Package ‘cdfquantreg’

July 22, 2025

Type Package

Title Quantile Regression for Random Variables on the Unit Interval

Version 1.3.1-2

Date 2023-08-20

Description Employs a two-parameter family of distributions for modelling random variables on the (0, 1) interval by applying the cumulative distribution function (cdf) of one parent distribution to the quantile function of another.

BugReports https://ummlab.wordpress.com/resources/cdfquantreg-bugs-report/

Depends R (>= 3.5.0)

License GPL-3

Imports pracma (>= 2.3), Formula (>= 1.2), stats, MASS

Suggests knitr, rmarkdown

VignetteBuilder knitr

LazyData true

Encoding UTF-8

RoxygenNote 7.2.1

NeedsCompilation no

Author Yiyun Shou [aut, cre],
Michael Smithson [aut]

Maintainer Yiyun Shou <yiyun.shou@anu.edu.au>

Repository CRAN

Date/Publication 2023-09-03 06:30:02 UTC

Contents

`cdfquantreg-package` *Quantile Regression for Random Variables on the Unit Interval*

Description

Employs a two-parameter family of distributions for modelling random variables on the (0, 1) interval by applying the cumulative distribution function (cdf) of one parent distribution to the quantile function of another.

Details

Package:	cdfquantreg
Type:	Package
Date:	2022-05-19
License:	GPL-3

The cdfquantreg package includes 36 members of a two-parameter family of distributions for modelling random variables on the (0, 1) interval (see cdfqrFamily). This family has explicit pdfs, cdfs, and quantile functions. The two parameters consist of a location parameter and a dispersion parameter. The location parameter models the median and the dispersion parameter models the spread of other quantiles around the median (see Smithson and Shou, 2016, for details about the distribution family and the models). Separate submodels may be specified for the location and for the dispersion parameters, permitting different or overlapping sets of predictors in each.

The package offers maximum likelihood (see cdfquantreg)and bootstrap (see qrBoot) estimation methods. All model functions return S3 objects. In addition to the usual goodness of fit information, the package provides root-mean-squared errors in both the raw and logit scales, and the gradient. Model diagnostics include raw, Pearson, and deviance residuals (see residuals.cdfqr), and dfbetas (see influence.cdfqr).

For each distribution, the package provides evaluations of the pdf (dq), cdf (pq), and quantile (qq), as well as random samples from any of them (rq). Evaluations of skew and kurtosis (qrPwlm) also are available using probability-weighted L-moments.

Author(s)

Yiyun Shou and Michael Smithson

Maintainer: Yiyun Shou (`<yiyun.shou@anu.edu.au>`)

References

Shou, Y. and Smithson, M., (2019). cdfquantreg: An R Package for CDF-Quantile Regression. Journal of Statistical Software,88(1), pp.1–30, doi: 10.18637/jss.v088.i01

See Also

`cdfqrFamily`

`Ambdata` *Ambiguity-Conflict data*

Description

A data from a study that investigates the judgment under ambiguity and conflict

Usage

```
Ambdata
```

Format

A data frame with 166 rows and 2 variables:

ID subject ID

value Rating in each judgment scenario

scenario Index for judgment scenarios

Source

https://pubmed.ncbi.nlm.nih.gov/16594767/

---

anova.cdfqr *Model comparison test for fitted cdfqr models*

Description

Likelihood Ratio Tests for fitted cdfqr Objects.

Usage

```
## S3 method for class 'cdfqr'
anova(object, ..., test = "LRT")
```

Arguments

`object`	The fitted cdfqr model.
`...`	One or more cdfqr model objects for model comparison.
`test`	The model comparison test, currently only 'LRT' is implemented.

Examples

```
data(cdfqrExampleData)
fit_null <- cdfquantreg(crc99 ~ 1 | 1, 't2','t2', data = JurorData)
fit_mod1 <- cdfquantreg(crc99 ~ vert | confl, 't2','t2', data = JurorData)
anova(fit_null, fit_mod1)
```

AnxStrData *Stress-Anxiety data*

Description

A data from a study that investigates the relationship between stress and anxiety.

Usage

```
AnxStrData
```

Format

A data frame with 166 rows and 2 variables:

Anxiety Scores on Anxiety subscale

Stress Scores on Stress subscale

Source

https://pubmed.ncbi.nlm.nih.gov/16594767/

bugsLikelihood *Likelihood Functions for Generating OpenBUGS Model File*

Description

Likelihood functions for generating OpenBUGS model file.

Usage

```
bugsLikelihood(fd, sd)
```

Arguments

fd	A string that specifies the parent distribution.
sd	A string that specifies the sub-family distribution.

Value

A string to be written in the BUGS model file.

Examples

```
bugsLikelihood('t2','t2')
```

bugsModel *Generating OpenBUGS Model File*

Description

Generating OpenBUGS model file

Usage

```
bugsModel(formula, fd, sd, random = NULL, modelname = "bugmodel", wd = getwd())
```

Arguments

`formula` A formula object, with the DV on the left of an ~ operator, and predictors on the right. For the part on the right of ’~’, the specification of submodels can be separated by ’|’. So `y ~ X1 | X2` means the DV is `y`,`X1` is the term in the mean submodel, and `X2` is the term in the dispersion submodel.

`fd` A string that specifies the parent distribution (see cdfqrFamily).

`sd` A string that specifies the sub-family distribution.

`random` Character or vector of characters that indicates the random effect factors.

`modelname` The name of the model file; optional.

`wd` The working directory in which OpenBUGS will work (i.e., generate the model files and chain information).

Value

A model ‘.txt’ file is generated in the specified working directory. The function also returns a list of values:

init1,init2 Default initial values for MCMC two chain procedure.

vars A list of variables that are included in the estimation.

nodes_sample a list of characters that specify the nodes to be monitored.

Examples

```
## Not run:
# Need write access in the working directory before executing the code.
# No random component
bugsModel(y ~ x1 | x2, 't2','t2', random = NULL)
# Random component as subject ID
bugsModel(y ~ x1 | x2, 't2','t2', random = 'ID')

## End(Not run)
```

`cdfft` *The Family of Finite-Tailed Distributions*

Description

Density function, distribution function, quantile function, and random generation of variates for a specified cdf-quantile distribution.

Usage

```
cdfft(q, sigma, theta, fd, sd, mu = NULL, inner = TRUE, version)

pdfft(y, sigma, theta, fd, sd, mu = NULL, inner = TRUE, version)

qqft(p, sigma, theta, fd, sd, mu = NULL, inner = TRUE, version)

rqft(n, sigma, theta, fd, sd, mu = NULL, inner = TRUE, version)
```

Arguments

`q`	vector of quantiles.
`sigma`	vector of standard deviations.
`theta`	vector of skewness.
`fd`	A string that specifies the parent distribution. At the moment, only "arcsinh", "cauchit" and "t2" can be used. See details.
`sd`	A string that specifies the child distribution. At the moment, only "arcsinh", "cauchy" and "t2" can be used. See details.
`mu`	vector of means if 3-parameter case is used.
`inner`	A logic value that indicates if the inner (`inner = TRUE`) case or outer (`inner = FALSE`) will be used.
`version`	A string indicates that which version will be used. "V" is the tilt parameter function while "W" indicates the Jones Pewsey transformation.
`y`	vector of quantiles.
`p`	vector of probabilities.
`n`	Number of random samples.

Value

`pdfft` gives the density, `rqft` generates random variate, `qqft` gives the quantile function, and `cdfft` gives the cumulative density of specified distribution.

`cdfqr.control` *Control Optimization Parameters for CDF-Quantile Probability Distributions*

Description

Control Optimization Parameters for CDF-Quantile Probability Distributions.

Usage

```
cdfqr.control(method = "BFGS", maxit = 5000, trace = FALSE)
```

Arguments

`method`	Characters string specifying the method argument passed to optim.
`maxit`	Integer specifying the maxit argument (maximal number of iterations) passed to optim.
`trace`	Logical or integer controlling whether tracing information on the progress of the optimization should be produced

Value

A list with the arguments specified.

Examples

```
data(cdfqrExampleData)
fit <- cdfquantreg(crc99 ~ vert | confl, 't2', 't2',
data = JurorData,control = cdfqr.control(trace = TRUE))
```

`cdfqrFamily` *Overview of the family of distributions*

Description

The cdfquantreg family consists of the currently available distributions that can be used to fit quantile regression models via the cdfquantreg() function.

Usage

```
cdfqrFamily(shape = "all")
```

Arguments

`shape`	To show all distributions or the set of distribution for a specific type of shape. Can be `BM`, `TM`,`LL` or `FT` for Bimodal, Trimodal, Logit-logistic or Finite-tailed shapes, respectively.

Details

The cdfquantreg package includes a two-parameter family of distributions for modeling random variables on the (0, 1) interval by applying the cumulative distribution function (cdf) of one “parent” distribution to the quantile function of another.

The naming of these distributions is “parent - child” or “fd - sd”, where “fd” is the parent distribution, and “sd” is the child distribution.

The distributions have four characteristic shapes: Logit-logistic, bimodal, trimodal, and finite-tailed. Here is the list of currently available distributions.

Bimodal Shape Distributions

Distribution	R input	Alternative Input	Shape
Burr VII-ArcSinh	`fd = "burr7", sd = "arcsinh"`	`family = "burr7-arcsinh"`	Bimodal
Burr VII-Cauchy	`fd = "burr7", sd = "cauchy"`	`family = "burr7-cauchy"`	Bimodal
Burr VII-T2	`fd = "burr7", sd = "t2"`	`family = "burr7-t2"`	Bimodal
Burr VIII-ArcSinh	`fd = "burr8", sd = "arcsinh"`	`family = "burr8-arcsinh"`	Bimodal
Burr VIII-Cauchy	`fd = "burr8", sd = "cauchy"`	`family = "burr8-cauchy"`	Bimodal
Burr VIII-T2	`fd = "burr8", sd = "t2"`	`family = "burr8-t2"`	Bimodal
Logit-ArcSinh	`fd = "logit", sd = "arcsinh"`	`family = "logit-arcsinh"`	Bimodal
Logit-Cauchy	`fd = "logit", sd = "cauchy"`	`family = "logit-cauchy"`	Bimodal
Logit-T2	`fd = "logit", sd = "t2"`	`family = "logit-t2"`	Bimodal
T2-ArcSinh	`fd = "t2", sd = "arcsinh"`	`family = "t2-arcsinh"`	Bimodal
T2-Cauchy	`fd = "t2", sd = "cauchy"`	`family = "t2-cauchy"`	Bimodal

Trimodal Shape Distributions

Distribution	R input	Alternative Input	Shape
ArcSinh-Burr VII	`fd = "arcsinh", sd = "burr7"`	`family = "arcsinh-burr7"`	Trimodal
ArcSinh-Burr VIII	`fd = "arcsinh", sd = "burr8"`	`family = "arcsinh-burr8"`	Trimodal
ArcSinh-Logistic	`fd = "arcsinh", sd = "logistic"`	`family = "arcsinh-logistic"`	Trimodal
ArcSinh-T2	`fd = "arcsinh", sd = "t2"`	`family = "arcsinh-t2"`	Trimodal
Cauchit-Burr VII	`fd = "cauchit", sd = "burr7"`	`family = "cauchit-burr7"`	Trimodal
Cauchit-Burr VIII	`fd = "cauchit", sd = "burr8"`	`family = "cauchit-burr8"`	Trimodal
Cauchit-Logistic	`fd = "cauchit", sd = "logistic"`	`family = "cauchit-logistic"`	Trimodal
Cauchit-T2	`fd = "cauchit", sd = "t2"`	`family = "cauchit-t2"`	Trimodal
T2-Burr VII	`fd = "t2", sd = "burr7"`	`family = "t2-burr7"`	Trimodal
T2-Burr VIII	`fd = "t2", sd = "burr8"`	`family = "t2-burr8"`	Trimodal
T2-Logistic	`fd = "t2", sd = "logistic"`	`family = "t2-logistic"`	Trimodal

Logit-logistic Shape Distributions

Distribution	R input	Alternative Input	Shape
Burr VII-Burr VII	`fd = "burr7", sd = "burr7"`	`family = "burr7-burr7"`	Logit-logistic
Burr VII-Burr VIII	`fd = "burr7", sd = "burr8"`	`family = "burr7-burr8"`	Logit-logistic
Burr VII-Logistic	`fd = "burr7", sd = "logistic"`	`family = "burr7-logistic"`	Logit-logistic
Burr VIII-Burr VII	`fd = "burr8", sd = "burr7"`	`family = "burr8-burr7"`	Logit-logistic
Burr VIII-Burr VIII	`fd = "burr8", sd = "burr8"`	`family = "burr8-burr8"`	Logit-logistic
Burr VIII-Logistic	`fd = "burr8", sd = "logistic"`	`family = "burr8-logistic"`	Bimodal
Logit-Burr VII	`fd = "logit", sd = "burr7"`	`family = "logit-burr7"`	Logit-logistic
Logit-Burr VIII	`fd = "logit", sd = "burr8"`	`family = "logit-burr8"`	Logit-logistic
Logit-Logistic	`fd = "logit", sd = "logistic"`	`family = "logit-logistic"`	Logit-logistic

Finite-tailed Shape Distributions

Distribution	R input	Alternative Input	Shape

ArcSinh-ArcSinh	`fd = "arcsinh", sd = "arcsinh"`	`family = "arcsinh-arcsinh"`	Finite-tailed
ArcSinh-Cauchy	`fd = "arcsinh", sd = "cauchy"`	`family = "arcsinh-cauchy"`	Finite-tailed
Cauchit-ArcSinh	`fd = "cauchit", sd = "arcsinh"`	`family = "cauchit-arcsinh"`	Finite-tailed
Cauchit-Cauchy	`fd = "cauchit", sd = "cauchy"`	`family = "cauchit-cauchy"`	Finite-tailed
T2-T2	`fd = "t2", sd = "t2"`	`family = "t2-t2"`	Finite-tailed

Kumaraswamy Distribution

Distribution	**R input**	**Alternative Input**	**Shape**
Kumaraswamy	`fd = "", sd = ""`	`family = "-"`	

Value

A list of distributions that are available in the current version of package.

Examples

```
cdfqrFamily()
```

`cdfquantreg` *CDF-Quantile Probability Distributions*

Description

`cdfquantreg` is the main function to fit a cdf quantile regression with a variety of distributions.

Usage

```
cdfquantreg(
  formula,
  fd = NULL,
  sd = NULL,
  data,
  family = NULL,
  start = NULL,
  control = cdfqr.control(...),
  ...
)
```

Arguments

`formula` A formula object, with the dependent variable (DV) on the left of an ~ operator, and predictors on the right. For the part on the right of ’~’, the specification of the location and dispersion submodels can be separated by ’|’. So `y ~ X1 | X2` specifies that the DV is `y`, `X1` is the predictor in the location submodel, and `X2` is the predictor in the dispersion submodel.

`fd` A string that specifies the parent distribution.

`sd` A string that specifies the child distribution.

`data` The data in a data.frame format

`family` If ‘fd‘ and ‘sd‘ are not provided, the name of a member of the family of distributions can be provided (See `cdfqrFamily` for details of family functions)

`start` The starting values for model fitting. If not provided, default values will be used.

`control` Control optimization parameters (See `cdfqr.control`))

`...` Currently ignored.

Details

The cdfquantreg function fits a quantile regression model with a distributions from the cdf-quantile family selected by the user (Smithson and Shou, 2015). The model is specified in a two-part formula, one part containing the predictors of the location parameter, and the second part containing the predictors of the dispersion parameter. The models are fitted in two stages, the first of which uses the Nelder-Mead algorithm and the second of which takes the estimates from the first stage and applies the BFGS algorithm to refine the estimates.

Value

An object of class `cdfquantreg` will be returned. Generic functions such as summary,print (e.g., print.cdfqr) and coef can be used to extract output (see summary.cdfqr for more details about the generic functions that can be used). Class of object is a list with the following output:

coefficients A named vector of coefficients.

residuals Raw residuals, the difference between the fitted values and the data.

fitted The fitted values, including full model fitted values, fitted values for the mean component, and fitted values for the dispersion component.

rmse The model root mean squared errors

rmseLogit The root mean squared errors between the logit of the fitted values, and the logit of the response values.

vcov The variance-covariance matrix of the coefficient estimates.

AIC, BIC Akaike’s Information Criterion and Bayesian Information Criterion.

deviance The deviance for the model.

Examples

```
data(cdfqrExampleData)
fit <- cdfquantreg(crc99 ~ vert | confl, fd ='t2',sd ='t2', data = JurorData)

summary(fit)
```

cdfquantregC *Censored CDF-Quantile Probability Distributions*

Description

cdfquantregC is the a function to fit a censored cdf quantile regression with a variety of distributions .

Usage

```
cdfquantregC(
  formula,
  fd = NULL,
  sd = NULL,
  data,
  family = NULL,
  censor = "DB",
  c1 = NULL,
  c2 = NULL,
  start = NULL,
  control = cdfqr.control(...),
  ...
)
```

Arguments

formula A formula object, with the dependent variable (DV) on the left of an ~ operator, and predictors on the right. For the part on the right of ’~’, the specification of the location and dispersion submodels can be separated by ’|’. So y ~ X1 | X2 specifies that the DV is y, X1 is the predictor in the location submodel, and X2 is the predictor in the dispersion submodel.

fd A string that specifies the parent distribution.

sd A string that specifies the child distribution.

data The data in a data.frame format

family If ‘fd‘ and ‘sd‘ are not provided, the name of a member of the family of distributions can be provided (See cdfqrFamily for details of family functions)

censor A string variable to indicate how many censored point is used- only left censored `LC`, or only right-hand censored `RC`, or both sides `DB`.

c1 The left censored value, if NULL, the minimum value in the data will be used

c2 The right censored value, if NULL, the maximum value in the data will be used

start The starting values for model fitting. If not provided, default values will be used.

control Control optimization parameters (See cdfqr.control))

... Currently ignored.

Details

The cdfquantreg function fits a quantile regression model with a distributions from the cdf-quantile family selected by the user (Smithson and Shou, 2015). The model is specified in a two-part formula, one part containing the predictors of the location parameter, and the second part containing the predictors of the dispersion parameter. The models are fitted in two stages, the first of which uses the Nelder-Mead algorithm and the second of which takes the estimates from the first stage and applies the BFGS algorithm to refine the estimates.

Value

An object of class `cdfquantreg` will be returned. Generic functions such as summary,print (e.g., print.cdfqr) and coef can be used to extract output (see summary.cdfqr for more details about the generic functions that can be used). Class of object is a list with the following output:

coefficients A named vector of coefficients.

residuals Raw residuals, the difference between the fitted values and the data.

fitted The fitted values, including full model fitted values, fitted values for the mean component, and fitted values for the dispersion component.

rmse The model root mean squared errors

rmseLogit The root mean squared errors between the logit of the fitted values, and the logit of the response values.

vcov The variance-covariance matrix of the coefficient estimates.

AIC, BIC Akaike's Information Criterion and Bayesian Information Criterion.

deviance The deviance for the model.

Examples

```
data(cdfqrExampleData)
fit <- cdfquantregC(crc99 ~ vert | confl, c1 = 0.001, c2= 0.999,
fd ='t2',sd ='t2', data = JurorData)

summary(fit)
```

---

cdfquantregFT *CDF-Quantile Finite Tailed Probability Distributions*

Description

`cdfquantregFT` is a function to fit a cdf quantile regression with a variety of finite tailed distributions. It can account for data that has boundary values.

Usage

```
cdfquantregFT(
  formula,
  fd = NULL,
  sd = NULL,
  mu.fo = NULL,
  inner = FALSE,
  version = "V",
  data,
  family = NULL,
  start = NULL,
  ssn = 20,
  control = cdfqr.control(...),
  ...
)
```

Arguments

`formula` A formula object, with the dependent variable (DV) on the left of an ~ operator, and predictors on the right. For the part on the right of ’~’, the specification of the dispersion (sigma; first) and skewness (theta; second) submodels can be separated by ’|’. So `y ~ X1 | X2` specifies that the DV is `y`, `X1` is the predictor in the dispersion submodel, and `X2` is the predictor in the skewness submodel.

`fd` A string that specifies the parent distribution. At the moment, only "arcsinh", "cauchit" and "t2" can be used. See details.

`sd` A string that specifies the child distribution. At the moment, only "arcsinh", "cauchy" and "t2" can be used. See details.

`mu.fo` A formula object to indicate the predictors for the location submodel if the 3-parameter distribution is used, only input as ~ `predictors`

`inner` A logic value that indicates if the inner (`inner = TRUE`) case or outer (`inner = FALSE`) will be used. Currently inner case can only be used for 2-parameter distributions.

`version` A string indicates that which version will be used. "V" is the tilt transformation while "W" indicates the Jones Pewsey transformation.

`data` The data in a data.frame format

`family` If ‘fd‘ and ‘sd‘ are not provided, the name of a member of the family of distributions can be provided (see below) for details of family functions)

`start` The starting values for model fitting. If not provided, default values will be used.

`ssn` The number of searches on optimal starting values to be performed. If model does not converge, can increase this number.

`control` Control optimization parameters (See `cdfqr.control`))

`...` Currently ignored.

Details

The cdfquantregFT function fits a quantile regression model with a distributions from the cdf-quantile finite tailed distributions. Here is the list of currently available distributions.

Bimodal Shape Distributions

Distribution	R input	Alternative Input	Available Version
ArcSinh-ArcSinh	`fd = "arcsinh", sd = "arcsinh"`	`family = "arcsinh-arcsinh"`	`"V", "W"`
ArcSinh-Cauchy	`fd = "arcsinh", sd = "cauchy"`	`family = "arcsinh-cauchy"`	`"V", "W"`
Cauchit-ArcSinh	`fd = "cauchit", sd = "arcsinh"`	`family = "cauchit-arcsinh"`	`"V", "W"`
Cauchit-Cauchy	`fd = "cauchit", sd = "cauchy"`	`family = "cauchit-cauchy"`	`"V", "W"`
T2-T2	`fd = "t2", sd = "t2"`	`family = "t2-cauchy"`	`"V", "W"`

Value

An object of class `cdfqrFT` will be returned. Generic functions such as summary,print and coef can be used to extract output (see summary.cdfqr for more details about the generic functions that can be used). Class of object is a list with the following output:

coefficients A named vector of coefficients.

residuals Raw residuals, the difference between the fitted values and the data.

fitted The fitted values, including full model fitted values, fitted values for the mean component, and fitted values for the dispersion component.

rmse The model root mean squared errors

rmseLogit The root mean squared errors between the logit of the fitted values, and the logit of the response values.

vcov The variance-covariance matrix of the coefficient estimates.

AIC, BIC Akaike's Information Criterion and Bayesian Information Criterion.

deviance The deviance for the model.

Examples

```
data(cdfqrExampleData)
fit <- cdfquantregFT(pnurse ~ Ambulance |Ambulance ,
 fd = "arcsinh", sd = "arcsinh",  inner = FALSE, version = "V", data = yoon)
summary(fit)
```

`cdfquantregH` *Zero/One inflated CDF-Quantile Probability Distributions*

Description

`cdfquantregH` is the a function to fit a Zero/One inflated CDF-Quantile regression with a variety of distributions .

Usage

```
cdfquantregH(
  formula,
  zero.fo = ~1,
  one.fo = ~1,
  fd = NULL,
  sd = NULL,
  data,
  family = NULL,
  type = "ZI",
  start = NULL,
  control = cdfqr.control(...),
  ...
)
```

Arguments

`formula` A formula object, with the dependent variable (DV) on the left of an ~ operator, and predictors on the right. For the part on the right of '~', the specification of the location and dispersion submodels can be separated by '|'. So `y ~ X1 | X2` specifies that the DV is `y`, `X1` is the predictor in the location submodel, and `X2` is the predictor in the dispersion submodel.

`zero.fo` A formula object to indicate the predictors for the zero component, only input as ~ `predictors`

`one.fo` A formula object to indicate the predictors for the one component, only input as ~ `predictors`

`fd` A string that specifies the parent distribution.

`sd` A string that specifies the child distribution.

`data` The data in a data.frame format

`family` If ‘fd‘ and ‘sd‘ are not provided, the name of a member of the family of distributions can be provided (See `cdfqrFamily` for details of family functions)

`type` A string variable to indicate whether the model is zero-inflated `` `ZI` ``, or one-inflated `` `OI` ``, or zero-one inflated `` `ZO` ``.

`start` The starting values for model fitting. If not provided, default values will be used.

`control` Control optimization parameters (See `cdfqr.control`))

`...` Currently ignored.

Details

The cdfquantreg function fits a quantile regression model with a distributions from the cdf-quantile family selected by the user (Smithson and Shou, 2015). The model is specified in a two-part formula, one part containing the predictors of the location parameter, and the second part containing the predictors of the dispersion parameter. The models are fitted in two stages, the first of which uses the Nelder-Mead algorithm and the second of which takes the estimates from the first stage and applies the BFGS algorithm to refine the estimates.

Value

An object of class `cdfqrH` will be returned. Generic functions such as summary,print (e.g., print.cdfqr) and coef can be used to extract output (see summary.cdfqr for more details about the generic functions that can be used). Class of object is a list with the following output:

coefficients A named vector of coefficients.

residuals Raw residuals, the difference between the fitted values and the data.

fitted The fitted values, including full model fitted values, fitted values for the mean component, and fitted values for the dispersion component.

vcov The variance-covariance matrix of the coefficient estimates.

AIC, BIC Akaike's Information Criterion and Bayesian Information Criterion.

Examples

```
data(cdfqrExampleData)
# For one-inflated model
ipcc_high <- subset(IPCC, mid == 1 & high == 1 & prob!=0)
fit <- cdfquantregH(prob ~ valence | valence,one.fo = ~valence,
  fd ='t2',sd ='t2', type = "OI", data = ipcc_high)

summary(fit)

# For zero-inflated model
ipcc_low <- subset(IPCC, mid == 0 & high == 0 & prob!=1)
fit <- cdfquantregH(prob ~ valence | valence, zero.fo = ~valence,
  fd ='t2',sd ='t2', type = "ZI", data = ipcc_low)


# For zero &one-inflated model
ipcc_mid <- subset(IPCC, mid == 1 & high == 0)
fit <- cdfquantregH(prob ~ valence | valence, zero.fo = ~valence,
  one.fo = ~valence,
  fd ='t2',sd ='t2', type = "ZO", data = ipcc_mid)
```

---

dq *The Family of Distributions*

Description

Density function, distribution function, quantile function, and random generation of variates for a specified cdf-quantile distribution.

Usage

```
dq(x, mu, sigma, fd, sd)

rq(n, mu, sigma, fd, sd)

qq(p, mu, sigma, fd, sd)

pq(q, mu, sigma, fd, sd)
```

Arguments

x	vector of quantiles.
mu	vector of means.
sigma	vector of standard deviations.
fd	A string that specifies the parent distribution.
sd	A string that specifies the sub-family distribution.
n	Number of random samples.
p	vector of probabilities.
q	vector of quantiles.

Value

dq gives the density, rq generates random variates, qq gives the quantile function, and pq gives the cumulative density of specified distribution.

Examples

```
x <- rq(5, mu = 0.5, sigma = 1, 't2','t2'); x

dq(x, mu = 0.5, sigma = 1, 't2','t2')

qtil <- pq(x, mu = 0.5, sigma = 1, 't2','t2');qtil

qq(qtil , mu = 0.5, sigma = 1, 't2','t2')
```

ExtEvent *Extinction Study data-set*

Description

Probability of Human Extinction Study

Usage

```
ExtEvent
```

Format

A data frame with 1170 rows and 11 variables:

ID Subject ID

gend Gender of subjects, '0'is male, '1'is female

nation The nation of the participants come from

UK effect coding for nation

IND effect coding for nation

political political orientation of subjects

format The format of probability elicitation

order the order of probability judgement task.

SECS_6 Social conservativsm question on attitude toward gun ownership.

EQ1_P Probability estimates for general threats.

EQ3_P Probability estimates for the greatest threat.

Source

https://www.michaelsmithson.online/

`influence.cdfqr` *Influence Diagnosis For Fitted Cdfqr Object*

Description

Influence Diagnosis (dfbetas) For Fitted Cdfqr Object

Usage

```
## S3 method for class 'cdfqr'
influence(
  model,
  method = "dfbeta",
  type = c("full", "location", "dispersion", "skew", "zero", "one"),
  what = "full",
  plot = FALSE,
  id = FALSE,
  ...
)

## S3 method for class 'cdfqr'
dfbeta(
  model,
  type = c("full", "location", "dispersion", "skew", "zero", "one"),
  what = "full",
  ...
)

## S3 method for class 'cdfqr'
dfbetas(
  model,
  type = c("full", "location", "dispersion", "skew", "zero", "one"),
  what = "full",
  ...
)
```

Arguments

model	A cdfqr model object
method	Currently only 'dfbeta' method is available.
type	A string that indicates whether the results for all parameters are to be returned, or only the submodel's parameters returned.
what	for influence statistics based on coefficient values, indicate the predictor variables that needs to be tested.
plot	if plot is needed.
id	for plot only, if TRUE, the case ids will be displayed in the plot.
...	Pass onto other functions or currently ignored

Value

A matrix, each row of which contains the estimated influence on parameters when that row's observation is removed from the sample.

See Also

lm.influence, influence.measures

Examples

```
data(cdfqrExampleData)
fit <- cdfquantreg(crc99 ~ vert | confl, 't2', 't2', data = JurorData)
#It takes some time especially the data is large.
influcne <- influence(fit)
plot(influcne[,2])

## Not run:
# Same as influence(fit)
dfbetval <- dfbetas(fit)

## End(Not run)
```

IPCC *IPCC data-set*

Description

The IPCC data-set comprises the lower, best, and upper estimates for the phrases "likely" and "unlikely" in six IPCC report sentences.

Usage

```
IPCC
```

Format

A data frame with 4014 rows and 8 variables:

subj Subject ID number

treat Experimental conditions

valence Valence of the sentences

prob raw probability estimates

probm Linear transformed prob into (0, 1) interval

mid Distinguish lower, best and upper estiamtes

high Distinguish lower, best and upper estiamtes

Question IPCC question number

Source

https://pubmed.ncbi.nlm.nih.gov/19207697/

IPCCAUS *IPCC data-set - Australian data*

Description

The IPCC-AUS data-set comprises the best estimates for the phrases in IPCC report sentences.

Usage

```
IPCCAUS
```

Format

A data frame with 4014 rows and 8 variables:

ID Subject ID

gender Gender of subjects, ‘0‘is male, ‘1‘is female

age age of subjects

cfprob personal probability.

bestprob nominated probability.

Source

https://pubmed.ncbi.nlm.nih.gov/19207697/

IPCC_Wide *IPCC data-set - Wide format*

Description

The IPCC-wide data-set comprises the best estimates for the phrases "likely" and "unlikely" in six IPCC report sentences.

Usage

```
IPCC_Wide
```

Format

A data frame with 4014 rows and 8 variables:

Q4 Each column indicates the estimates for one sentence.

Q5 Each column indicates the estimates for one sentence.

Q6 Each column indicates the estimates for one sentence.

Q8 Each column indicates the estimates for one sentence.

Q9 Each column indicates the estimates for one sentence.

Q10 Each column indicates the estimates for one sentence.

Source

https://pubmed.ncbi.nlm.nih.gov/19207697/

JurorData *Juror data*

Description

Juror Judgment Study.

Usage

```
JurorData
```

Format

A data frame with 104 rows and 3 variables:

crc99 The ratings of confidence levels with rescaling into the (0, 1) interval to avoide 1 and 0 values.

vert was the dummy variable for coding the conditions of verdict types, whereas

confl was the dummy variable for coding the conflict conditions

Source

doi:10.1375/pplt.2004.11.1.154

plot.cdfqr *Plot Fitted Values/Residuals of A Cdfqr Object or Distribution*

Description

Plot Fitted Values/Residuals of A cdfqr Object or Distribution

Usage

```
## S3 method for class 'cdfqr'
plot(
  x,
  mu = NULL,
  sigma = NULL,
  theta = NULL,
  fd = NULL,
  sd = NULL,
  n = 10000,
  inner = TRUE,
  version = "V",
  type = c("fitted"),
  ...
)
```

Arguments

`x` If the plot is based on the fitted values, provide a fitted cdfqr object, alternatively, mu and sigma, and the distribution can be specified.

`mu` Location parameter value

`sigma` Sigma parameter value

`theta` Skew parameter value

`fd` A string that specifies the parent distribution.

`sd` A string that specifies the sub-family distribution.

`n` The number of random variates to be generated for user specified plot.

`inner` If finite-tailed distribution is used: a logic value that indicates if the inner (`inner = TRUE`) case or outer (`inner = FALSE`) will be used. Currently inner case can only be used for 2-parameter distributions.

`version` If finite-tailed distribution is used: A string indicates that which version will be used. "V" is the tilt parameter function while "W" indicates the Jones Pewsey transformation.

`type` Currently only fitted values are available for generating plots.

`...` other plot parameters pass onto `plot`.

Examples

```
data(cdfqrExampleData)
fit <- cdfquantreg(crc99 ~ vert | confl, 't2','t2', data = JurorData)
plot(fit)
```

---

predict.cdfqr *Methods for Cdfqr Objects*

Description

Methods for obtaining the fitted/predicted values for a fitted cdfqr object.

Usage

```
## S3 method for class 'cdfqr'
predict(
  object,
  newdata = NULL,
  type = c("full", "mu", "sigma", "theta", "one", "zero"),
  quant = 0.5,
  ...
)
```

```
## S3 method for class 'cdfqr'
fitted(
  object,
  type = c("full", "mu", "sigma", "theta", "one", "zero"),
  plot = FALSE,
  ...
)
```

Arguments

`object`	A cdfqr model fit object
`newdata`	Optional. A data frame in which to look for variables with which to predict. If not provided, the fitted values are returned
`type`	A character that indicates whether the full model prediction/fitted values are needed, or values for the ‘mu‘ and ‘sigma‘ submodel only.
`quant`	A number or a numeric vector (must be in (0, 1)) to specify the quantile(s) of the predicted value (when ‘newdata‘ is provided, and predicted values for responses are required). The default is to use median to predict response values.
`...`	currently ignored
`plot`	if a plot is needed.

Examples

```
data(cdfqrExampleData)
fit <- cdfquantreg(crc99 ~ vert | confl, 't2','t2', data = JurorData)

plot(predict(fit))
plot(predict(fit))
```

qrBoot *Bootstrapping for cdf quantile regression*

Description

qrBoot provides a simple bootstrapping method for estimating the parameters of a cdf quantile regression model.

Usage

```
qrBoot(object, rn, f = coef, R = 500, ci = 0.95)
```

Arguments

`object`	The fitted cdfqr model object
`rn`	The sample size of bootstrap samples
`f`	A function whose one argument is the name of a cdfqr object that will be applied to the updated cdfqr object to compute the statistics of interest. The default is coef.
`R`	Number of bootstrap samples.
`ci`	The confidence interval level to obtain the bootstrap confidence intervals

Value

A matrix that includes the original statistics, bootstrap means, and bootstrap confidence intervals

Examples

```
data(cdfqrExampleData)
fit <- cdfquantreg(crc99 ~ vert | confl, 't2', 't2', data = JurorData)
qrBoot(fit, rn = 50, R = 50)
```

qrGrad *Give the Gradient Function for CDF-Quantile Distribution Models*

Description

Give the Gradient Function for CDF-Quantile Distribution models

Usage

```
qrGrad(fd, sd)
```

Arguments

`fd`	A string that specifies the parent distribution.
`sd`	A string that specifies the sub-family distribution.

Value

grad The gradient function of parameter estimates, given a specified cdf-quantile distribution

Examples

```
qrGrad('t2','t2')
```

qrLogLik *Log Likelihood for Fitting Cdfquantile Distributions*

Description

Function to give the (negative) log likelihood for fitting cdfquantile distributions.

Usage

```
qrLogLik(y, mu, sigma, fd, sd, total = TRUE)
```

Arguments

`y`	the vector to be evaluated.
`mu`	mean of the distribution.
`sigma`	sigma of the distribution.
`fd`	A string that specifies the parent distribution.
`sd`	A string that specifies the sub-family distribution.
`total`	whether the sum of logliklihood is calculated

Value

The negative log likelihood for fitting the data with a cdfquantile distribution.

Examples

```
y <- rbeta(20, 0.5, 0.5)
qrLogLik(y, mu = 0.5, sigma = 1, 't2','t2')
```

qrPwlm *Probability Weighted L-moment Skewness and Kurtosis*

Description

Calculate the skew and kurtosis statistics based on probability weighted moments, via simulation method.

Usage

```
qrPwlm(x, n = NULL, mu = NULL, sigma = NULL, fd = NULL, sd = NULL)
```

Arguments

`x`	The vector of values for the calculation of Skewness and Kurtosis.
`n`	The number of samples drawn in the simulation. The higher this value, the greater accuracy.
`mu`	vector of means.
`sigma`	vector of standard deviations.
`fd`	A string that specifies the parent distribution.
`sd`	A string that specifies the sub-family distribution.

Details

This function computes the L-moment measures of skew and kurtosis, which may be computed via linear combinations of probability-weighted moments (Greenwood, Landwehr, Matalas and Wallis, 1979).

Value

The tau3(skew) and tau4(kurtosis) values of the L-moment.

References

Greenwood, J. A., Landwehr, J. M., Matalas, N. C., & Wallis, J. R. (1979). Probability weighted moments: definition and relation to parameters of several distributions expressable in inverse form. Water Resources Research, 15(5), 1049-1054.

Examples

```
qrPwlm(n = 1000, mu = 0.5, sigma = 1, fd = 't2', sd = 't2')
```

`qrStart` *Starting Value Generation for CDF quantile Regressions*

Description

`qrStart` is the function for generating starting values for a cdf-quantile GLM null model.

Usage

```
qrStart(ydata, fd = NULL, sd = NULL, skew = FALSE)
```

Arguments

`ydata`	The variable to be modeled
`fd`	A string that specifies the parent distribution.
`sd`	A string that specifies the sub-family distribution.
`skew`	If ture, the starting values will be generated for the finited tailed distribution case.

Details

The start values for the location parameter in a null model are the median of the empirical distribution, and a starting value for the dispersion parameter based on a specific quantile of the empirical distribution, specified according to the theoretical distribution on which the model is based. The start values for all new predictor coefficients in both the location and dispersion submodels are assigned the value 0.1.

Value

A vector that consists initial values for mu and sigma.

Examples

```
x <- rbeta(100, 1, 2)
qrStart(x, fd='t2', sd='t2')
#[1] -0.5938286  1.3996999
```

---

residuals.cdfqr *Register method for cdfqr object functions*

Description

Register method for cdfqr object functions.

Usage

```
## S3 method for class 'cdfqr'
residuals(object, type = c("raw", "pearson", "deviance"), ...)
```

Arguments

`object`	The cdfqr model project
`type`	The type of residuals to be extracted: `'raw'`, `'pearson'`,`'std.pearson'`, or `'deviance'`,
`...`	currently ignored

Value

residuals of a specified type.

Examples

```
data(cdfqrExampleData)
fit <- cdfquantreg(crc99 ~ vert | confl, 't2','t2', data = JurorData)

residuals(fit, "pearson")
```

scaleTR *Transform Values into (0, 1) Interval*

Description

scaleTR is function that rescales values of a variable into the (0, 1) interval.

Usage

```
scaleTR(y, high = NULL, low = NULL, data = NULL, N = NULL, scale = 0.5)
```

Arguments

y	A numeric vector, or a variable in a dataframe.
high	The highest possible value of that variable. The value should be equal or greater than the maximum value of y. If not supplied, the maximum value of y will be used.
low	The lowest possible value of that variable. The value should be equal or smaller than the minimum value of y. If not supplied, the minimum value of y will be used.
data	A dataframe that contains the variable y.
N	A integer, normally is the sample size or the number of values. If not supplied, the length of y will be used.
scale	A compressing parameter that determines the extend to which the boundary values are going to be pushed away from the boundary. See details.

Details

scaleTR used the method suggested by Smithson and Verkuilen (2006) and applies linear transformation to values into the open interval (0, 1). It first transform the values from their original scale by taking $y' = (y - a)/(b - a)$, where a is the lowest possible value of that variable and b is the highest possible value of that variable. Next, it compresses the range to avoid zeros and ones by taking $y" = (y'(N - 1) + c)/N$, where N is the sample size and c is the compressing parameter. The smaller value c is, the boundary values would be more approaching zeros and ones, and have greater impact on the estimation of the dispersion parameters in the cdf quantile model.

See Also

cdfquantreg

Examples

```
y <- rnorm(20, 0, 1)
ynew <- scaleTR(y)
```

summary.cdfqr *S3 Methods for getting output from fitted cdfqr Objects.*

Description

Give the S3 Methods for CDF-Quantile Distribution Models

Usage

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

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

## S3 method for class 'cdfqr'
coef(object, type = "full", ...)

## S3 method for class 'cdfqr'
vcov(object, type = "full", ...)

## S3 method for class 'cdfqr'
update(object, formula., zero.fo., one.fo., mu.fo., ..., evaluate = TRUE)

## S3 method for class 'cdfqr'
confint(object, parm, level = 0.95, submodel = "full", ...)

## S3 method for class 'cdfqr'
formula(x, ...)

## S3 method for class 'cdfqr'
nobs(object, ...)

## S3 method for class 'cdfqr'
deviance(object, ...)

## S3 method for class 'cdfqrH'
logLik(object, ...)

## S3 method for class 'cdfqrH'
confint(
  object,
  parm,
  level = 0.95,
  type = c("full", "mean", "sigma", "zero", "one"),
  ...
)
```

```
## S3 method for class 'cdfqrFT'
confint(object, parm, level = 0.95, submodel = ”full”, ...)
```

Arguments

`...` Pass onto other functions or currently ignored

`x, object` The fitted cdfqr model.

`digits` Number of digits to be retained in printed output.

`type, submodel` The parts of coefficients or variance-covariance matrix to be extracted.Can be "full", "mean",or "sigma".

`formula.` Changes to the formula. See `update.Formula` for details.

`zero.fo., one.fo., mu.fo.` Changes to the formulas for zero/one component for hurdle models, and for location submodel for finite-tailed models.

`evaluate` If true evaluate the new updated model else return the call for the new model.

`parm` a specification of which parameters are to be given confidence intervals, either a vector of numbers or a vector of names. If missing, all parameters are considered.

`level` the confidence level required.

Examples

```
data(cdfqrExampleData)
fit <- cdfquantreg(crc99 ~ vert | confl, 't2','t2', data = JurorData)

summary(fit)
print(fit)
logLik(fit)
coef(fit)
deviance(fit)
vcov(fit)
confint(fit)

#Update the model
fit2 <- update(fit, crc99 ~ vert*confl | confl)
summary(fit2)
```

yoon *Patient Time Data*

Description

Data from Modeling Proportion of Patient Time in Emergency Ward Stages

Usage

```
yoon
```

Format

A data frame with 1170 rows and 11 variables:

id case identification

Day day of the week (0 = Sunday)

Ambulance 0 = walk-in; 1 = ambulance-arrival

Triage triage level

Triage1 1 = triage level 1

Triage2 1 = triage level 2

Triage3 1 = triage level 3

Triage4 1 = triage level 4

Triage5 1 = triage level 5

Lab 1 = laboratory test(s) conducted

Xray 1 = x-ray conducted

Other 1 = other intervention

LOS length of stay in minutes

LOSh length of stay in hours

preg proportion of time in registration stage

ptriage proportion of time in triage stage

pnurse proportion of time in nursing care stage

pphysician proportion of time in consultation with physician(s)

pdecis proportion of time in decisional stage

pregptriage preg + ptriage

pphysdecis pphysician + pdecis

prnurse pnurse/(pnurse + pregptriage)

prphysdec pphysdecis /(pphysdecis + pregptriage)

Source

doi:10.1017/S1481803500006539

Index