Package ‘intReg’

February 20, 2015

Version 0.2-8
Date 2012-01-12
Title Interval Regression
Author Ott Toomet <otoomet@gmail.com>
Maintainer Ott Toomet <otoomet@gmail.com>
Depends R (>= 2.7.0), miscTools, maxLik (>= 1.0-3)
Imports sets
Suggests Ecdat
Description Estimating interval regression models. Supports both common and observation-specific boundaries.
License GPL (>= 2)
URL http://www.sampleSelection.org
NeedsCompilation no
Repository CRAN
Date/Publication 2015-01-04 23:52:18

R topics documented:

  coef.intReg .................................................. 2
  intReg ......................................................... 3
  model.frame.intReg .......................................... 6
  model.matrix.intReg ......................................... 7
  nObs.intReg .................................................. 7

Index 9
 coef.intReg  

Extract (only informative) coefficients, standard errors, and variance-covariance matrix from `intReg` model.

**Description**

Internally, the interval boundaries are treated as fixed parameters. These methods extract only the informative ones (by default), i.e. they do not return the interval boundaries.

**Usage**

```r
## S3 method for class 'intReg'
coef(object, boundaries = FALSE, ...)
## S3 method for class 'intReg'
stdErr(x, boundaries = FALSE, ...)
## S3 method for class 'intReg'
summary(object, boundaries = FALSE, ...)
## S3 method for class 'intReg'
vcov(object, boundaries = FALSE, ...)
```

**Arguments**

- `object`: object of class `intReg`, estimated interval regression model
- `boundaries`: logical, whether to return (fixed) interval boundary parameters
- `...`: arguments for other methods

**Value**

a named numeric vector or matrix, for the estimated coefficients and standard errors, and variance-covariance matrix respectively.

**Author(s)**

Ott Toomet <otoomet@gmail.com>

**See Also**

`summary.intReg` which provides related functionality.

**Examples**

```r
## Example of observation-specific boundaries
## Estimate the willingness to pay for the Kakadu National Park
## Data given in intervals -- 'lower' for lower bound and 'upper' for upper bound.
## Note that dichotomous-choice answers are already coded to 'lower' and 'upper'
data(Kakadu, package="Ecdat")
set.seed(1)
Kakadu <- Kakadu[sample(nrow(Kakadu), 400),]
```
# subsample to speed up the estimation

```r
lb <- log(Kakadu$lower)
ub <- Kakadu$upper
ub[ub > 998] <- Inf
ub <- log(ub)
y <- cbind(lb, ub)
m <- intReg(y ~ sex + log(income) + age + schooling +
  recparks + jobs + lowrisk + wildlife + future + aboriginal + finben +
  mineparks + moreparks + gov +
  envcon + vpark + tenv + major, data=Kakadu)

# You may want to compare the results to Werner (1999),
# Journal of Business and Economics Statistics 17(4), pp 479-486
print(coef(m))
print(coef(m, boundaries=TRUE))
print(nObs(m))
```

---

**intReg**  
*Interval Regression*

## Description

This function estimates interval regression using either common intervals for all observations, or observation-specific intervals. Normal, logistic, log-log, and Cauchy disturbances are supported.

## Usage

```r
intReg(formula, start, boundaries, ..., 
  contrasts = NULL, Hess = FALSE, model = TRUE, 
  method = c("probit", "logistic", "cloglog", "cauchit", "model.frame"), print.level = 0, 
  data, subset, weights, na.action, 
  iterlim=100)
```

## Arguments

- **formula**: an object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted. The left-hand-side variable must be either a factor, describing the intervals where the observations fall to, or a matrix of two columns, describing the interval boundaries for each observation. See details below, see also `lm` for explanation how to write formulas.

- **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 `intReg` is called.

- **weights**: an optional vector of weights to be used in the fitting process. Should be `NULL` or a numeric vector. If non-NULL, weighted likelihood is maximized (that is, maximizing `sum(w*loglik)`).
**intReg**

- **start**: Initial values for the optimization algorithm. If ‘NULL’, initial values are calculated based on interval means. Note that intReg expects the full-length initial values, including the parameter boundaries and the standard deviation of the error term. See the example below.
- **boundaries**: Boundaries for intervals. See details.
- **...**: Further arguments to `maxLik`
- **subset**: An optional logical vector specifying a subset of observations to be used in the fitting process.
- **na.action**: A function which indicates what should happen when the data contain ‘NA’s. The default is set by the ‘na.action’ setting of sQuoteoptions, and is ‘na.fail’ if that is unset. The “factory-fresh” default is ‘na.omit’. Another possible value is ‘NULL’, no action. Value ‘na.exclude’ can be useful.
- **contrasts**: An optional list. See the ‘contrasts.arg’ of `model.matrix.default`.
- **Hess**: Should Hessian of the model be returned.
- **model**: Logical for whether the model matrix should be returned.
- **method**: Character, distribution of disturbances or ‘model.frame’. ‘probit’, ‘logistic’, ‘cloglog’ and ‘cauchit’ assume these disturbance distributions. ‘model.frame’ returns the model frame. The default value ‘probit’ assumes normal distribution.
- **print.level**: Output of run-time information: higher level prints more.
- **iterlim**: Maximum number of optimization iterations.

**Details**

Interval regression is a form of linear regression where only intervals (i.e. numeric upper and lower bounds) where the observations fall are visible of the otherwise continuous outcome variable. The current implementation assumes known distribution of the error term (default is normal) and estimates the model using Maximum Likelihood.

The intervals may be specified in two ways: either common intervals for all the observations by using argument ‘boundaries’, or by specifying the response as Nx2 matrix, where columns correspond to the lower- and upper bound for the individual observations. If you specify the common boundaries, you may want to give names to the components of ‘boundaries’. Otherwise they will be names ‘Boundary 1’, ‘Boundary 2’ etc. These names will appear to the estimation results. For observations-specific boundaries, the names are generated automatically.

**Value**

Object of class ‘intReg’ which inherits from class ‘maxLik’.

There are several methods, including `summary` and `coef`, partly inherited from ”maxLik” class.

Note that the boundaries are passed as fixed parameters to the maxLik estimation routine and hence returned as fixed estimates with standard errors set to zero.

**Note**

A lot of code is borrowed from `polr` from the package MASS
intReg

Author(s)

Ott Toomet <otoomet@gmail.com>

See Also

polr

Examples

## Example of observation-specific boundaries
## Estimate the willingness to pay for the Kakadu National Park
## Data given in intervals -- 'lower' for lower bound and 'upper' for upper bound.
## Note that dichotomous-choice answers are already coded to 'lower' and 'upper'
data(Kakadu, package="Ecdat")
set.seed(1)
Kakadu <- sample(nrow(Kakadu), 500),]
    # subsample to speed up the estimation
## Estimate in log form, change 999 to Inf
lb <- log(Kakadu$lower)
ub <- Kakadu$upper
ub[ub > 998] <- Inf
ub <- log(ub)
y <- cbind(lb, ub)
m <- intReg(y ~ sex + log(income) + age + schooling +
            recparks + jobs + lowrisk + wildlife + future + aboriginal + finben +
            mineparks + moreparks + gov +
            envcon + vpark + tenv + major, data=Kakadu)
## You may want to compare the results to Werner (1999),
## Journal of Business and Economics Statistics 17(4), pp 479-486
print(summary(m))
##
## Example of setting initial values
##
## st <- coef(m, boundaries=TRUE)
st[1:19] <- 1 # set all coefficients to 1
st["sigma"] <- 1 # set standard deviation to 1
m <- intReg(y ~ sex + log(income) + age + schooling +
            recparks + jobs + lowrisk + wildlife + future + aboriginal + finben +
            mineparks + moreparks + gov +
            envcon + vpark + tenv + major,
            start=st,
data=Kakadu)
## Note: the results will be the same as above
##
## Example of common intervals for all the observations
##
library(Ecdat)
data(bwages)
## calculate an ordinary Mincer-style wage regression. First by OLS and
## thereafter cut the wage to intervals and estimate with 'intReg'
## Note: gross hourly wage rate in EUR
ols <- lm(log(wage) ~ factor(educ) + poly(exper, 2), data=bwages)
cat("OLS estimate:\n")
print(summary(ols))
## Now we censor the wages to intervals
intervals <- c(0, 5, 10, 15, 25, Inf)
salary <- cut(bwages$wage, intervals)
int <- intReg(salary ~ factor(educ) + poly(exper, 2), data=bwages, boundaries=log(intervals))
## Note: use logs for the intervals in Euros. We do not have to
## transform salaris to log form as this does not change the intervals.
## Ignore any warnings
print(summary(int))

---

**model.frame.intReg**

*Data used for estimating interval regression model*

**Description**

Return the variables used for estimating an interval regression model

**Usage**

```r
## S3 method for class 'intReg'
model.frame( formula, ... )
```

**Arguments**

- `formula`: object of class `intReg`.
- `...`: further arguments passed to other methods (e.g. `model.frame`).

**Value**

A data.frame containing all variables used for the estimation.

The response variable is given as a factor (in case of common boundaries) or as two numeric vectors, corresponding to the lower- and upper boundary for each observation.

**Author(s)**

Arne Henningsen, Ott Toomet <otoomet@ut.ee>

**See Also**

`intReg`, and `model.frame`
model.matrix.intReg

Description
Create design matrix of interval regression models

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

Arguments
object
object of class intReg.
...
currently not used.

Value
The design matrix of interval regression models.

Author(s)
Arne Henningsen, Ott Toomet <otoomet@ut.ee>

See Also
intReg, model.matrix, and model.frame.intReg

nObs.intReg

Description
Extracts the number of observations in the intReg model

Usage
## S3 method for class 'intReg'
nObs( x, ... )

Arguments
x
object of class ‘intReg’, estimated interval regression model
...
other parameters for different methods
Value

Integer, number of observations

Author(s)

Ott Toomet <otoomet@gmail.com>

See Also

summary.intReg which provides related functionality.

Examples

```r
## example of observation-specific boundaries
## estimate the willingness to pay for the Kakadu National Park
## Data given in intervals -- 'lower' for lower bound and 'upper' for upper bound.
## Note that dichotomous-choice answers are already coded to 'lower' and 'upper'
data(Kakadu, package="Ecdat")
Kakadu <- Kakadu[sample(nrow(Kakadu), 500),]
    # subsample to speed up the estimation
## Estimate in log form, change 999 to Inf
lb <- log(Kakadu$lower)
ub <- Kakadu$upper
ub[ub > 998] <- Inf
ub <- log(ub)
y <- cbind(lb, ub)
m <- intReg(y ~ sex + log(income) + age + schooling + 
        recparks + jobs + lowrisk + wildlife + future + aboriginal + finben + 
        mineparks + moreparks + gov + 
        envcon + v parks + tenv + major, data=Kakadu)
## You may want to compare the results to Werner (1999),
## Journal of Business and Economics Statistics 17(4), pp 479-486
print(nObs(m))
```
Index

*Topic methods
  coef.intReg, 2
  model.frame.intReg, 6
  model.matrix.intReg, 7
  nObs.intReg, 7

*Topic models
  intReg, 3

*Topic regression
  intReg, 3

  coef, 4
  coef.intReg, 2

  intReg, 3, 6, 7

  lm, 3

  maxLik, 4
  model.frame, 6
  model.frame.intReg, 6, 7
  model.matrix, 7
  model.matrix.default, 4
  model.matrix.intReg, 7

  nObs.intReg, 7

  polr, 4, 5
  print.coef.intReg(coef.intReg), 2
  print.intReg(intReg), 3
  print.summary.intReg(coef.intReg), 2

  stdEr.intReg(coef.intReg), 2
  summary, 4
  summary.intReg, 2, 8
  summary.intReg(coef.intReg), 2

  vcov.intReg(coef.intReg), 2