Package 'quantreg.nonpar'

October 13, 2022

Type Package

Title Nonparametric Series Quantile Regression

Version 1.0

Date 2016-03-31

Author Michael Lipsitz, Alexandre Belloni, Victor Chernozhukov, Ivan Fernandez-Val

Maintainer Ivan Fernandez-Val <ivanf@bu.edu>

Depends R (>= 2.10), quantreg, mnormt, fda, Rearrangement

Description Implements the nonparametric quantile regression method developed by Belloni, Chernozhukov, and Fernandez-Val (2011) to partially linear quantile models. Provides point estimates of the conditional quantile function and its derivatives based on series approximations to the nonparametric part of the model. Provides pointwise and uniform confidence intervals using analytic and resampling methods.

License GPL (>= 2)

NeedsCompilation no

Repository CRAN

Date/Publication 2016-04-01 14:26:39

R topics documented:

`quantreg.nonpar-package`

Nonparametric Series Quantile Regression

Description

Implements the nonparametric quantile regression methods developed by Belloni, Chernozhukov, and Fernandez-Val (2011) to partially linear quantile models. Provides point estimates of the conditional quantile function and its derivatives based on series approximations to the nonparametric part of the model. Provides pointwise and uniform confidence intervals using analytic and resampling methods.

Details

Package:	quantreg.nonpar
Type:	Package
Version:	1.0
Date:	2014-11-05
License:	GPL(>=2)

This package is used to generate point estimates and uniform and pointwise confidence intervals in nonparametric series quantile regression models. One may use `npqr` to generate such estimates and confidence intervals and test hypotheses on the conditional quantile function and its derivatives.

Author(s)

Michael Lipsitz, Alexandre Belloni, Victor Chernozhukov, Ivan Fernandez-Val

Maintainer: Ivan Fernandez-Val <ivanf@bu.edu>

References

Belloni, A., Chernozhukov, V., and I. Fernandez-Val (2011), "Conditional quantile processes based on series or many regressors," arXiv: 1105:6154.

Koenker, R. (2011), "Additive models for quantile regression: Model selection and confidence bandaids," Brazilian Journal of Probability and Statistics 25(3), pp. 239-262.

Koenker, R. and G. Bassett (1978): "Regression Quantiles," Econometrica 46, pp. 33-50.

Ramsay, J.O., Wickham, H., Graves, S., and G. Hooker (2013), "fda: Functional Data Analysis," R package version 2.3.6, http://CRAN.R-project.org/package=fda

ddpoly *Compute Second Derivative of Orthogonal Polynomials*

Description

Returns or evaluates the second derivatives of orthogonal polynomials of degree 1 to degree over the specified set of points x: the polynomials are all orthogonal to the constant polynomial of degree 0. Alternatively, evaluates the second derivatives of raw polynomials.

Usage

```
ddpoly(x, ..., degree = 1, coefs = NULL, raw = FALSE)
```

Arguments

x	a numeric vector at which to evaluate the polynomial. x can also be a matrix. Missing values are not allowed in x.
...	further vectors.
degree	the degree of the polynomial. Must be less than the number of unique points if raw = TRUE.
coefs	for prediction, coefficients from a previous fit.
raw	if true, use raw and not orthogonal polynomials.

Value

A matrix with rows corresponding to points in x and columns corresponding to the degree, with attributes "degree" specifying the degrees of the columns (prior to taking the derivatives) and (unless raw = TRUE) "coefs" which contains the centering and normalization constants used in constructing the orthogonal polynomials. The matrix has been given class c("poly","matrix").

Note

Both the code and the description of ddpoly borrow heavily from the poly command in the stats package.

Author(s)

Michael Lipsitz, Alexandre Belloni, Victor Chernozhukov, Ivan Fernandez-Val

References

Chambers, J.M. and Hastie, T.J. (1992) Statistical Models in S. Wadsworth & Brooks/Cole. Kennedy, W.J. Jr and Gentle, J.E. (1980) Statistical Computing. Marcel Dekker.

See Also

poly, dpoly

dpoly *Compute Derivative of Orthogonal Polynomials*

Description

Returns or evaluates the first derivatives of orthogonal polynomials of degree 1 to `degree` over the specified set of points `x`: the polynomials are all orthogonal to the constant polynomial of degree 0. Alternatively, evaluates the first derivatives of raw polynomials.

Usage

```
dpoly(x, ..., degree = 1, coefs = NULL, raw = FALSE)
```

Arguments

`x`	a numeric vector at which to evaluate the polynomial. `x` can also be a matrix. Missing values are not allowed in `x`.
`...`	further vectors.
`degree`	the degree of the polynomial. Must be less than the number of unique points if `raw = TRUE`.
`coefs`	for prediction, coefficients from a previous fit.
`raw`	if true, use raw and not orthogonal polynomials.

Value

A matrix with rows corresponding to points in `x` and columns corresponding to the degree, with attributes `"degree"` specifying the degrees of the columns (prior to taking the derivative) and (unless `raw = TRUE`) `"coefs"` which contains the centering and normalization constants used in constructing the orthogonal polynomials. The matrix has been given class `c("poly","matrix")`.

Note

Both the code and the description of `dpoly` borrow heavily from the `poly` command in the `stats` package.

Author(s)

Michael Lipsitz, Alexandre Belloni, Victor Chernozhukov, Ivan Fernandez-Val

References

Chambers, J.M. and Hastie, T.J. (1992) Statistical Models in S. Wadsworth & Brooks/Cole. Kennedy, W.J. Jr and Gentle, J.E. (1980) Statistical Computing. Marcel Dekker.

See Also

`poly`, `ddpoly`

formulaDeriv *Derivative of Right Hand Side of Formula*

Description

Takes the symbolic derivative (or multiple derivatives) of the right hand side of a formula and returns a matrix with the derivative evaluated at each observation in a dataset

Usage

```
formulaDeriv(inFormula, derivVar, data, nDerivs = 1)
```

Arguments

inFormula	a formula object, with the response Y on the left of a ~ operator, and the covariate terms, separated by + operators on the right, not including the regressor whose effect is to be estimated nonparametrically. Operators such as '*', ':', 'log()', and 'I()' are allowable. However, factor variables should be constructed prior to entry in the formula: the 'factor()' operator is not allowable.
derivVar	a character object giving the name of the variable with respect to which the derivative will be taken.
data	a data.frame in which to interpret the variables named in the formula and derivVar arguments.
nDerivs	an integer: the number of derivatives to be taken.

Value

formulaDeriv returns a matrix whose dimensions are the number of observations in data and the number of variables on the right hand side of formula. Each row is the derivative of formula evaluated at the corresponding observation in data

Author(s)

Michael Lipsitz, Alexandre Belloni, Victor Chernozhukov, Ivan Fernandez-Val

See Also

npqr

gaus *Gaussian Process Inference for NPQR*

Description

A method for the generic function npqr. It computes, via a Gaussian method, the t-statistic used to conduct inference in nonparametric series quantile regression models, as well as outputting confidence intervals and hypothesis test p-values at a user-specified level.

Usage

```
gaus(data = data, B = B, taus, formula, basis = NULL, alpha=0.05,
var, load, rearrange=F, rearrange.vars="quantile", uniform=F,
se="unconditional", average = T, nderivs = 1, method = "fn")
```

Arguments

data	a data.frame in which to interpret the variables named in the formula argument.
B	the number of simulations to be performed.
taus	a numerical vector, whose entries are strictly between 0 and 1, containing the quantile indexes of interest for the quantile effects.
formula	a formula object, with the response Y on the left of a ~ operator, and the covariate terms, separated by + operators on the right, not including the regressor whose effect is to be estimated nonparametrically. Operators such as '*', ':', 'log()', and 'I()' are allowable. However, factor variables should be constructed prior to entry in the formula: the 'factor()' operator is not allowable.
basis	either a basis generated using the fda package of type "bspline" or "fourier", a factor variable, or an orthogonal polynomial basis generated using the poly command. This basis is the series regressor to be added to formula.
alpha	a real number between 0 and 1: the desired significance level (e.g., 0.05).
var	a column name within data whose values will be used, in combination with basis, to create the vectors used in the nonparametric part of the model.
load	optional manual input of loading vector (or matrix of loading vectors) that will be used as data points at which inference will be performed and over which hypothesis tests will be conducted. Each vector of load should be input as the concatenation of vectors whose entries correspond to the entries of v and $Z(w)$, respectively (for example, the average values of each variable for the parametric part of the model, v, and a specific point for the nonparametric part of the model, $Z(w)$).
rearrange	a boolean specifiying whether estimates will be monotonized prior to performing inference (requires that average=FALSE and nderivs=0).
rearrange.vars	if rearrange = TRUE, specifies whether monotonization will occur over "quantile", "var" (the variable of interest), or "both".

uniform	a boolean specifying whether inference will be uniform across observations and quantiles or done in a pointwise manner.
se	either "conditional" or "unconditional". Specifies whether standard errors, for pivotal and gaussian processes, will be conditional on the sample or not.
average	if load is not input, if average=TRUE, specifies that inference should be performed on the average value of a derivative (as specified by nderivs) of the conditional quantile function (inference cannot be performed when average=TRUE and nderivs=0). If average=FALSE, inference will be run at each unique value of the variable of interest in the dataset.
nderivs	the number of derivatives of the conditional quantile function upon which inference should be performed.
method	method to be implemented in quantile regressions: passed to function rq.

Value

gaus returns a list containing the following elements:

qfits	a list whose length is equal to the length of taus. Each element is an rq.object returned by rq for the corresponding quantile.
point.est	a matrix containing the point estimates of interest (e.g., the average derivative of the function) for each pair of loading vectors and taus. The matrix is j by i, where j is the number of loading vectors specified (i.e., the number of observations in the dataset if average=FALSE and 1 if average=TRUE) and i is the number of taus specified.
var.unique	a vector containing all values of the covariate of interest with no repeated values.
CI	an array containing the two-sided confidence interval for each pair of loading vectors and taus. The array is j by i by 2, where j is the number of loading vectors specified (i.e., the number of observations in the dataset if average=FALSE and 1 if average=TRUE) and i is the number of taus specified. The final dimension indexes the lower and upper bounds of the confidence interval, respectively.
CI.oneSided	an array containing the one-sided confidence bounds for each pair of loading vectors and taus. The array is j by i by 2, where j is the number of loading vectors specified (i.e., the number of observations in the dataset if average=FALSE and 1 if average=TRUE) and i is the number of taus specified. The final dimension indexes the lower and upper confidence bounds, respectively.
std.error	a matrix containing estimated standard errors for the quantile regression point estimates for each pair of loading vectors and taus. Depending on user selections, these may be conditional on the sample or unconditional. The array is j by i, where j is the number of loading vectors specified (i.e., the number of observations in the dataset if average=FALSE and 1 if average=TRUE) and i is the number of taus specified.
pvalues	a vector containing the p-values for hypothesis tests of three null hypotheses. First, that theta(tau,w) <= 0 for all (tau,w) pairs, where theta is the quantity of interest (e.g., the derivative of the function at each quantile and at each observation). Second, that theta(tau,w) >= 0 for all (tau,w) pairs. Third, that theta(tau,w) = 0 for all (tau,w) pairs.

load — the loading vector or matrix of loading vectors used as data points at which inference was performed and over which hypothesis tests were conducted. If load was not input by the user, load is generated based on average and nderivs.

Author(s)

Michael Lipsitz, Alexandre Belloni, Victor Chernozhukov, Ivan Fernandez-Val

References

Belloni, A., Chernozhukov, V., and I. Fernandez-Val (2011), "Conditional quantile processes based on series or many regressors," arXiv:1105.6154.

See Also

npqr

---

gbootstrap *Gradient Bootstrap Inference for NPQR*

Description

A method for the generic function npqr. It computes, via a gradient bootstrap method, the t-statistic used to conduct inference in nonparametric series quantile regression models, as well as outputting confidence intervals and hypothesis test p-values at a user-specified level.

Usage

```
gbootstrap(data = data, B = B, taus, formula, basis = NULL, alpha = 0.05,
var, load, rearrange=F, rearrange.vars="quantile", uniform=F,
average=T, nderivs=1, method = "fn")
```

Arguments

data — a data.frame in which to interpret the variables named in the formula argument.

B — the number of bootstrap repetitions to be performed.

taus — a numerical vector, whose entries are strictly between 0 and 1, containing the quantile indexes of interest.

formula — a formula object, with the response Y on the left of a ~ operator, and the covariate terms, separated by + operators on the right, not including the regressor whose effect is to be estimated nonparametrically. Operators such as '*', ':', 'log()', and 'I()' are allowable. However, factor variables should be constructed prior to entry in the formula: the 'factor()' operator is not allowable.

basis — either a basis generated using the fda package of type "bspline" or "fourier", a factor variable, or an orthogonal polynomial basis generated using the poly command. This basis is the series regressor to be added to formula.

alpha	a real number between 0 and 1: the desired significance level (e.g., 0.05).
var	a column name within data whose values will be used, in combination with basis, to create the vectors used in the nonparametric part of the model.
load	optional manual input of loading vector (or matrix of loading vectors) that will be used as data points at which inference will be performed and over which hypothesis tests will be conducted. Each vector of load should be input as the concatenation of vectors whose entries correspond to the entries of v and $Z(w)$, respectively (for example, the average values of each variable for the parametric part of the model, v, and a specific point for the nonparametric part of the model, $Z(w)$).
rearrange	a boolean specifiying whether estimates will be monotonized prior to performing inference (requires that average=FALSE and nderivs=0).
rearrange.vars	if rearrange = TRUE, specifies whether monotonization will occur over "quantile", "var" (the variable of interest), or "both".
uniform	a boolean specifying whether inference will be uniform across observations and quantiles or done in a pointwise manner.
average	if load is not input, if average=TRUE, specifies that inference should be performed on the average value of a derivative (as specified by nderivs) of the conditional quantile function (inference cannot be performed when average=TRUE and nderivs=0). If average=FALSE, inference will be run at each unique value of the variable of interest in the dataset.
nderivs	the number of derivatives of the conditional quantile function upon which inference should be performed.
method	method to be implemented in quantile regressions: passed to function rq.

Value

gbootstrap returns a list containing the following elements:

qfits	a list whose length is equal to the length of taus. Each element is an rq.object returned by rq for the corresponding quantile.
point.est	a matrix containing the point estimates of interest (e.g., the average derivative of the function) for each pair of loading vectors and taus. The matrix is j by i, where j is the number of loading vectors specified (i.e., the number of observations in the dataset if average=FALSE and 1 if average=TRUE) and i is the number of taus specified.
var.unique	a vector containing all values of the covariate of interest with no repeated values.
CI	an array containing the two-sided confidence interval for each pair of loading vectors and taus. The array is j by i by 2, where j is the number of loading vectors specified (i.e., the number of observations in the dataset if average=FALSE and 1 if average=TRUE) and i is the number of taus specified. The final dimension indexes the lower and upper bounds of the confidence interval, respectively.
CI.oneSided	an array containing the one-sided confidence bounds for each pair of loading vectors and taus. The array is j by i by 2, where j is the number of loading vectors specified (i.e., the number of observations in the dataset if average=FALSE and 1 if average=TRUE) and i is the number of taus specified. The final dimension indexes the lower and upper confidence bounds, respectively.

std.error	a matrix containing estimated standard errors for the quantile regression point estimates for each pair of loading vectors and taus. The array is j by i, where j is the number of loading vectors specified (i.e., the number of observations in the dataset if average=FALSE and 1 if average=TRUE) and i is the number of taus specified.
pvalues	a vector containing the p-values for hypothesis tests of three null hypotheses. First, that theta(tau,w) <= 0 for all (tau,w) pairs, where theta is the quantity of interest (e.g., the derivative of the function at each quantile and at each observation). Second, that theta(tau,w) >= 0 for all (tau,w) pairs. Third, that theta(tau,w) = 0 for all (tau,w) pairs.
load	the loading vector or matrix of loading vectors used as data points at which inference was performed and over which hypothesis tests were conducted. If load was not input by the user, load is generated based on average and nderivs.

Author(s)

Michael Lipsitz, Alexandre Belloni, Victor Chernozhukov, Ivan Fernandez-Val

References

Belloni, A., Chernozhukov, V., and I. Fernandez-Val (2011), "Conditional quantile processes based on series or many regressors," arXiv:1105.6154.

See Also

npqr

---

india *Childhood Malnutrition in India*

Description

Demographic and Health Survey data on childhood nutrition in India.

Usage

```
data(india)
```

Format

A data frame with 37623 observations on the following 21 variables.

cheight child's height (centimeters); a numeric vector

cage child's age (months); a numeric vector

breastfeeding duration of breastfeeding (months); a numeric vector

csex child's sex; a factor with levels male female

`ctwin` whether or not child is a twin; a factor with levels `single` `birth` `twin`

`cbirthorder` birth order of the child; a factor with levels `1` `2` `3` `4` `5`

`mbmi` mother's BMI (kilograms per meter squared); a numeric vector

`mage` mother's age (years); a numeric vector

`medu` mother's years of education; a numeric vector

`edupartner` father's years of education; a numeric vector

`munemployed` mother's employment status; a factor variable with levels `unemployed` `employed`

`mreligion` mother's religion; a factor variable with levels `christian` `hindu` `muslim` `other` `sikh`

`mresidence` mother's residential classification; a factor with levels `urban` `rural`

`wealth` mother's relative wealth; a factor with levels `poorest` `poorer` `middle` `richer` `richest`

`electricity` electricity access; a factor with levels `no` `yes`

`radio` radio ownership; a factor with levels `no` `yes`

`television` television ownership; a factor with levels `no` `yes`

`refrigerator` refrigerator ownership; a factor with levels `no` `yes`

`bicycle` bicycle ownership; a factor with levels `no` `yes`

`motorcycle` motorcycle ownership; a factor with levels `no` `yes`

`car` car ownership; a factor with levels `no` `yes`

Source

http://www.econ.uiuc.edu/~roger/research/bandaids/india.Rda

References

Koenker, R. (2011), "Additive models for quantile regression: Model selection and confidence bandaids," Brazilian Journal of Probability and Statistics 25(3), pp. 239-262.

`load.sum` *Appropriate Summary Statistics for Factors, Ordered Factors, and Numeric Variables*

Description

Returns the medians of a vector of ordered factor variables, the modes of a vector of unordered factor variables, and the means of a vector of numeric variables.

Usage

```
load.sum(vec)
```

Arguments

`vec`	A vector of ordered factor variables, a vector of unordered factor variables, or a vector of numeric variables.

Value

`load.sum` returns the medians of a vector of ordered factor variables, the mode of a vector of unordered factor variables, and the mean of a vector of numeric variables.

Author(s)

Michael Lipsitz, Alexandre Belloni, Victor Chernozhukov, Ivan Fernandez-Val

See Also

npqr

`msqrt` *Square Root of Matrix by Spectral Decomposition*

Description

Obtains the square root of a symmetric matrix by spectral decomposition.

Usage

```
msqrt(a)
```

Arguments

`a`	a matrix

Value

`msqrt` returns the square root of a symmetric matrix, obtained via spectral decomposition

Author(s)

Michael Lipsitz, Alexandre Belloni, Victor Chernozhukov, Ivan Fernandez-Val

See Also

npqr

`no.process` *Estimation for NPQR with No Inference*

Description

A method for the generic function `npqr`. It computes the quantile regression fits without performing inference

Usage

```
no.process(data = data, taus, formula, basis = NULL,
var, load, rearrange=F, rearrange.vars="quantile",
average=T, nderivs=1, method = "fn")
```

Arguments

`data`	a data.frame in which to interpret the variables named in the `formula` argument.
`taus`	a numerical vector, whose entries are strictly between 0 and 1, containing the quantile indexes of interest.
`formula`	a formula object, with the response Y on the left of a ~ operator, and the covariate terms, separated by + operators on the right, not including the regressor whose effect is to be estimated nonparametrically. Operators such as '*', ':', 'log()', and 'I()' are allowable. However, factor variables should be constructed prior to entry in the formula: the 'factor()' operator is not allowable.
`basis`	either a basis generated using the `fda` package of type "bspline" or "fourier", a factor variable, or an orthogonal polynomial basis generated using the `poly` command. This basis is the series regressor to be added to `formula`.
`var`	a column name within `data` whose values will be used, in combination with `basis`, to create the vectors used in the nonparametric part of the model.
`load`	optional manual input of loading vector (or matrix of loading vectors) that will be used as data points at which inference will be performed and over which hypothesis tests will be conducted. Each vector of `load` should be input as the concatenation of vectors whose entries correspond to the entries of v and $Z(w)$, respectively (for example, the average values of each variable for the parametric part of the model, v, and a specific point for the nonparametric part of the model, $Z(w)$).
`rearrange`	a boolean specifiying whether estimates will be monotonized (requires that `average=FALSE` and `nderivs=0`).
`rearrange.vars`	if `rearrange = TRUE`, specifies whether monotonization will occur over "quantile", "var" (the variable of interest), or "both".
`average`	if `load` is not input, if `average=TRUE`, specifies that inference should be performed on the average value of a derivative (as specified by `nderivs`) of the conditional quantile function (inference cannot be performed when `average=TRUE` and `nderivs=0`). If `average=FALSE`, inference will be run at each unique value of the variable of interest in the dataset.

nderivs	the number of derivatives of the conditional quantile function upon which point estimates should be generated.
method	method to be implemented in quantile regressions: passed to function rq.

Value

no.process returns a list containing the following elements:

qfits	a list whose length is equal to the length of taus. Each element is an rq.object returned by rq for the corresponding quantile.
point.est	a matrix containing the point estimates of interest (e.g., the average derivative of the function) for each pair of loading vectors and taus. The matrix is j by i, where j is the number of loading vectors specified (i.e., the number of observations in the dataset if average=FALSE and 1 if average=TRUE) and i is the number of taus specified.
var.unique	a vector containing all values of the covariate of interest with no repeated values.
load	the loading vector or matrix of loading vectors used as data points at which point estimates were generated. If load was not input by the user, load is generated based on average and nderivs.

Author(s)

Michael Lipsitz, Alexandre Belloni, Victor Chernozhukov, Ivan Fernandez-Val

References

Belloni, A., Chernozhukov, V., and I. Fernandez-Val (2011), "Conditional quantile processes based on series or many regressors," arXiv:1105.6154.

See Also

npqr

---

npqr *Nonparametric Series Quantile Regression*

Description

Implements the nonparametric quantile regression methods developed by Belloni, Chernozhukov, and Fernandez-Val (2011) to partially linear quantile models, $Y = g(w,u) + v'\gamma(u)$, $u|v,w\, U[0,1]$. Provides point estimates of the conditional quantile function and its derivatives based on series approximations to the nonparametric part of the model, $g(w,u)$, approximated by $Z(w)'\beta(u)$. Provides pointwise and uniform confidence intervals using analytic and resampling methods.

Usage

```
npqr(formula, data, basis = NULL, var, taus = c(0.25, 0.5, 0.75),
print.taus = NULL, B = 200, nderivs = 1, average = T,
load = NULL, alpha = 0.05, process = "pivotal", rearrange = F,
rearrange.vars="quantile", uniform = F, se = "unconditional",
printOutput = T, method = "fn")
```

Arguments

formula	a formula object, with the response Y on the left of a ~ operator, and the covariate terms, separated by + operators on the right, not including the regressor whose effect is to be estimated nonparametrically. Operators such as '*', ':', 'log()', and 'I()' are allowable. However, factor variables should be constructed prior to entry in the formula: the 'factor()' operator is not allowable.
data	a data.frame in which to interpret the variables named in the formula and var arguments. Observations in data used to construct the loading vector (either manually or automatically) will be hereafter referred to as w.
basis	a nonparametric basis object (created with the package fda), an orthogonal polynomial basis of class "poly", or a factor variable that will be used to estimate the effect of var.
var	a column name within data whose values will be used, in combination with basis, to create the vectors used in the nonparametric part of the model.
taus	a vector of quantiles of interest.
print.taus	a vector of quantiles (which must be a subset of taus), estimates for which will be printed as output.
B	the number of simulations (for the pivotal and gaussian methods) or bootstrap repetitions (for the weighted bootstrap and gradient bootstrap methods) to be performed.
nderivs	if load is not input, the number of derivatives of the conditional quantile function upon which inference should be performed.
average	if load is not input, if average=TRUE, specifies that inference should be performed on the average value of a derivative (as specified by nderivs) of the conditional quantile function (inference cannot be performed when average=TRUE and nderivs=0). If average=FALSE, inference will be run at each unique value of the variable of interest in the dataset.
load	optional manual input of loading vector (or matrix of loading vectors) that will be used as data points at which inference will be performed and over which hypothesis tests will be conducted. Each vector of load should be input as the concatenation of vectors whose entries correspond to the entries of v and $Z(w)$, respectively (for example, the average values of each variable for the parametric part of the model, v, and a specific point for the nonparametric part of the model, $Z(w)$).
alpha	a real number between 0 and 1: the desired significance level (e.g., 0.05).
process	either "pivotal", "gaussian", "wbootstrap", "gbootstrap", or "none": specifies the process used to estimate confidence intervals and p-values of hypothesis tests (or, if process = "none", specifies that inference should not be performed).

rearrange	a boolean specifiying whether estimates will be monotonized prior to performing inference (requires that average=FALSE and nderivs=0).
rearrange.vars	if rearrange = TRUE, specifies whether monotonization will occur over "quantile", "var" (the variable of interest), or "both".
uniform	a boolean specifying whether inference will be done uniformly across observations and quantiles or in a pointwise manner.
se	either "conditional" or "unconditional". Specifies whether standard errors, for pivotal and gaussian methods, will be conditional on the sample or not.
printOutput	a boolean specifying whether or not output will be printed.
method	method to be implemented in quantile regressions: passed to function rq.

Details

The loading vector may be specified in one of two ways: it may be input manually with load. If load is not specified, the loading vector will be calculated automatically using average and nderivs as parameters.

Note that derivatives calculated automatically will always be with respect to the nonparametric variable of interest, var. This means that, for example, if var=logprice, where logprice is the natural logarithm of price, then the derivative will be taken with respect to logprice, not with respect to price. Specification of var will not admit mathematical functions such as log. Specification of formula will admit some functions (e.g., log, multiplication of covariates, interaction of covariates). However, formula will not admit some formula operators; in particular, factor variables must be saved as new variables prior to entry into formula. See the vignette for more information.

Value

returns a list of results

CI	an array containing the two-sided confidence interval for each pair of loading vectors and taus. The array is j by i by 2, where j is the number of loading vectors specified (i.e., the number of observations in the dataset if average=FALSE and 1 if average=TRUE) and i is the number of taus specified. The final dimension indexes the lower and upper bounds of the confidence interval, respectively.
CI.oneSided	an array containing the one-sided confidence bounds for each pair of loading vectors and taus. The array is j by i by 2, where j is the number of loading vectors specified (i.e., the number of observations in the dataset if average=FALSE and 1 if average=TRUE) and i is the number of taus specified. The final dimension indexes the lower and upper confidence bounds, respectively.
point.est	a matrix containing the point estimates of interest (e.g., the average derivative of the conditional quantile function) for each pair of loading vectors and taus. The matrix is j by i, where j is the number of loading vectors specified (i.e., the number of observations in the dataset if average=FALSE and 1 if average=TRUE) and i is the number of taus specified.
std.error	a matrix containing estimated standard errors for the point estimates for each pair of loading vectors and taus. Depending on user selections, these may be conditional on the sample or unconditional. The array is j by i, where j is the number of loading vectors specified (i.e., the number of observations in the

dataset if `average=FALSE` and 1 if `average=TRUE`) and i is the number of `taus` specified.

`pvalues` a vector containing the p-values for hypothesis tests of three null hypotheses. First, that theta(tau,w) <= 0 for all (tau,w) pairs, where theta is the quantity of interest (e.g., the derivative of the function at each quantile and at each observation). Second, that theta(tau,w) >= 0 for all (tau,w) pairs. Third, that theta(tau,w) = 0 for all (tau,w) pairs.

`taus` This is the input vector of quantile indexes.

`coefficients` a list of length equal to the number of `taus` specified. Each element of the list contains the coefficients from the nonparametric quantile regression performed at the corresponding `taus`.

`var.unique` a vector containing all values of the covariate of interest with no repeated values.

`load` the loading vector or matrix of loading vectors used as data points at which inference was performed and over which hypothesis tests were conducted. If `load` was not input by the user, `load` is generated based on `average` and `nderivs`.

Author(s)

Michael Lipsitz, Alexandre Belloni, Victor Chernozhukov, Ivan Fernandez-Val

References

Belloni, A., Chernozhukov, V., and I. Fernandez-Val (2011), "Conditional quantile processes based on series or many regressors," arXiv: 1105:6154.

Koenker, R. (2011), "Additive models for quantile regression: Model selection and confidence bandaids," Brazilian Journal of Probability and Statistics 25(3), pp. 239-262.

Koenker, R. and G. Bassett (1978): "Regression Quantiles," Econometrica 46, pp. 33-50.

Ramsay, J.O., Wickham, H., Graves, S., and G. Hooker (2013), "fda: Functional Data Analysis," R package version 2.3.6, http://CRAN.R-project.org/package=fda

See Also

`rq`

Examples

```
data(india)

## Subset the data for speed
india.subset<-india[1:1000,]

formula=cheight~mbmi+breastfeeding+mage+medu+edupartner

basis.bsp <- create.bspline.basis(breaks=quantile(india$cage,c(0:10)/10))

n=length(india$cage)
B=500
alpha=.95
```

```
taus=c(1:24)/25
print.taus=c(1:4)/5

## Inference on average growth rate

piv.bsp <- npqr(formula=formula, data=india.subset, basis=basis.bsp,
var="cage", taus=taus, print.taus=print.taus, B=B, nderivs=1,
average=1, alpha=alpha, process="pivotal", rearrange=FALSE,
uniform=TRUE, se="unconditional", printOutput=TRUE, method="fn")

yrange<-range(piv.bsp$CI)
xrange<-c(0,1)
plot(xrange,yrange,type="n",xlab="",ylab="Average Growth (cm/month)")
lines(piv.bsp$taus,piv.bsp$point.est)
lines(piv.bsp$taus,piv.bsp$CI[1,,1],col="blue")
lines(piv.bsp$taus,piv.bsp$CI[1,,2],col="blue")
title("Average Growth Rate")

## Estimation on average growth acceleration with no inference

piv.bsp.secondderiv <- npqr(formula=formula, data=india.subset,
basis=basis.bsp, var="cage", taus=taus, print.taus=print.taus,
B=B, nderivs=2, average=0, alpha=alpha, process="none",
se="conditional", rearrange=FALSE, printOutput=FALSE, method="fn")

xsurf<-as.vector(piv.bsp.secondderiv$taus)
ysurf<-as.vector(piv.bsp.secondderiv$var.unique)
zsurf<-t(piv.bsp.secondderiv$point.est)

persp(xsurf, ysurf, zsurf, xlab="Quantile", ylab="Age (months)",
zlab="Growth Acceleration", ticktype="detailed", phi=30,
theta=120, d=5, col="green", shade=0.75, main="Growth Acceleration")
```

`pivotal` *Pivotal Process Inference for NPQR*

Description

A method for the generic function `npqr`. It computes, via a pivotal method, the t-statistic used to conduct inference in nonparametric series quantile regression models, as well as outputting confidence intervals and hypothesis test p-values at a user-specified level.

Usage

```
pivotal(data=data, B=B, taus, formula, basis = NULL, alpha=0.05,
var, load, rearrange=F, rearrange.vars="quantile", uniform=F,
se="unconditional", average=T, nderivs=1, method="fn")
```

Arguments

data	a data.frame in which to interpret the variables named in the formula argument.
B	the number of simulations to be performed.
taus	a numerical vector, whose entries are strictly between 0 and 1, containing the quantile indexes of interest.
formula	a formula object, with the response Y on the left of a ~ operator, and the covariate terms, separated by + operators on the right, not including the regressor whose effect is to be estimated nonparametrically. Operators such as '*', ':', 'log()', and 'I()' are allowable. However, factor variables should be constructed prior to entry in the formula: the 'factor()' operator is not allowable.
basis	either a basis generated using the fda package of type "bspline" or "fourier", a factor variable, or an orthogonal polynomial basis generated using the poly command. This basis is the series regressor to be added to formula.
alpha	a real number between 0 and 1: the desired significance level (e.g., 0.05).
var	a column name within data whose values will be used, in combination with basis, to create the vectors used in the nonparametric part of the model.
load	optional manual input of loading vector (or matrix of loading vectors) that will be used as data points at which inference will be performed and over which hypothesis tests will be conducted. Each vector of load should be input as the concatenation of vectors whose entries correspond to the entries of v and $Z(w)$, respectively (for example, the average values of each variable for the parametric part of the model, v, and a specific point for the nonparametric part of the model, $Z(w)$).
rearrange	a boolean specifiying whether estimates will be monotonized prior to performing inference (requires that average=FALSE and nderivs=0).
rearrange.vars	if rearrange = TRUE, specifies whether monotonization will occur over "quantile", "var" (the variable of interest), or "both".
uniform	a boolean specifying whether inference will be uniform across observations and quantiles or done in a pointwise manner.
se	either "conditional" or "unconditional". Specifies whether standard errors, for pivotal and gaussian processes, will be conditional on the sample or not.
average	if load is not input, if average=TRUE, specifies that inference should be performed on the average value of a derivative (as specified by nderivs) of the conditional quantile function (inference cannot be performed when average=TRUE and nderivs=0). If average=FALSE, inference will be run at each unique value of the variable of interest in the dataset.
nderivs	the number of derivatives of the conditional quantile function upon which inference should be performed.
method	method to be implemented in quantile regressions: passed to function rq.

Value

pivotal returns a list containing the following elements:

qfits	a list whose length is equal to the length of taus. Each element is an rq.object returned by rq for the corresponding quantile.
point.est	a matrix containing the point estimates of interest (e.g., the average derivative of the function) for each pair of loading vectors and taus. The matrix is j by i, where j is the number of loading vectors specified (i.e., the number of observations in the dataset if average=FALSE and 1 if average=TRUE) and i is the number of taus specified.
var.unique	a vector containing all values of the covariate of interest with no repeated values.
CI	an array containing the two-sided confidence interval for each pair of loading vectors and taus. The array is j by i by 2, where j is the number of loading vectors specified (i.e., the number of observations in the dataset if average=FALSE and 1 if average=TRUE) and i is the number of taus specified. The final dimension indexes the lower and upper bounds of the confidence interval, respectively.
CI.oneSided	an array containing the one-sided confidence bounds for each pair of loading vectors and taus. The array is j by i by 2, where j is the number of loading vectors specified (i.e., the number of observations in the dataset if average=FALSE and 1 if average=TRUE) and i is the number of taus specified. The final dimension indexes the lower and upper confidence bounds, respectively.
std.error	a matrix containing estimated standard errors for the quantile regression point estimates for each pair of loading vectors and taus. Depending on user selections, these may be conditional on the sample or unconditional. The array is j by i, where j is the number of loading vectors specified (i.e., the number of observations in the dataset if average=FALSE and 1 if average=TRUE) and i is the number of taus specified.
pvalues	a vector containing the p-values for hypothesis tests of three null hypotheses. First, that theta(tau,w) <= 0 for all (tau,w) pairs, where theta is the quantity of interest (e.g., the derivative of the function at each quantile and at each observation). Second, that theta(tau,w) >= 0 for all (tau,w) pairs. Third, that theta(tau,w) = 0 for all (tau,w) pairs.
load	the loading vector or matrix of loading vectors used as data points at which inference was performed and over which hypothesis tests were conducted. If load was not input by the user, load is generated based on average and nderivs.

Author(s)

Michael Lipsitz, Alexandre Belloni, Victor Chernozhukov, Ivan Fernandez-Val

References

Belloni, A., Chernozhukov, V., and I. Fernandez-Val (2011), "Conditional quantile processes based on series or many regressors," arXiv:1105.6154.

See Also

npqr

poly.wrap *Orthogonal Polynomial Wrapper*

Description

A wrapper for poly, dpoly, and ddpoly.

Usage

```
poly.wrap(x, degree = 1, coefs = NULL, nderivs = 1, raw = FALSE)
```

Arguments

x	a numeric vector at which to evaluate the polynomial. x can also be a matrix. Missing values are not allowed in x.
degree	the degree of the polynomial. Must be less than the number of unique points if raw = TRUE.
coefs	for prediction, coefficients from a previous fit.
nderivs	allowable values are 0, 1, and 2. If nderivs = 0, all other arguments are passed to poly. If nderivs = 1, all other arguments are passed to dpoly. If nderivs = 2, all other arguments are passed to ddpoly.
raw	if true, use raw and not orthogonal polynomials.

Value

poly.wrap returns the value returned by poly, dpoly, or ddpoly, depending on the value of nderivs.

Author(s)

Michael Lipsitz, Alexandre Belloni, Victor Chernozhukov, Ivan Fernandez-Val

See Also

poly, dpoly, ddpoly

removeI *Remove I() Tags From Formula*

Description

Remove I() tags from a formula. Used in the process of computing the symbolic derivative of the right hand side of a formula.

Usage

```
removeI(inString)
```

Arguments

inString	a character object

Value

removeI returns a character object identical to inString but with any I() tags removed

Author(s)

Michael Lipsitz, Alexandre Belloni, Victor Chernozhukov, Ivan Fernandez-Val

See Also

formulaDeriv

wbootstrap *Weighted Bootstrap Inference for NPQR*

Description

A method for the generic function npqr. It computes, via a weighted bootstrap method, the t-statistic used to conduct inference in nonparametric series quantile regression models, as well as outputting confidence intervals and hypothesis test p-values at a user-specified level.

Usage

```
wbootstrap(data = data, B = B, taus, formula, basis = NULL, alpha=0.05,
var, load, rearrange=F, rearrange.vars="quantile", uniform=F,
average=T, nderivs=1, method = "fn")
```

Arguments

data	a data.frame in which to interpret the variables named in the formula argument.
B	the number of bootstrap repetitions to be performed.
taus	a numerical vector, whose entries are strictly between 0 and 1, containing the quantile indexes of interest.
formula	a formula object, with the response Y on the left of a ~ operator, and the covariate terms, separated by + operators on the right, not including the regressor whose effect is to be estimated nonparametrically. Operators such as '*', ':', 'log()', and 'I()' are allowable. However, factor variables should be constructed prior to entry in the formula: the 'factor()' operator is not allowable.
basis	either a basis generated using the fda package of type "bspline" or "fourier", a factor variable, or an orthogonal polynomial basis generated using the poly command. This basis is the series regressor to be added to formula.
alpha	a real number between 0 and 1: the desired significance level (e.g., 0.05).
var	a column name within data whose values will be used, in combination with basis, to create the vectors used in the nonparametric part of the model.
load	optional manual input of loading vector (or matrix of loading vectors) that will be used as data points at which inference will be performed and over which hypothesis tests will be conducted. Each vector of load should be input as the concatenation of vectors whose entries correspond to the entries of v and $Z(w)$, respectively (for example, the average values of each variable for the parametric part of the model, v, and a specific point for the nonparametric part of the model, $Z(w)$).
rearrange	a boolean specifiying whether estimates will be monotonized prior to performing inference (requires that average=FALSE and nderivs=0).
rearrange.vars	if rearrange = TRUE, specifies whether monotonization will occur over "quantile", "var" (the variable of interest), or "both".
uniform	a boolean specifying whether inference will be uniform across observations and quantiles or done in a pointwise manner.
average	if load is not input, if average=TRUE, specifies that inference should be performed on the average value of a derivative (as specified by nderivs) of the conditional quantile function (inference cannot be performed when average=TRUE and nderivs=0). If average=FALSE, inference will be run at each unique value of the variable of interest in the dataset.
nderivs	the number of derivatives of the function itself upon which inference should be performed.
method	method to be implemented in quantile regressions: passed to function rq.

Value

wbootstrap returns a list containing the following elements:

qfits	a list whose length is equal to the length of taus. Each element is an rq.object returned by rq for the corresponding quantile.

`point.est` a matrix containing the point estimates of interest (e.g., the average derivative of the function) for each pair of loading vectors and `taus`. The matrix is j by i, where j is the number of loading vectors specified (i.e., the number of observations in the dataset if `average=FALSE` and 1 if `average=TRUE`) and i is the number of `taus` specified.

`var.unique` a vector containing all values of the covariate of interest with no repeated values.

`CI` an array containing the two-sided confidence interval for each pair of loading vectors and `taus`. The array is j by i by 2, where j is the number of loading vectors specified (i.e., the number of observations in the dataset if `average=FALSE` and 1 if `average=TRUE`) and i is the number of `taus` specified. The final dimension indexes the lower and upper bounds of the confidence interval, respectively.

`CI.oneSided` an array containing the one-sided confidence bounds for each pair of loading vectors and `taus`. The array is j by i by 2, where j is the number of loading vectors specified (i.e., the number of observations in the dataset if `average=FALSE` and 1 if `average=TRUE`) and i is the number of `taus` specified. The final dimension indexes the lower and upper confidence bounds, respectively.

`std.error` a matrix containing estimated standard errors for the point estimates for each pair of loading vectors and `taus`. Depending on user selections, these may be conditional on the sample or unconditional. The array is j by i, where j is the number of loading vectors specified (i.e., the number of observations in the dataset if `average=FALSE` and 1 if `average=TRUE`) and i is the number of `taus` specified.

`pvalues` a vector containing the p-values for hypothesis tests of three null hypotheses. First, that theta(tau,w) <= 0 for all (tau,w) pairs, where theta is the quantity of interest (e.g., the derivative of the function at each quantile and at each observation). Second, that theta(tau,w) >= 0 for all (tau,w) pairs. Third, that theta(tau,w) = 0 for all (tau,w) pairs.

`load` the loading vector or matrix of loading vectors used as data points at which inference was performed and over which hypothesis tests were conducted. If `load` was not input by the user, `load` is generated based on `average` and `nderivs`.

Author(s)

Michael Lipsitz, Alexandre Belloni, Victor Chernozhukov, Ivan Fernandez-Val

References

Belloni, A., Chernozhukov, V., and I. Fernandez-Val (2011), "Conditional quantile processes based on series or many regressors," arXiv:1105.6154.

See Also

npqr

Index