income from all sources. Coded 1 = "under $10,000", 2 = "between $10,000 and $20,000", 3 = "between $20,000 and $30,000", 4 = "between $30,000 and $40,000", 5 = "between $40,000 and $50,000", 6 = "between $50,000 and $60,000", 7 = "between $60,000 and $70,000", 8 = "between $70,000 and $80,000", 9 = "between $80,000 and $90,000", 10 = "between $90,000 and $100,000", 11 = "$100,000 or more".
- `white` Binary indicator coded 1 if subject is white and 0 otherwise.
- `support` Support for Proposition 209. Coded 1 if subject voted against or opposed and 0 if subject voted for or favored
- `infopro` Information on Proposition 209. Coded from 1 to 4, with 4 meaning respondents had a great deal of information about Proposition 209 prior to the election, and 1 meaning respondents reported no information about the proposition before the election.

`watchpro` Binary indicator coded 1 if subject watched the Fox Debate about affirmative action and 0 otherwise. This is the outcome ("treatment") of interest.

`conditn` Binary indicator coded 1 if subject was (randomly) prompted to watch the Fox Debate about affirmative action. This is the encouragement (instrumental) variable.

References

Bethany Albertson and Adria Lawrence. (2009). After the credits roll: The long-term effects of educational television on public knowledge and attitudes. *American Politics Research*. 37(2): 275-300.

Peter M. Aronow and Allison Carnegie. (2013). Beyond LATE: Estimation of the average treatment effect with an instrumental variable. *Political Analysis*. 21.4 (2013): 492-506.

Peter M. Aronow and Allison Carnegie. (2013). Replication data for: Beyond LATE: Estimation of the average treatment effect with an instrumental variable. *Dataverse Network.* http://hdl.handle.net/1902.1/21729 (accessed May 14, 2015).

`icsw.tsls` *Two-stage least squares with inverse complier score weighting*

Description

Estimate average treatment effects using two-stage least squares with a binary instrument and binary treatment and weighting with inverse complier scores (probabilities of compliance). Optionally, bootstrap the entire estimation process for the purpose of hypothesis testing and constructing confidence intervals.

Usage

```
icsw.tsls(D, X, Y, Z, W, weights = NULL,
  R = 0, estimand = c("ATE", "ATT"),
  min.prob.quantile = NULL,
  min.prob = NULL, ...)

icsw.tsls.fit(D, X, Y, Z, W, weights,
  estimand = c("ATE", "ATT"),
  min.prob.quantile = NULL,
  min.prob = NULL, ...)
```

Arguments

`D`	Binary treatment of interest.
`X`	Matrix of covariates for two-stage least squares. Add a constant if desired (see examples).
`Y`	Outcome.
`Z`	Binary instrument.

`W`	Matrix of covariates for compliance model.
`weights`	Observation weights.
`R`	Number of bootstrap replicates.
`estimand`	Whether to estimate average treatment effect (default) or average treatment effect on the treated.
`min.prob.quantile`	Compliance scores are truncated to this quantile of positive compliance scores.
`min.prob`	Compliance scores are truncated to this value.
`...`	Additional arguments to `compliance.score`.

Value

If R = 0 or icsw.tsls.fit is called directly, a model fit, as described in `lm.wfit`.

If R > 0, a list with elements

`fitted.model`	A model fit, as returned by `lm.wfit`.
`coefs.boot`	p by R matrix of model coefficients for each of R bootstrap replicates.
`coefs.se.boot`	Vector of standard deviations of coefficients under bootstrap resampling (i.e., bootstrap standard errors).

Author(s)

Peter M. Aronow <peter.aronow@yale.edu>; Dean Eckles <icsw@deaneckles.com>; Kyle Peyton <kyle.peyton@yale.edu>

References

Peter M. Aronow and Allison Carnegie. (2013). Beyond LATE: Estimation of the average treatment effect with an instrumental variable. *Political Analysis*.

See Also

`compliance.score` for calculating compliance scores used in example.

`tsls.wfit` for regression via 2SLS with weights.

Examples

```
# Load example dataset, see help(FoxDebate) for details
data(FoxDebate)

# Ipw reweighting step Aronow and Carnegie (2013) use for missing data
covmat <- with(FoxDebate, cbind(partyid, pnintst, watchnat, educad, readnews, gender,
                                income, white))

# IPW reweighting step Aronow and Carnegie (2013) use for missing data
Ymis <- is.na(FoxDebate[, "infopro"])
```

```
IPWweight <- 1 / (1 - predict(glm(Ymis ~ covmat, family = binomial(link = "probit")),
                                  type = "response"))
IPWweight[Ymis] <- 0

N <- length(FoxDebate[, "infopro"])
alpha <- 0.275

# Compute the ATE of watching the Fox Debate on knowledge . This replicates the
#   ATE from column 2 of Table 1 in Aronow and Carnegie (2013)
icsw.out <- with(FoxDebate, icsw.tsls(D = watchpro, X = cbind(1, covmat),
                                          Y = infopro, Z = conditn, W = covmat,
                                          min.prob.quantile = 1 / (N^alpha),
                          weights = IPWweight))
round(icsw.out$coefficients["D"], 2)

# Example with bootstrap (this takes awhile!)
icsw.out <- with(FoxDebate, icsw.tsls(D = watchpro, X = cbind(1, covmat),
                                      Y = infopro, Z = conditn, W = covmat,
                                      min.prob.quantile = 1 / (N^alpha),
                                      weights = IPWweight, R = 1000))

# Display vector of coefficients
icsw.out$coefficients

# Display vector of (bootstrapped) SEs
icsw.out$coefs.se.boot
```

`tsls.wfit` *Fit instrumental variables model via two-stage least squares, with weights.*

Description

Fits linear first- and second-stage models using weighted linear regression.

Usage

```
tsls.wfit(X, Y, Z, weights, ...)
```

Arguments

`X`	Matrix of covariates and treatment, including constant if intercept is desired.
`Y`	Vector outcome.
`Z`	Matrix of covariates and instrument, including constant if intercept is desired.
`weights`	Observation weights.
`...`	Additional arguments to `lm.wfit`.

Value

A list with the second stage model fit, as returned by `lm.wfit`.

Author(s)

Peter Aronow <peter.aronow@yale.edu>, Dean Eckles <dean@deaneckles.com>, Kyle Peyton <kyle.peyton@yale.edu>

References

Peter M. Aronow and Allison Carnegie. (2013). Beyond LATE: Estimation of the average treatment effect with an instrumental variable. *Political Analysis*.

See Also

`lm.wfit`.

`compliance.score` for calculating compliance scores used in example.

Examples

```
# Load example dataset, see help(FoxDebate) for details
data(FoxDebate)

# Estimate compliance scores with covariates, assuming (default)
#   case of two-sided non-compliance.
covmat <- with(FoxDebate, cbind(partyid, pnintst, watchnat, educad, readnews, gender,
                                income, white))

cscoreout <- with(FoxDebate, compliance.score(D = watchpro, Z = conditn,
                              W = covmat))

# Extract vector of estimated compliance scores
cscore <- cscoreout$C.score

# Winsorising as described in Aronow and Carnegie (2013)
N <- length(FoxDebate[, "infopro"])
alpha <- 0.275
qcscore <- quantile(cscore, 1 / (N^alpha))
cscore[cscore < qcscore] <- qcscore

# IPW reweighting step Aronow and Carnegie (2013) use for missing data
Ymis <- is.na(FoxDebate[,"infopro"])

IPWweight <- 1 / (1 - predict(glm(Ymis ~ covmat, family = binomial(link = "probit")),
                              type = "response"))
IPWweight[Ymis] <- 0

# Compute LATE via 2SLS with IPW weights. This replicates
#   Table 2 Column 1 in Aronow and Carnegie (2013)
outputTSLS <- with(FoxDebate, tsls.wfit(X = cbind(1, covmat, watchpro), Y = infopro,
                        Z = cbind(1, covmat, conditn), weights = IPWweight))
round(outputTSLS$coefficients, 2)
```

```
# Compute ATE via 2SLS with (IPW weights / compliance scores). This
#   replicates Table 2 Column 2 in Aronow and Carnegie (2013)
outputICSW <- with(FoxDebate, tsls.wfit(cbind(1, watchpro, covmat), infopro,
                        cbind(1, conditn, covmat), w = IPWweight / cscore))
round(outputICSW$coefficients, 2)
```

Index