Qreg

A GAUSS Library for Computing Quantile Regression

Key Definitions and Implementation

Damien Jacomy, Jérôme Messines and Thierry Roncalli

Groupe de Recherche Opérationnelle, Crédit Lyonnais, France

Contents

1   Introduction

Quantile regression turns out to be a very suitable method when usual linear regression comes up against robustness problems. Moreover, the entire conditional distribution of a response Y given values x of a predictor X may prove as relevant as the only conditional *mean* function. That is why we try in this article to summarize key definitions and practical use of quantile regression models.

First, we detail basic principles of the method and its implementation thanks to linear programming. We try to solve the following problem: given a response Y to a predictor X, we use a linear model of the conditional τ-quantile of Y given $X = x$. Then, using the above-mentioned results, we explain how we can implement the methods proposed by YU and JONES [8] as regards local linear (or quadratic) regression and show the advantages of those techniques.

2   Main topic

Given a random variable Y and its distribution function $\mathbf{F}$, we denote by

$$\mathbf{Q}\left(\tau\right) = \inf\left\{y \mid \mathbf{F}\left(y\right) \geq \tau\right\} \tag{1}$$

the τth quantile of Y. The sample analogue q of $\mathbf{Q}\left(\tau\right)$ is called the τth sample quantile and may be formulated as the solution of the following optimization problem, given a random sample $(y_n)_{n=1,\ldots,N}$:

$$\min_{q \in \mathbb{R}} \sum_{\{n \mid y_n \geq q\}} \tau\left|y_n - q\right| + \sum_{\{n \mid y_n < q\}} (1 - \tau)\left|y_n - q\right| \tag{2}$$

For any $0 < \tau < 1$, we denote $\boldsymbol{\rho}_\tau\left(u\right) = u \times \left(\tau - \mathbf{1}_{[u<0]}\right)$, so that the problem may be formulated as follows

$$\min_{q \in \mathbb{R}} \sum_{n=1}^N \boldsymbol{\rho}_\tau\left(y_n - q\right) \tag{3}$$

which yields a natural generalization to the regression context. The linear conditional quantile function can be estimated by solving, for $X \in \mathbb{R}^K$ and $\beta \in \mathbb{R}^K$:

$$\min_{\beta \in \mathbb{R}^K} \sum_{n=0}^N \boldsymbol{\rho}_\tau\left(y_n - x_n^\top \beta\right) \tag{4}$$

3   Introducing M-estimators

3.1   Background on M-estimators

We use here the results presented in GOURIÉROUX and MONFORT [1]. Considering a parametric or semi-parametric model with parameter $\theta \in \Theta$ and observations $(X_n, Y_n)_{n=1,\ldots,N}$, we have:

Definition 1 *A M-estimator of a function $g\left(\theta\right)$ is the solution of an optimization problem in the form*

$$\min_{g \in g(\Theta)} \sum_{n=1}^N \Psi\left(Y_n, X_n; g\right) \tag{5}$$

The study of M-estimators is very similar to that of maximum likelihood estimators. The original idea is to replace the above-mentioned problem by the associated limit problem:

$$\min_{g \in g(\Theta)} \mathbb{E}_X \mathbb{E}_0 \Psi\left(Y, X; g\right) \tag{6}$$

where $\mathbb{E}_0$ is the expectation under the "true" conditional distribution of Y given $X = x$ and $\mathbb{E}_X$ is the expectation under the "true" marginal distribution of X.

The following theorem ensures convergence of M-estimators towards the "true" limit:

Theorem 1 *If*

1. *the pairs (X_n, Y_n) are i.i.d.,*

2. *$\mathcal{G} = g(\Theta)$ is a compact (or open) set,*

3. *Ψ is a continuous function of g, and its expectation under to the "true" distribution of (X, Y) exists for any g,*

4. *$\frac{1}{N}\sum_{n=1}^{N}\Psi(Y_n, X_n; g)$ is almost surely uniformly convergent on $\mathcal{G}$ to $\mathbb{E}_X\mathbb{E}_0\Psi(Y_n, X_n; g)$,*

5. *the only solution of the problem (6) is $g_\infty^0 = g_\infty(\theta_0)$, where θ_0 is the parameter of the "true" distribution,*

then, there exists a M-estimator that converges to $g(\theta_0)$ if and only if $\forall\theta \in \Theta$, $g_\infty(\theta) = g(\theta)$.

Thus when regularity conditions are established, we only have to prove that the solution of (6) is the expected one, whatever the value of the parameter.

3.2 Quantile regression and M-estimators

Let us fix $\tau \in\ ]0, 1[$, and study how the program (4) provides us with an estimation of $\mathbf{Q}_\tau(Y \mid X = x)$. Suppose that $Y_n = \mathbf{q}(X_n, \beta) + u_n$, where $\mathbf{q}$ is a continuous function of β and the random variables u_n are *i.i.d.* such that $\Pr\{u_n \leq 0\} = \tau$. We then have $\mathbf{Q}_\tau(Y \mid X = x; \beta) = \mathbf{q}(X, \beta)$. The point is to verify the hypothesis of Theorem 1 when estimating β with $\Psi(Y_n, X_n; \beta) = \boldsymbol{\rho}_\tau(Y_n - \mathbf{q}(X_n, \beta))$. Suppose now that

- the pairs (X_n, Y_n) are independent and identically distributed,

- and $\forall a \in \mathbb{R}$, $\mathbb{E}[|Y - a|]$ exists.

Following Theorem 1, the sequence of M-estimators $\hat{\beta}_N$ converges to β_0 if and only if β_0 is the only solution of

$$\min_{\beta \in \mathbb{R}^K} \Psi_\infty(Y, X; \beta) = \mathbb{E}_X\mathbb{E}_0\boldsymbol{\rho}_\tau(Y - \mathbf{q}(X, \beta)) \tag{7}$$

Proposition 2 *If the τ-quantile of $Y \mid X = x$ is unique and defined by $\mathbf{q}(X, \beta_0)$, and if β is entirely determined by the conditional τ-quantile [1], then β_0 is the only solution of $\min_{\beta \in \mathbb{R}^K}\Psi_\infty(Y, X; \beta)$.*

Proof. Let f and $\mathbf{F}$ be the density and distribution functions of the residuals u. Let a be any real number. We have:

$$
\begin{aligned}
\mathbb{E}\boldsymbol{\rho}_\tau(u - a) &= \int\left[\tau(u - a)\mathbf{1}_{[u-a>0]} + (\tau - 1)(u - a)\mathbf{1}_{[u-a\leq 0]}\right]f(u)\,\mathrm{d}u \\
&= \int\int\tau\mathbf{1}_{[u>t>a]}\,\mathrm{d}t \cdot f(u)\,\mathrm{d}u + \int\int(1 - \tau)\mathbf{1}_{[u\leq t\leq a]}\,\mathrm{d}t \cdot f(u)\,\mathrm{d}u \\
&\qquad\text{Interchanging symbols } \int, \text{ we get} \\
&= \int\left[\tau(1 - \mathbf{F}(t))\mathbf{1}_{[t>a]} + (1 - \tau)\mathbf{F}(t)\mathbf{1}_{[t\leq a]}\right]\,\mathrm{d}t \\
&= \int\left[(\tau - \mathbf{F}(t))\mathbf{1}_{[t>a]} + (1 - \tau)\mathbf{F}(t)\right]\,\mathrm{d}t \tag{8}
\end{aligned}
$$

The minimum of this expression is achieved for any a verifying $\Pr\{u \leq a\} = \tau$. The problem (6) may be reformulated with u:

$$\min_{\beta \in \mathbb{R}^K}\boldsymbol{\rho}_\tau(u + \mathbf{q}(X, \beta_0) - \mathbf{q}(X, \beta))$$

The solution is then defined by $\mathbf{q}(X, \beta_0) - \mathbf{q}(X, \beta) = 0$, i.e. β_0. ∎

[1] i.e. $[\forall x,\, h(x, \beta) = h(x, \beta_0)] \Rightarrow \beta = \beta_0$.

4   Estimation of the parameters by linear programming

We suppose here that we have for the conditional quantile function a linear statistical model in the form $\mathbf{q}(X, \beta) = X^\top \beta$. In this case, the optimization may be formulated as an usual linear programming problem.

4.1   Reformulation of the problem

Write Y with only positive terms:

$$
\begin{aligned}
Y_n &= X_n^\top \beta + u_n \\
&= \sum_{k=1}^{K} x_{n,k} \beta_k + u_n \\
&= \sum_{k=1}^{K} x_{n,k} \left(\beta_n^+ - \beta_n^- \right) + u_n^+ - u_n^-
\end{aligned}
\tag{9}
$$

The optimization program (4) can be written as

$$
\begin{aligned}
z &= \arg\min c^\top z \\
&\text{u.c.} \quad \left\{ \begin{array}{l} Az = y \\ z \geq 0 \end{array} \right.
\end{aligned}
\tag{10}
$$

where

$$
\begin{aligned}
X &= (X_1, \ldots, X_N)^\top \\
A &= (X, -X, I_N, -I_N) \\
y &= (Y_1, \ldots, Y_N)^\top \\
z &= \begin{pmatrix} \beta^+ \\ \beta^- \\ u^+ \\ u^- \end{pmatrix} \in \mathbb{R}^{2K+2N} \\
c &= \begin{pmatrix} \mathbf{0} \\ \mathbf{0} \\ \tau \mathbf{1} \\ (1-\tau)\mathbf{1} \end{pmatrix}
\end{aligned}
\tag{11}
$$

Thus, solving the former optimization problem consists in minimizing a linear function subject to linear constraints.

When considering the simplex method to solve the program, one is confronted with computational issues: how to save and manipulate the constraints matrix, when the number of observations becomes important — $\dim A \simeq N \times 2N$. Moreover, one has to deal with the well-known slowness of the procedure. That is why the interior-point method, rather fast when applied to large data sets, is more suitable.

4.2   Interior-point method

Theoretical developments of this method are extensively presented in BONNANS, GILBERT, LEMARÉCHAL and SAGASTIZÁBAL [2]. We refer to PORTNOY and KOENKER [5] for its application to quantile regression.

The original idea of interior-point methods is to replace a linear program with bounded variables, such as

$$
\min_x c^\top x \qquad \text{u.c. } x \geq 0
\tag{12}
$$

with a sequence of μ-programs

$$
\min_x c^\top x - \mu \sum_m \ln x_m
\tag{13}
$$

The logarithmic barrier functions accounts for positivity constraints on x coordinates, becoming infinite as x approaches the boundary of the feasible set. Starting from a feasible point, we perform a standard Newton optimization procedure to find the "right" direction and we choose the step that minimizes the barrier function. For more details, please refer to PORTNOY and KOENKER [5].

4.3   Application to quantile regression

Just know that for reasons of symmetry as well as computability, we introduce the dual program:

$$
\begin{aligned}
a \quad &= \quad \arg\max y^\top a \\
&\text{u.c.} \quad \left\{ \begin{array}{l} X^\top a = (1-\tau)\, X^\top \mathbf{1} \\ a \in [0,1]^N \end{array} \right.
\end{aligned}
\tag{14}
$$

Please refer to PORTNOY and KOENKER [5] for further details.

4.4   Implementation

The procedures of the **Qreg** library are adapted from a program by Roger Koenker. This program may be found at

$$\texttt{http://www.econ.uiuc.edu/\~roger/research/rqn}$$

5   Local linear (or quadratic) regression

5.1   Principle

We want to estimate at any point the conditional τ^{th}-quantile of Y given $X = x$. Our key article is YU and JONES [8]. The most natural idea for such an estimation is to choose observations of Y such as the associated X is near x and to calculate the τth-quantile on this sample. Denoting h the bandwidth, it implies calculating the quantile of observations of Y with weight 1 if distance from X to x is at most h and 0 else.

In the same approach, we can consider performing kernel weighted local linear fitting, allowing all observations to be taken into account. Denoting $\mathcal{K}$ the kernel function (Gaussian, Epanechnikov, etc.), we calculate the quantile of Y thanks to the program (2) in a weighted version. Our quantile q is then obtained by solving the following problem

$$
\min_{q \in \mathbb{R}} \sum_{n=1}^{N} w_n \boldsymbol{\rho}_\tau \left(y_n - q \right)
\tag{15}
$$

with

$$
w_n = \mathcal{K}\left(\frac{x - x_n}{h} \right)
\tag{16}
$$

h is now a scale parameter controlling the amount of smoothing applied to the data. In practice, this approach gives questionable results when there is a strong functional dependency between x and the quantile of $Y \mid X = x$ — for example, when the dependency is linear with steep gradient. Overcoming this bias implies performing local linear regression.

Let us define a local linear model for the quantile, centered in x:

$$
\mathbf{Q}_\tau \left(Y \mid X = x + \delta \right) \simeq a \cdot \delta + q
\tag{17}
$$

We want to estimate the constant q of this regression ($\delta = 0$). The problem to solve is then:

$$
\min_{(q,a) \in \mathbb{R}^2} \sum_{n=1}^{N} w_n \boldsymbol{\rho}_\tau \left(y_n - a\left(x_n - x\right) - q \right)
\tag{18}
$$

We may also consider a quadratic model defined by

$$
\min_{(q,a,b) \in \mathbb{R}^3} \sum_{n=1}^{N} w_n \boldsymbol{\rho}_\tau \left(y_n - a\left(x_n - x\right) - b\left(x_n - x\right)^2 - q \right)
\tag{19}
$$

5.2 Implementation

The choice of h for the local linear regression is extensively discussed in Yu and Jones [8] and Ruppert, Sheather and Wand [9]. In practice, for thousands of observations, the parameter h does not have a strong influence. Yu and Jones propose a rule-of-the-thumb method for the estimation of h in quantile regression given the optimal h for local O.L.S. of Y on X, noted h_{mean}. They refer to Ruppert, Sheather and Wand [9] for the choice of h_{mean} (This method requires the calculus of the second derivative of $\mathbb{E}(Y \mid X = x)$ in x).

Our procedures only use the formula in Ruppert, Sheather and Wand [9] for h_{mean}, without calculating the second derivative, and we follow Yu and Jones [8] to choose the optimal bandwidth h. A small correction is possible via a global variable, but is not often needed.

Results are presented in Figures 1 and 2. We find that the linear regression gives correct results in quasi-affine zones. But when the second derivative is high enough, results are much more questionable. That is why in this case quadratic regression seems more suitable.

5.3 Beyond the **Normal** distribution

When we perform a linear regression, we assume that (Y, X) is a gaussian vector. When Y and X are not normal-distributed but the copula is Normal, linear regression could be performed with some modifications (the regression is done in the Ψ projection space). When copula is not Normal, some preliminary tests show that local regression gives very good results (in particular for the local quadratic regression). For more details, please refer to Costinot, Roncalli and Teiletche [10].

References

[1] Gouriéroux, C and A. Monfort [1996], Statistique et Modèles Econométriques, *second edition*, Economica, Paris
(English translation: Statistics and Econometric Models, Cambridge University Press, Cambridge)

[2] Bonnans, J.F., J.C. Gilbert, C. Lemaréchal and C. Sagastizábal [1997], Optimisation Numérique — Aspects théoriques et pratiques, Springer-Verlag, Berlin

[3] Koenker, R. and G. Basset [1978], Regression quantiles, *Econometrica*, **46**, 33-50

[4] Koenker, R. [2000], Galton, Edgeworth, Frisch, and prospects for quantile regression in econometrics, *Journal of Econometrics*, **95**, 347-374

[5] Portnoy, S. and R. Koenker [1997], The Gaussian hare and the Laplacean tortoise: computability of ℓ_1 vs ℓ_2 regression estimators, *Statistical Science*, **12**, 279-300

[6] Buchinsky, M. [1997], Recent advances in quantile regression models: a practical guideline for empirical research, forthcoming in *Journal of Human Resources*

[7] Press, W.H., S.A. Teukolsky, W.T. Vetterling and B.P. Flannery [1992], Numerical Recipes in Fortran, second edition, Cambridge University Press, Cambridge

[8] Yu, K. and M.C. Jones [1998], Local linear quantile regression, *Journal of the American Statistical Association*, **93**, 228-237

[9] Ruppert, D., S.J. Sheather and M.P. Wand [1995], An effective bandwidth selector for local least squares regression, *Journal of the American Statistical Association*, **90**, 1257-1270

[10] Costinot, A., T. Roncalli and J. Teiletche [2000], Revisiting the dependence between financial markets with copulas, Groupe de Recherche Opérationnelle, Crédit Lyonnais, *Working Paper*
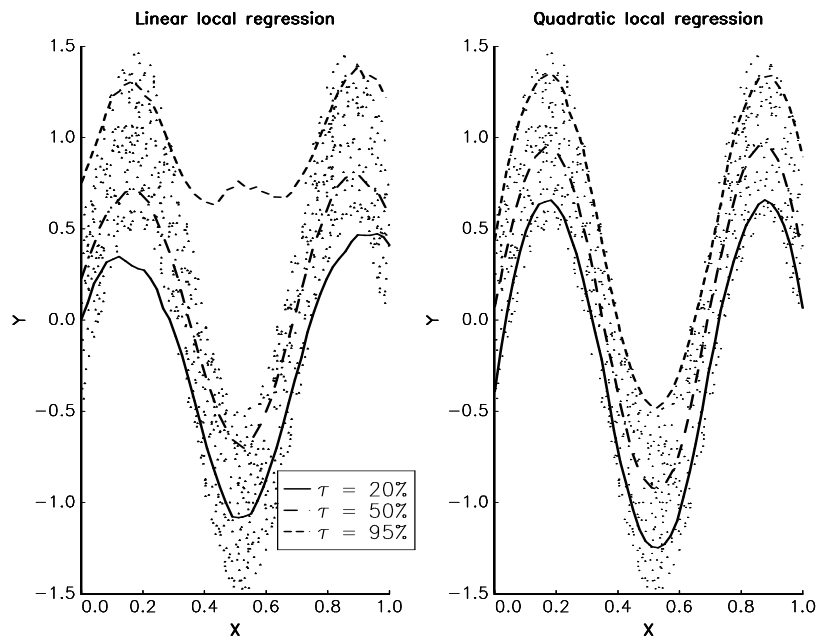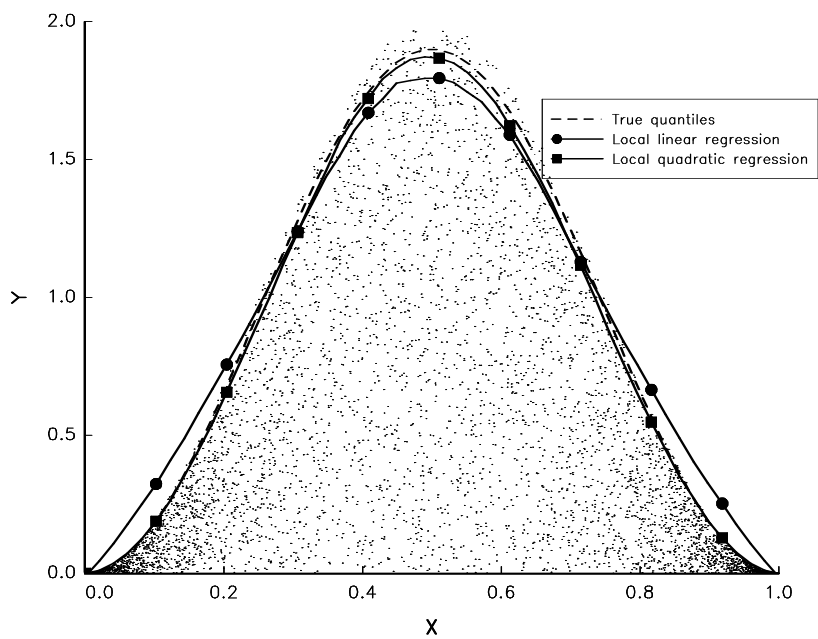
Figure 1: Linear vs quadratic regression (I)

Figure 2: Linear vs quadratic regression (II)

A The **GAUSS Qreg** library

A.1 Introduction

A.1.1 Installation

1. The file *libQreg.zip* is a zipped archive file. Copy this file under the root directory of GAUSS, for example **C:\GAUSS**.

2. Unzip it with archive mode. It is automatically recognized by WinZip. With Unzip or PKunzip, use the -d flag

   pkunzip -d libQreg.zip

   Directories will then be created and files will be copied over them:

target_path	*readme.qreg*
target_path**examples****qreg**	examples and tutorial files
target_path**lib**	library file
target_path**src****qreg**	source code files

3. Run GAUSS. **Log on to the *src\qreg* directory**[2] and add the path to the library file *qreg.lcg* in the following way:

   lib qreg /addpath

A.1.2 Getting started

Gauss 3.2 for OS/2, Windows NT/95 or Unix is required to use the **Qreg** routines.

A.1.2.1   The file *readme.Qreg*   The file *readme.Qreg* contains last minute information on the **Qreg** procedures. Please read it before using them.

A.1.2.2   Setup   In order to use these procedures, the **Qreg** library must be active. This is done by including **Qreg** in the LIBRARY statement at the top of your program:

   library Qreg;

To reset global variables in subsequent executions of the program, the following instruction should be used:

   setQreg;

If you plan to make any right-hand reference to the global variables, you will also need the statement:

   #include *target_path***src****qreg****qreg.ext;**

The **Qreg** version number is stored as a global variable:

_Qreg_ver	3×1 matrix where the first element indicates the major version number, the second element the minor version number, and the third element the revision number

---

[2]You may use the commands ChangeDir or chdir. Note that you can verify that you are in the *src\qreg* directory with the cdir(0) command.

A.1.3 What is **Qreg**?

Qreg is a GAUSS library for computing quantile regression. **Qreg** contains the procedures whose list is given below. See the command reference part for a full description.

- **localQreg**: Local Quantile Regression.

- **Qreg**: Linear Quantile Regression.

- **setQreg**: Resets the global variables.

- **_Qreg_solveLP1**: Solves the dual program using using interior-point method.

- **_Qreg_solveLP2**: Solves the dual program using using QP method.

A.1.4 Using Online Help

Qreg library supports Windows Online Help. Before using the browser, you have to verify that the **Qreg** library is activated by the `library` command.

A.2 Command reference

The following global variables and procedures are defined in **Qreg**. They are the *reserved words* of **Qreg**.

```
localqreg, _localqreg, qreg, _qreg_algr, _qreg_big, _qreg_eps, _qreg_h, _qreg_maxit, _qreg_mtd,
_qreg_printiters, _qreg_sigma, _qreg_solvelp1, _qreg_solvelp2, _qreg_ver, setqreg
```

The default global control variables are

_Qreg_algr	1
_Qreg_big	10^{20}
_Qreg_eps	0.0001
_Qreg_h	1
_Qreg_maxit	50
_Qreg_mtd	1
_Qreg_PrintIters	0
_Qreg_sigma	0.99995

localQreg

■ **Purpose**
   Local quantile regression.

■ **Format**
   q = localQreg(Y,X,tau,xstar);

■ **Input**
Y	$N \times 1$ vector, dependent variable
X	$N \times 1$ vector, independent variable
tau	$M \times 1$ vector, quantile levels
xstar	$P \times 1$ vector, quantile points

■ **Output**
q	$P \times M$ matrix, estimated quantile $Y \mid X = x^\star$

■ **Globals**
__altnam	$P \times 1$ vector, variable names (default = 0)
__output	scalar, (default = 1)
	1 — print the statistics
	0 — no printing
_Qreg_algr	scalar, LP algorithm (default = 1)
	1 for the interior-point method
	2 for the QP method
_Qreg_h	scalar, multiplicative factor of the default bandwidth (default = 1)
_Qreg_mtd	scalar, regression type (default = 1)
	1 for linear regression
	2 for quadratic regression

■ **Remark**
   See the procedures _**Qreg_SolveLP1** and _**Qreg_SolveLP2** for the optimization step.

■ **Source**
   src/qreg.src

Qreg

■ **Purpose**

Linear quantile regression.

■ **Format**

{beta,u_plus,u_minus} = Qreg(Y,X,tau,w);

■ **Input**

Y	$N \times 1$ vector, dependent variable
X	$N \times K$ vector, independent variables
tau	$M \times 1$ vector, quantile levels
w	$N \times 1$ vector, containing weights (0 for uniform weights)

■ **Output**

beta	$(K + 1) \times M$ matrix, estimated coefficients
u_plus	$N \times M$ matrix, positive part of residuals
u_minus	$N \times M$ matrix, negative part of residuals

■ **Globals**

__altnam	$P \times 1$ vector, variable names (default = 0)
__output	scalar, (default = 1)
	1 — print the statistics
	0 — no printing
_Qreg_algr	scalar, LP algorithm (default = 1)
	1 for the interior-point method
	2 for the QP method

■ **Remark**

See the procedures `_Qreg_SolveLP1` and `_Qreg_SolveLP2` for the optimization step. The linear regression is

$$Y = \beta_0 + \beta X + u$$

and the procedure returns the vector $\begin{bmatrix} \hat{\beta}_0 & \hat{\beta} \end{bmatrix}^\top$.

■ **Source**

src/qreg.src

setQreg

■ **Purpose**

Resets the global variables declared in *qreg.dec*.

■ **Format**

setQreg;

■ **Remark**

The default global control variables are

`_Qreg_algr`	1
`_Qreg_big`	10^{20}
`_Qreg_eps`	0.0001
`_Qreg_h`	1
`_Qreg_maxit`	50
`_Qreg_mtd`	1
`_Qreg_PrintIters`	0
`_Qreg_sigma`	0.99995

■ **Source**

src/qreg.src

_Qreg_solveLP1 & _Qreg_solveLP2

■ **Purpose**

Solves the linear program

$$\min_x c^\top x \qquad \text{u.c. } Ax = b \text{ and } 0 \leq x \leq x^+$$

■ **Remark**

We do not describe these procedures because they are internally used by `Qreg` and `localQreg`. But if you want to use them, do not forget that they returns the dual solution (not the primal solution).

■ **Source**

src/qreg.src

A.3   Tutorial

- This first example shows the use of the different gloval variables.

```
new;
library Qreg,pgraph;
setQreg;

cls;

N = 100;
X = rndu(N,2);
Y = 12 + 3*X[.,1] + 5*X[.,2] + .2*rndn(rows(X),1);
tau = 0.05|0.50|0.95|0.75;

output file = qreg1.out reset;

_Qreg_algr = 1;       /* Interior-point method      */
_Qreg_PrintIters = 0; /* Do not print the iterations */
__output = 2;         /* Print the statistics       */
{beta,u_plus,u_minus} = Qreg(Y,X,tau,0);

_Qreg_PrintIters = 3; /* Printing every 3 iterations */
{beta,u_plus,u_minus} = Qreg(Y,X,tau,0);

_Qreg_algr = 2;       /* QP method                  */
{beta,u_plus,u_minus} = Qreg(Y,X,tau,0);

__altnam = ''First'' | ''Second'';   /* Use variable names */
{beta,u_plus,u_minus} = Qreg(Y,X,tau,0);

output off;


rndseed 123;

N = 1000;
X = 10*rndu(N,1)-5;
Y = 5 + 2*X + rndn(rows(X),1)*10;
tau = 0.05|0.50|0.95|0.75;
_Qreg_algr = 1;
{beta,u_plus,u_minus} = Qreg(Y,X,tau,0);

xstar = seqa(-5,10/200,201);
q = beta[1,.] + beta[2,.] .* xstar;

graphset;
  begwind;
  makewind(9,6.855,0,0,1);
  makewind(9,6.855,0,0,1);

  fonts(''simplex simgrma'');

  setwind(1);
    _pdate = 0; _pframe = 0; _pnum = 2;
    _plctrl  = 15; _pltype  = 6; _plwidth = 5; _pstype = 8|9|10|11;
    xlabel(''\214X''); ylabel(''\214Y'');
    xtics(-5,5,1,1);
    ytics(-20,30,10,1);
    _plegstr = ''\202t\201 = 5%\000\202t\201 = 50%\000\202t\201 = 95%\000\202t\201 = 75%'';

    _plegctl = {2 5 6.5 1};
    xy(xstar,q);

    _plegctl = 0;

  setwind(2);
    _plctrl  = -1; _pstype  = 3; _psymsiz = 0.4;
    xy(X,Y);
```

```
        graphprt(''-c=1 -cf=qreg1.ps'');

    endwind;
```

```
===============================================================================
  Qreg - Linear Quantile Regression   Version 1.0.0        4/27/2001   5:35 pm
===============================================================================

Total observations:                                        100
Number of variables:                                         2

VAR. / tau (in %)    5.000    50.000    95.000    75.000
-------------------------------------------------------------
CONSTANT         11.7837   11.9274   12.2430   12.0341
X01               2.8421    3.0253    3.0264    3.0195
X02               4.8994    5.0403    5.0190    5.0483

iter. / gap / mu:          3       1.8637755       0.009758235
iter. / gap / mu:          6      0.01000313    1.3313893e-005
iter. / gap / mu:          9   1.4972544e-005   3.4681612e-008

iter. / gap / mu:          3      0.21025711     0.0001474261
iter. / gap / mu:          6   0.00045417086   3.9935013e-007

iter. / gap / mu:          3       3.0758464      0.012127431
iter. / gap / mu:          6      0.08261169    0.00014484139
iter. / gap / mu:          9   5.9193175e-007   1.0355861e-014

iter. / gap / mu:          3      0.85150539    0.00045107856
iter. / gap / mu:          6     0.011686899   7.0206601e-006

===============================================================================
  Qreg - Linear Quantile Regression   Version 1.0.0        4/27/2001   5:35 pm
===============================================================================

Total observations:                                        100
Number of variables:                                         2

VAR. / tau (in %)    5.000    50.000    95.000    75.000
-------------------------------------------------------------
CONSTANT         11.7837   11.9274   12.2430   12.0341
X01               2.8421    3.0253    3.0264    3.0195
X02               4.8994    5.0403    5.0190    5.0483

===============================================================================
  Qreg - Linear Quantile Regression   Version 1.0.0        4/27/2001   5:35 pm
===============================================================================

Total observations:                                        100
Number of variables:                                         2

VAR. / tau (in %)    5.000    50.000    95.000    75.000
-------------------------------------------------------------
CONSTANT         11.7837   11.9274   12.2430   12.0340
X01               2.8421    3.0253    3.0264    3.0196
X02               4.8994    5.0403    5.0190    5.0485

===============================================================================
  Qreg - Linear Quantile Regression   Version 1.0.0        4/27/2001   5:35 pm
===============================================================================

Total observations:                                        100
Number of variables:                                         2

VAR. / tau (in %)    5.000    50.000    95.000    75.000
-------------------------------------------------------------
CONSTANT         11.7837   11.9274   12.2430   12.0340
First             2.8421    3.0253    3.0264    3.0196
Second            4.8994    5.0403    5.0190    5.0485
```

- We obtain Figure 1 with the following program:

```
new;
library Qreg,pgraph;
setQreg;

N = 1000;
X = rndu(N,1);
Y = sin(9*X) + (rndu(N,1) - 0.5);
tau = 0.2|0.5|0.95;
P = 50;
xstar = seqa(0,1/(P-1),P);

__output = 2;

_qreg_mtd = 1;
q1 = localQreg(Y,X,tau,xstar);
_qreg_mtd = 2;
q2 = localQreg(Y,X,tau,xstar);
```
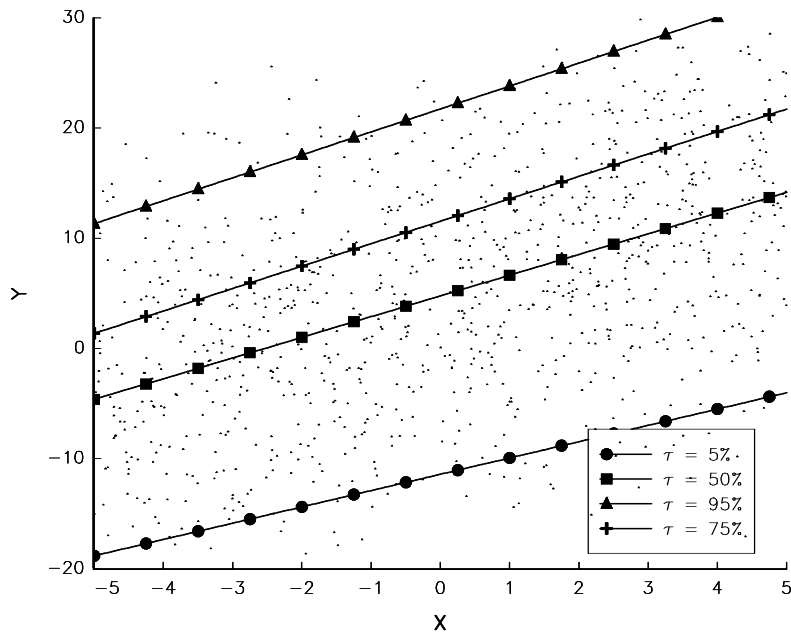
Figure 3: Linear quantile regression

```
graphset;
  begwind;
  makewind(9/2,6.855,0,0,1);
  makewind(9/2,6.855,0,0,1);
  makewind(9/2,6.855,9/2,0,1);
  makewind(9/2,6.855,9/2,0,1);

  _pdate = 0; _pframe = 0; _pnum = 2;
  _paxht = 0.25; _pnumht = 0.25; _ptitlht = 0.25; _plwidth = 5;
  xlabel(''\214X''); ylabel(''\214Y'');
  xtics(0,1,0.2,1);
  ytics(-1.5,1.5,0.5,1);
  fonts(''simplex simgrma'');

  setwind(1);
    title(''\214Linear local regression'');
    _pltype = 6|1|3; _plctrl = 0;
    _plegstr = ''\202t\201 = 20%\000\202t\201 = 50%\000\202t\201 = 95%'';
    _plegctl = {2 6 6 1};
    xy(xstar,q1);

    _plegctl = 0;

  setwind(2);
    _plctrl  = -1; _pstype  = 3; _psymsiz = 0.75;
    xy(X,Y);

  setwind(3);
    title(''\214Quadratic local regression'');
    _pltype = 6|1|3; _plctrl = 0;
    xy(xstar,q2);

  setwind(4);
    _plctrl  = -1; _pstype  = 3; _psymsiz = 0.5;
```

16

```
    xy(X,Y);

  graphprt(''-c=1 -cf=qreg2.ps'');

  endwind;
```

- We obtain Figure 2 with the following program:

```
new;
library Qreg,pgraph;
setQreg;

__output = 0;

N = 5000;
X = rndu(n, 1);
Y = rndu(rows(x),1) .* (cos(2*pi*X-pi)+1);
tau = 0.95;
P = 50;
xstar = seqa(0,1/(p-1),P);

_Qreg_mtd = 1;
q1 = localQreg(Y,X,tau,xstar);
_Qreg_mtd = 2;
q2 = localQreg(Y,X,tau,xstar);
q = q1~q2;

graphset;
  begwind;
  makewind(9,6.855,0,0,1);
  makewind(9,6.855,0,0,1);

  setwind(1);
    _pdate = 0; _pframe = 0; _pnum = 2;
    _plctrl  = 0|5|5; _pltype  = 3|6|6; _plwidth = 5;
    _pstype = 8|8|9;
    xlabel(''\214X''); ylabel(''\214Y'');
    ytics(0,2,0.5,1);
    xtics(0,1,0.2,1);
    _plegstr = ''True quantiles\000Local linear regression\000Local quadratic regression'';
    _plegctl = {2 4 6 5};
    xy(xstar,tau*(cos(2*pi*xstar-pi)+1)~q);

    _plegctl = 0;

  setwind(2);
    _plctrl  = -1; _pstype  = 3; _psymsiz = 0.2;
    xy(X,Y);

    graphprt(''-c=1 -cf=qreg3.ps'');

  endwind;
```