

# Package: quantCurves (via r-universe)

September 9, 2024

**Title** Estimate Quantiles Curves

**Description** Non-parametric methods as local normal regression, polynomial local regression and penalized cubic B-splines regression are used to estimate quantiles curves. See Fan and Gijbels (1996) <[doi:10.1201/9780203748725](https://doi.org/10.1201/9780203748725)> and Perperoglou et al.(2019) <[doi:10.1186/s12874-019-0666-3](https://doi.org/10.1186/s12874-019-0666-3)>.

**Version** 1.0.0

**License** CeCILL

**Encoding** UTF-8

**RoxygenNote** 7.1.2

**Imports** graphics, KernSmooth, locpol, np, quantreg, quantregGrowth, stats

**NeedsCompilation** no

**Author** Sandie Ferrigno [cre, aut], Dounia Essaket [ctb], Arthur Mouchot [ctb], Hugo Breton [ctb], Myriam Maumy [ctb]

**Maintainer** Sandie Ferrigno <[sandie.ferrigno@univ-lorraine.fr](mailto:sandie.ferrigno@univ-lorraine.fr)>

**Date/Publication** 2022-03-11 10:40:02 UTC

**Repository** <https://nparamestimcurves.r-universe.dev>

**RemoteUrl** <https://github.com/cran/quantCurves>

**RemoteRef** HEAD

**RemoteSha** 498963ea9c0746b3c4e599fbc6a3254d1bef3604

## Contents

bandwidth . . . . .	2
bsplines . . . . .	3
bSplinesData . . . . .	4
CentCurv . . . . .	5
compareCurv . . . . .	6
fourCurv . . . . .	7
locNormCst . . . . .	8

locNormCstData . . . . .	9
locNormLin . . . . .	10
locNormLinData . . . . .	11
polylocLin . . . . .	12
polylocLinData . . . . .	13
<b>Index</b>	<b>15</b>

---

bandwidth	<i>bandwidth selection function</i>
-----------	-------------------------------------

---

## Description

bandwidth selection function

## Usage

```
bandwidth(x, y, method = "CV")
```

## Arguments

x	the explanatory variable - numeric
y	the response variable - numeric
method	the bandwidth method choice: CV or plug-in. Default is CV.

## Value

Calculates the bandwidth value using cross validation or plug-in method (for localLin and localCst methods)

## Examples

```
#create a data frame
example<-data.frame(sample(30:42,10,rep=TRUE),sample(800:5000,10,rep=TRUE))
colnames(example)<-c("Gestational Age in weeks","Weight in grams")
x<-example$`Gestational Age in weeks`
y<-example$`Weight in grams`
#calculate the window value
bandwidth(x,y)
```

bsplines

*Cubic Penalized B-splines quantile regression***Description**

Cubic Penalized B-splines quantile regression

**Usage**

```
bsplines(
  x,
  y,
  lambdas,
  d = 3,
  cents = c(0.03, 0.25, 0.5, 0.75, 0.97),
  leg = TRUE,
  axes.lab = NULL
)
```

**Arguments**

x	the explanatory variable - numeric
y	the response variable - numeric
lambdas	tunes the tradeoff between the goodness of fit and the regularity of the spline - numeric value or numeric vector
d	differentiation order - 1, 2 or 3. Default is set to d=3.
cents	numeric vector that represents the centiles calculated. Default is set to cents=c(0.03,0.25,0.5,0.75,0.97)).
leg	Boolean. Should the legend be displayed (TRUE) or not (FALSE).
axes.lab	NULL or c("Nom_axe_X, Nom_axe_Y").

**Value**

Plots the curves at centiles selected and returns an object of class gcrq.

**Examples**

```
#create a sample data frame
weights=c(500,600,1000,1150,1200,1260,1240,1300,1370,1500,2000,2100,2150,2500,
2800,2900,3050,3200,2980,3000,3300,3100,3200,3600,3500,3700,3900,3900,4000,
4200,3000,4500,4300,4900,4350,3700,4000)
ages<-c(30,30,30,31,31,31,32,32,32,33,33,33,34,34,34,35,35,35,36,36,36,
37,37,37,38,38,38,39,39,39,40,40,40,41,41,41,42)
bsplines(ages,weights,lambdas=50)
```

bSplinesData

*Centile curves using B-splines compared to noise data***Description**

Centile curves using B-splines compared to noise data

**Usage**

```
bSplinesData(
  x,
  y,
  lambdas,
  data,
  cents = c(0.03, 0.25, 0.5, 0.75, 0.97),
  leg = FALSE
)
```

**Arguments**

x	the explanatory variable - numeric
y	the response variable - numeric
lambdas	to be set by user. Can be a vector or a single numeric value. Tunes the tradeoff between the goodness of fit and the regularity of the spline - numeric value or numeric vector
data	the noise data to be compared to
cents	A numeric vector that represents the centiles calculated. Default is set to cents=c(0.03,0.25,0.5,0.75,0.97)
leg	Boolean. Should the legend be displayed (TRUE) or not (FALSE).

**Value**

Plots centile curves with B-splines of different differential orders (d) and displays them on the same figure as the noise data

**Examples**

```
#create a sample data frame
sample<-data.frame(sample(30:42,30,rep=TRUE),sample(800:5000,30,rep=TRUE))
colnames(sample)<-c("Gestational Age in weeks","Weight in grams")
x<-sample$`Gestational Age in weeks`
y<-sample$`Weight in grams`
abnormal<-data.frame(sample(30:42,6,rep=TRUE),sample(800:5000,6,rep=TRUE))
colnames(abnormal)<-c("Gestational Age in weeks","Weight in grams")
bSplinesData(x,y,lambdas=1,abnormal)
```

CentCurv

*Centile curves based on one of the different methods***Description**

Centile curves based on one of the different methods

**Usage**

```
CentCurv(
  x,
  y,
  bandwidth.select = "CV",
  method,
  lambdas = 0,
  kernel = locpol::gaussK,
  d = 3,
  cents = c(0.03, 0.25, 0.5, 0.75, 0.97),
  disp_window = FALSE
)
```

**Arguments**

x	the explanatory variable - numeric
y	the response variable - numeric
bandwidth.select	the bandwidth method choice: CV or plug-in. Default is CV.
method	str - The method choosen for displaying the curve. Could be: "Local normal constant", "Local normal linear", "Polynomial local" or "B-splines".
lambdas	set to 0. To be set if method chosen is "B-Splines". Tunes the tradeoff between the goodness of fit and the regularity of the spline - numeric value or numeric vector
kernel	the Kernel function that will be used in the algorithm ("trig", "gauss", "circ", "cubic" or "epan").
d	differentiation order - 1, 2 or 3. Default is set to d=3.
cents	A numeric vector that represents the centiles calculated. Default is set to cents=c(0.03,0.25,0.5,0.75,0.97) )
disp_window	Boolean. Should the scale of bandwidth be displayed (TRUE) or not (FALSE).

**Value**

Plots centile curves according to the chosen method

**Examples**

```
#create an example data frame
weights=c(500,600,1000,1150,1200,1260,1240,1300,1370,1500,2000,2100,2150,2500,
2800,2900,3050,3200,2980,3000,3300,3100,3200,3600,3500,3700,3900,3900,4000,
4200,3000,4500,4300,4900,4350,3700,4000,5000,4300)
age<-c(30,30,30,31,31,31,32,32,32,33,33,33,34,34,34,35,35,35,36,36,36,
37,37,37,38,38,38,39,39,39,40,40,40,41,41,41,42,42,42)
sample<-data.frame(age,weights)
colnames(sample)<-c("Gestational Age in weeks","Weight in grams")
x<-sample$`Gestational Age in weeks`
y<-sample$`Weight in grams`
CentCurv(x,y,method='Polynomial local')
CentCurv(x,y,method='B-Splines',lambdas=1)
```

---

compareCurv

*Centile curves according to different methods*


---

**Description**

Centile curves according to different methods

**Usage**

```
compareCurv(x, y, bandwidth.method = "CV", lambdas, data, leg = FALSE)
```

**Arguments**

x	the explanatory variable - numeric
y	the response variable - numeric
bandwidth.method	the method chosen to calculate bandwidth. Could be cross validation or Plug-in. Default is set to CV.
lambdas	to be set for "B-Splines". Can be a vector or a single numeric value. Tunes the tradeoff between the goodness of fit and the regularity of the spline - numeric value or numeric vector
data	the abnormal/external data we want to compare the curves with
leg	Boolean. Should the legend be displayed (TRUE) or not (FALSE).

**Value**

Plots centile curves with the different methods and displays them on the same figure as the noise data to compare

**Examples**

```
#create a sample data frame
sample<-data.frame(sample(30:42,30,rep=TRUE),sample(800:5000,30,rep=TRUE))
colnames(sample)<-c("Gestational Age in weeks","Weight in grams")
x<-sample$`Gestational Age in weeks`
y<-sample$`Weight in grams`
abnormal<-data.frame(sample(30:42,6,rep=TRUE),sample(800:5000,6,rep=TRUE))
colnames(abnormal)<-c("Gestational Age in weeks","Weight in grams")
compareCurv(x,y,bandwidth.method="Plug-in",lambdas=1,abnormal)
```

fourCurv

*Centile curves based on each of the four different methods***Description**

Centile curves based on each of the four different methods

**Usage**

```
fourCurv(x, y, bandwidth.method = "CV", lambdas)
```

**Arguments**

x	the explanatory variable - numeric
y	the response variable - numeric
bandwidth.method	the bandwidth method choice: CV or plug-in. Default is CV (for Local Linear and Local Constant estimators)
lambdas	tunes the tradeoff between the goodness of fit and the regularity of the spline - numeric value or numeric vector (for penalized B-spline estimator).

**Value**

Four graphs, one for each of the following methods : Local Linear, Local Constant, Cubic Splines and penalized B-splines.

**Examples**

```
#create a sample data frame
weights=c(500,600,1000,1150,1200,1260,1240,1300,1370,1500,2000,2100,2150,2500,
2800,2900,3050,3200,2980,3000,3300,3100,3200,3600,3500,3700,3900,3900,4000,
4200,3000,4500,4300,4900,4350,3700,4000,5000,4300)
age<-c(30,30,30,31,31,31,32,32,32,33,33,33,34,34,34,35,35,35,36,36,36,
37,37,37,38,38,38,39,39,39,40,40,40,41,41,41,42,42,42)
sample<-data.frame(weights,age)
colnames(sample)<-c("Weight in grams","Gestational Age in weeks")
x<-sample$`Gestational Age in weeks`
y<-sample$`Weight in grams`
```

```
fourCurv(x,y, lambdas=seq(1,10))
```

---

locNormCst	<i>Normal local constant estimator</i>
------------	--

---

### Description

Normal local constant estimator

### Usage

```
locNormCst(
  x,
  y,
  bandwidth.method = "CV",
  kernel = locpol::gaussK,
  cents = c(0.03, 0.25, 0.5, 0.75, 0.97),
  disp_window = TRUE,
  leg = FALSE,
  axes.lab = NULL
)
```

### Arguments

x	the explanatory variable - numeric
y	the response variable - numeric
bandwidth.method	the bandwidth method choice: CV or plug-in. Default is CV.
kernel	Kernel used to perform the estimation, see Kernels (from locpol).
cents	A numeric vector that represents the centiles calculated. Default is set to cents=c(0.03,0.25,0.5,0.75,0.97)
disp_window	Boolean. Should the scale of bandwidth be displayed (TRUE) or not (FALSE).
leg	Boolean. Should the legend be displayed (TRUE) or not (FALSE).
axes.lab	NULL or c("Nom_axe_X, Nom_axe_Y").

### Value

Plots the centile curves and returns a list object containing bandwidth value and estimated centiles values.



**Examples**

```
#create an example data frame
weights=c(500,600,1000,1150,1200,1260,1240,1300,1370,1500,2000,2100,2150,2500,
2800,2900,3050,3200,2980,3000,3300,3100,3200,3600,3500,3700,3900,3900,4000,
4200,3000,4500,4300,4900,4350,3700,4000,5000,4300)
age<-c(30,30,30,31,31,31,32,32,32,33,33,33,34,34,34,35,35,35,36,36,36,
37,37,37,38,38,38,39,39,39,40,40,40,41,41,41,42,42,42)
sample<-data.frame(age,weights)
colnames(sample)<-c("Gestational Age in weeks","Weight in grams")
x<-sample$`Gestational Age in weeks`
y<-sample$`Weight in grams`
#calculate the centile and plot the curves
locNormCst(x,y, kernel=locpol::gaussK)
```

locNormCstData

*Centile curves using local polynomial compared to noise data***Description**

Centile curves using local polynomial compared to noise data

**Usage**

```
locNormCstData(
  x,
  y,
  bandwidth.method,
  cents = c(0.03, 0.25, 0.5, 0.75, 0.97),
  data,
  leg = FALSE
)
```

**Arguments**

x	the explanatory variable - numeric
y	the response variable - numeric
bandwidth.method	the method chosen to calculate bandwidth. Could be cross validation or Plug-in. Default is set to CV.
cents	A numeric vector that represents the centiles calculated. Default is set to cents=c(0.03,0.25,0.5,0.75,0.97)
data	the noise data we want to compare
leg	Boolean. Should the legend be displayed (TRUE) or not (FALSE).

**Value**

Plots centile curves with local constant polynomial and displays them on the same figure as the noise data to be compared

**Examples**

```
#create an example data frame
example<-data.frame(sample(30:42,50,rep=TRUE),sample(800:5000,50,rep=TRUE))
colnames(example)<-c("Gestational Age in weeks","Weight in grams")
x<-example$`Gestational Age in weeks`
y<-example$`Weight in grams`
abnormal<-data.frame(sample(30:42,10,rep=TRUE),sample(800:5000,10,rep=TRUE))
colnames(abnormal)<-c("Gestational Age in weeks","Weight in grams")
locNormCstData(x,y,bandwidth.method="Plug-in",cents=c(0.03,0.25,0.50,0.75,0.97),data=abnormal)
```

locNormLin

*Normal local Linear estimator***Description**

Normal local Linear estimator

**Usage**

```
locNormLin(
  x,
  y,
  bandwidth.method = "CV",
  kernel = locpol::gaussK,
  cents = c(0.03, 0.25, 0.5, 0.75, 0.97),
  disp_window = TRUE,
  leg = TRUE,
  axes.lab = NULL
)
```

**Arguments**

x	the explanatory variable - numeric
y	the response variable - numeric
bandwidth.method	the bandwidth method choice: CV or plug-in. Default is CV.
kernel	Kernel used to perform the estimation, see Kernels (from locpol).
cents	A numeric vector that represents the centiles calculated. Default is set to cents=c(0.03,0.25,0.5,0.75,0.97)
disp_window	Boolean. Should the scale of bandwidth be displayed (TRUE) or not (FALSE).
leg	Boolean. Should the legend be displayed (TRUE) or not (FALSE).
axes.lab	NULL or c("Nom_axe_X, Nom_axe_Y").

**Value**

Plots the centile curves and returns a list object containing bandwidth value and estimated centiles values.

**Examples**

```
#create an example data frame
weights=c(500,600,1000,1150,1200,1260,1240,1300,1370,1500,2000,2100,2150,2500,
2800,2900,3050,3200,2980,3000,3300,3100,3200,3600,3500,3700,3900,3900,4000,
4200,3000,4500,4300,4900,4350,3700,4000,5000,4300)
age<-c(30,30,30,31,31,31,32,32,32,33,33,33,34,34,34,35,35,35,36,36,36,
37,37,37,38,38,38,39,39,39,40,40,40,41,41,41,42,42,42)
sample<-data.frame(age,weights)
colnames(sample)<-c("Gestational Age in weeks","Weight in grams")
x<-sample$`Gestational Age in weeks`
y<-sample$`Weight in grams`
#calculate the centile and plot the curves
locNormLin(x,y)
```

---

locNormLinData

*Centile curves using local linear polynomial compared to noise data*


---

**Description**

Centile curves using local linear polynomial compared to noise data

**Usage**

```
locNormLinData(
  x,
  y,
  bandwidth.method,
  cents = c(0.03, 0.25, 0.5, 0.75, 0.97),
  data,
  leg = FALSE
)
```

**Arguments**

x	the explanatory variable - numeric
y	the response variable - numeric
bandwidth.method	the method chosen to calculate bandwidth. Could be cross validation or Plug-in. Default is set to CV.
cents	A numeric vector that represents the centiles calculated. Default is set to cents=c(0.03,0.25,0.5,0.75,0.97)
data	the noise data we want to compare
leg	Boolean. Should the legend be displayed (TRUE) or not (FALSE).

**Value**

Plots centile curves with local linear polynomial using a Gaussian kernel and displays them on the same figure as the noise data to be compared

**Examples**

```
#create a sample data frame
sample<-data.frame(sample(30:42,50,rep=TRUE),sample(800:5000,50,rep=TRUE))
colnames(sample)<-c("Gestational Age in weeks","Weight in grams")
x<-sample$`Gestational Age in weeks`
y<-sample$`Weight in grams`
abnormal<-data.frame(sample(30:42,10,rep=TRUE),sample(800:5000,10,rep=TRUE))
colnames(abnormal)<-c("Gestational Age in weeks","Weight in grams")
locNormLinData(x,y,bandwidth.method="Plug-in",cents=c(0.03,0.25,0.50,0.75,0.97),data=abnormal)
```

---

polylocLin

*Polynomial local linear estimator*

---

**Description**

Polynomial local linear estimator

**Usage**

```
polylocLin(
  x,
  y,
  bandwidth.method = "CV",
  cents = c(0.03, 0.25, 0.5, 0.75, 0.97),
  disp_window = TRUE,
  leg = TRUE,
  axes.lab = NULL
)
```

**Arguments**

x	the explanatory variable - numeric
y	the response variable - numeric
bandwidth.method	the bandwidth method choice: CV or plug-in. Default is CV.
cents	A numeric vector that represents the centiles calculated. Default is set to cents=c(0.03,0.25,0.5,0.75,0.97)
disp_window	Boolean. Should the scale of bandwidth be displayed (TRUE) or not (FALSE).
leg	Boolean. Should the legend be displayed (TRUE) or not (FALSE).
axes.lab	NULL or c("Nom_axe_X, Nom_axe_Y").

**Value**

Plots the centile curves and returns a list object containing bandwidth value and estimated centiles values.

**Examples**

```
#create an example data frame
weights=c(500,600,1000,1150,1200,1260,1240,1300,1370,1500,2000,2100,2150,2500,
2800,2900,3050,3200,2980,3000,3300,3100,3200,3600,3500,3700,3900,3900,4000,
4200,3000,4500,4300,4900,4350,3700,4000,5000,4300)
age<-c(30,30,30,31,31,31,32,32,32,33,33,33,34,34,34,35,35,35,36,36,36,
37,37,37,38,38,38,39,39,39,40,40,40,41,41,41,42,42,42)
sample<-data.frame(age,weights)
colnames(sample)<-c("Gestational Age in weeks","Weight in grams")
x<-sample$`Gestational Age in weeks`
y<-sample$`Weight in grams`
#calculate the centile and plot the curves
polylocLin(x,y)
```

---

polylocLinData

*Polynomial local linear estimator compared to noise data*

---

**Description**

Polynomial local linear estimator compared to noise data

**Usage**

```
polylocLinData(
  x,
  y,
  bandwidth.method = "Plug-in",
  cents = c(0.03, 0.25, 0.5, 0.75, 0.97),
  data,
  leg = FALSE
)
```

**Arguments**

**x** the explanatory variable - numeric

**y** the response variable - numeric

**bandwidth.method** the bandwidth method choice: CV or plug-in. Default is CV.

**cents** A numeric vector that represents the centiles calculated. Default is set to cents=c(0.03,0.25,0.5,0.75,0.97)

**data** the noise data we want to compare

**leg** Boolean. Should the legend be displayed (TRUE) or not (FALSE).

**Value**

Plots the centile curves and data points to compare with.

**Examples**

```
#create a sample data frame
sample<-data.frame(sample(30:42,50,rep=TRUE),sample(800:5000,50,rep=TRUE))
colnames(sample)<-c("Gestational Age in weeks","Weight in grams")
x<-sample$`Gestational Age in weeks`
y<-sample$`Weight in grams`
abnormal<-data.frame(sample(30:42,10,rep=TRUE),sample(800:5000,10,rep=TRUE))
colnames(abnormal)<-c("Gestational Age in weeks","Weight in grams")
polylocLinData(x,y,bandwidth.method="Plug-in",cents=c(0.03,0.25,0.50,0.75,0.97),data=abnormal)
```

# Index

bandwidth, [2](#)  
bsplines, [3](#)  
bSplinesData, [4](#)

CentCurv, [5](#)  
compareCurv, [6](#)

fourCurv, [7](#)

locNormCst, [8](#)  
locNormCstData, [9](#)  
locNormLin, [10](#)  
locNormLinData, [11](#)

polylocLin, [12](#)  
polylocLinData, [13](#)