Journal of Statistical Software

September 2025, Volume 114, Issue 8. doi: 10.18637/jss.v114.i08

Quantile Regression under Limited Dependent Variable in Stata

Javier Alejo
Universidad de la República

Gabriel Montes-Rojas
CONICET-Universidad de Buenos Aires

Abstract

This article develops a Stata command, `ldvqreg`, to estimate quantile regression models for the cases of censored (with lower and/or upper censoring) and binary dependent variables. The estimator is implemented using a smoothed version of the quantile regression objective function. Simulation exercises show that it correctly estimates the parameters and it should be implemented instead of the available quantile regression methods when censoring is present. Different empirical applications illustrate these methods.

Keywords: censoring, binary model, quantile regression, Stata.

1. Introduction

Quantile regression (QR) is an important method for modeling heterogeneous effects. It allows to study generalized regression models by focusing on the quantiles of the conditional distribution of an outcome variable, controlling for observable covariates. It has been applied in many empirical settings to provide a model-free (i.e. it does not require to specify the distribution of the random variables) and semi-parametric alternative to mean-based regression models (see Koenker (2005) for an extensive review). Let $\{y_i, \boldsymbol{x}_i\}, i = 1, 2, \ldots, n$ be a random sample where y_i is an outcome variable of interest and $\boldsymbol{x}_i$ is a set of covariates. QR studies linear models of the form

$$Q_\tau(y_i|\boldsymbol{x}_i) = \boldsymbol{x}_i'\boldsymbol{\beta}(\tau),$$

for $\tau \in (0, 1)$, where $Q_\tau(y|x)$ refers to the τ-th conditional quantile or percentile of the conditional distribution of y conditional on x. The parameters $\frac{\partial Q_\tau(y|\boldsymbol{x})}{\partial \boldsymbol{x}} = \boldsymbol{\beta}(\tau)$ (assuming continuous covariates, otherwise consider a discrete change) are of interest here as they represent the conditional effect of covariates on different τ quantiles of the outcome variable.

In a broad sense, a limited variable is understood as a continuous latent variable whose

range has been modified. Two classic examples are binary and censored dependent variable regression models. Although the latter is usually treated as separated, both belongs to the set of general limited dependent variable (LDV) models. Empirical set-ups where this occurs are very common, see Section IV in Wooldridge (2010). Mean-based regression models are analyzed in many ways with respect to censoring and truncation of the dependent variable. Typical examples in applied statistics and econometrics include tobit-type models, as well as logit and probit models. In those models, the censoring or truncation mechanism makes inference and point estimation more complex, as the researcher requires an appropriate model for interpretation of the effects (Gaussian distribution for tobit and probit, logistic for logit).

This issue becomes more interesting in QR models, where the censoring and truncation mechanisms interfere with the heterogeneity analysis. This paper provides an integrated package that allows to estimate LDV regression models for QR models in Stata (StataCorp 2019), thus allowing the researcher to separately identify the heterogeneity impact analysis from QR together with different restrictions on the domain of the dependent variable.

First, it considers lower and upper censoring by developing the counterpart of a mean-based tobit model for QR. Censored QR (CQR) has been studied in several papers, including to cite a few Powell (1984, 1986), Buchinsky (1991), Buchinsky and Hahn (1998), Chay and Powell (2001) and Chernozhukov and Hong (2002). See Fitzenberger (1997) for a literature review. Although most applications in economics involve only one type of censoring, there are some cases of double censoring. To mention a few, Sun (2006, Chapter 8), Wichert and Wilke (2008), and Lin, He, and Portnoy (2012) provide examples.

Second, it provides an estimator for semi-parametric binary regression. This has been studied in the seminal papers of Manski (1985, 1991) and Kordas (2006) among others to develop the binary QR (BQR) estimator. This allows to apply QR models in a wide variety of frameworks where the dependent variable is binary. For instance, a binary model to predict the conditional probability of an event is usually an important input in treatment effects literature using Propensity-Score estimators (via matching and/or weighting). QR provides a general framework for studying treatment heterogeneity.

The key characteristic of these estimators with LDV is that quantile models are invariant to monotone non-decreasing transformations. Thus the conditional effects of covariates can be recovered from the transformed model. Moreover, one important common feature is that the usual algorithm for QR involves linear programming (see for instance the **qreg** package in Stata). Recent applications of QR emphasize that the estimators are improved in both asymptotics and numerical accuracy if the objective function is replaced by a smooth counterpart, see e.g., Horowitz (1992), Kordas (2006), Kaplan and Sun (2017) and de Castro, Galvao, Kaplan, and Liu (2019). The proposed package follows this strategy.

Alternative procedures have been developed to accommodate for censoring in QR in different softwares. Here we briefly list available codes in Stata and R (R Core Team 2025).

In Stata there are several related options. Jolliffe, Krushelnytskyy, and Semykina (2000) (**clad** package) uses Buchinsky (1991) and Buchinsky and Hahn (1998) algorithm to compute QR with censoring. Baker (2013) uses Monte Carlo integration techniques to accomplish the same (**mcmccqreg** package). Another alternative is the `cqiv` command developed by Chernozhukov, Fernandez-Val, Han, and Kowalski (2012). Our command contributes to this list. To our knowledge, there are no available estimators for Manski's type semi-parametric binary regression using BQR.

In R, censored QR estimators can be implemented using the Frumento (2021) CRAN package **ctqr** and some features incorporated in the Koenker (2021) QR package **quantreg**. See Koenker (2008) and Frumento and Bottai (2017) for a detailed description of the R implementation. Also within the **quantreg** there is an option to implement binary QR estimation, `rq.bin`.

This paper describes software material for Stata. It provides an integrated framework to implement LDV models in QR, making use of smoothing techniques to optimize the estimation algorithm. In particular, it develops a package to implement censored QR and binary QR.

The remainder of the paper is organized as follows. Section 2 presents the censored and binary QR models. Section 3 summarizes the Stata command features. Section 4 present numerical simulations and an empirical application to women's labor supply in Uruguay. Section 5 concludes.

2. LDV models in quantile regression

Consider the following conditional τ-quantile model,

$$Q_\tau(y_i^*|\boldsymbol{x}_i) = \boldsymbol{x}_i'\boldsymbol{\beta}(\tau),$$

where y_i^* is a latent unobservable variable and x_i corresponds to observable covariates. Assume that we have another observable variable $y_i = h(y_i^*)$, where $h(\cdot)$ is a non-decreasing monotone transformation, which is defined as a LDV. The main feature in these models is that we observe y_i but not y_i^*. However, by a common characteristic in quantile models, the so-called the quantile invariance property implies that $Q_\tau[h(y^*)|\boldsymbol{x}] = h[Q_\tau(y^*|\boldsymbol{x})]$, then $Q_\tau[y|\boldsymbol{x}] = h[Q_\tau(y^*|\boldsymbol{x})] = h[\boldsymbol{x}'\boldsymbol{\beta}(\tau)]$. That is, the observable variable quantile is a transformed version of the original linear model.

This model includes as particular cases of LDV the *censored quantile regression* (CQR) and the *binary dependent variable regression* (BQR) models. We study these two cases in the following sections.

2.1. Censored quantile regression

Consider the case where there is an upper censoring (at value c_H) and lower censoring (at value c_L, for $c_H > c_L$) of the dependent variable such that

$$y_i = \begin{cases} c_L & if\ y_i^* < c_L \\ y_i^* & if\ c_H \geq y_i^* \geq c_L \\ c_H & if\ y_i^* > c_H \end{cases}$$

For this case $y = h(y^*) = min[max(y^*, c_L), c_H]$, which is a non-decreasing monotone transformation. This model supports the case of no or partial censoring if we consider $c_L = -\infty$ and/or $c_H = +\infty$.

Powell (1984, 1986) proposes to estimate $\boldsymbol{\beta}(\tau)$ by

$$\hat{\boldsymbol{\beta}}(\tau) = \arg\min_{\boldsymbol{b}\in\mathbb{R}^K}\ n^{-1}\sum_{i=1}^{n}\rho_\tau\{y_i - \min[\max(\boldsymbol{x}_i'\boldsymbol{b}, c_L), c_H]\}$$

where $\rho_\tau(u)$ is the check function as in Koenker and Bassett (1978). This is defined as the censored quantile regression (CQR) model.

2.2. Binary quantile regression

Consider now the case of a binary dependent variable model with

$$y_i = \begin{cases} 0 & if\ y_i^* \leq 0 \\ 1 & if\ y_i^* > 0 \end{cases}$$

Note that this is also a non-decreasing monotone transformation of the dependent variable, $y = h(y^*) = 1(y^* > 0)$. With this idea Manski (1975, 1985) proposes a maximum score estimator based on:

$$\hat{\boldsymbol{\beta}}(\tau) = \arg\min_{\boldsymbol{b}:\|\boldsymbol{b}\|=1} n^{-1}\sum_{i=1}^{n}\rho_\tau[y_i - I\{\boldsymbol{x}_i'\boldsymbol{b} \geq 0\}].$$

which can be written as

$$\hat{\boldsymbol{\beta}}(\tau) = \arg\max_{\boldsymbol{b}:\|\boldsymbol{b}\|=1} n^{-1}\sum_{i=1}^{n}[y_i - (1-\tau)]I\{\boldsymbol{x}_i'\boldsymbol{b} \geq 0\}.$$

This is defined as the binary quantile regression (BQR) model.

2.3. Smoothed quantile regression

Both, the censored and the binary cases, share a common feature. The theoretical formulation provides estimators with a proper characterization, identification and consistent estimation (asymptotically normal only for the former, BQR requires further assumptions). However, its numerical performance in applied cases is very poor. In particular, the maximization problem does not provide satisfactory numerical solutions in general. This is a common feature in some variants of QR models, and the consensus in the literature is to provide smooth objective functions alternatives.

For our purposes, Horowitz (1992) and Kordas (2006) propose to smooth the objective function by using

$$K(\boldsymbol{x}_i'\boldsymbol{b}/h_n),$$

the integral of a kernel function with h_n bandwidth instead of $I\{\boldsymbol{x}_i'\boldsymbol{b} \geq 0\}$. This provides remarkable improvements in applied cases. Our proposed estimator follows this strategy. The smoothing function we use is the cumulative distribution of a Gaussian kernel. We also allow for logit, Epanechnikov and bi-weight as additional options in the command. One important issue is that for the binary choice model, Manski's estimator is not asymptotically normal, but the smoother version as proposed by Kordas (2006) is.

Both in the case of censored and binary regression models we use the same heuristic rule for the choice of bandwidth: the same formula used by Stata to estimate densities with kernel functions. This is,

$$h_n = \frac{0.9 \cdot \hat{\sigma}_u}{n^{1/5}}$$

where $\hat{\sigma}_u$ is an estimate of the standard deviation of the latent variable conditional distribution. In the case of censored data we use the $\hat{\sigma}_u$ estimated by the tobit model, while in the case of the binary data we set $\hat{\sigma}_u = 1$ (the usual normalization of this parameter in the probit model).

2.4. Prediction of censored quantiles and probabilities

An important issue in censored models is the appropriate prediction exercise.

For the CQR model we can compute the prediction for a given censured quantile τ as

$$\hat{Q}_\tau(y|\boldsymbol{x}_i) = min\{max[\boldsymbol{x}_i'\hat{\boldsymbol{\beta}}(\tau); c_L]; c_H\},$$

where $\hat{\boldsymbol{\beta}}(\tau)$ are the CQR estimates.

Following Kordas (2006), for the BQR model, we consider the probability of $y = 1$, which corresponds to $y^* > 0$. This can be estimated by computing $\boldsymbol{x}'\boldsymbol{\beta}(U) > 0$ where $U \sim U(0, 1)$. Given that for the binary case $P(y = 1|\boldsymbol{x}) = E(y|\boldsymbol{x})$, then $P(y = 1|\boldsymbol{x}) = E[1(\boldsymbol{x}'\boldsymbol{\beta}(U) > 0)|]$. Therefore,

$$P(y = 1|\boldsymbol{x}) = \int_0^1 I\{\boldsymbol{x}'\boldsymbol{\beta}(\tau) > 0\}d\tau.$$

Then, this can be estimated by a grid of quantile indexes $\{\tau_1, \tau_2, \ldots, \tau_m\}$ by computing

$$\hat{P}(y = 1|\boldsymbol{x}_i) = m^{-1}\sum_{j=1}^{m} I\{\boldsymbol{x}_i'\hat{\boldsymbol{\beta}}(\tau_j) > 0\}, \tag{1}$$

where $\hat{\boldsymbol{\beta}}(\tau)$ is the corresponding BQR estimator. An smoothed version of Equation 1 replaces the indicator function by the integral of the kernel $K(\cdot)$:

$$\hat{P}(y = 1|\boldsymbol{x}_i) = m^{-1}\sum_{j=1}^{m} K[\boldsymbol{x}_i'\hat{\boldsymbol{\beta}}(\tau_j)/h_n]. \tag{2}$$

Equations 1 and 2 are used to compute the probability of censoring for the CQR model with $c_L \leq y \leq c_H$. Similar to the previous example we get

$$\hat{P}(y = c_L|\boldsymbol{x}_i) = \hat{P}(y^* < c_L|\boldsymbol{x}_i) = m^{-1}\sum_{j=1}^{m} I\{\boldsymbol{x}_i'\hat{\boldsymbol{\beta}}(\tau_j) < c_L\}$$

and

$$\hat{P}(y = c_H|\boldsymbol{x_i}) = \hat{P}(y^* > c_H|\boldsymbol{x_i}) = m^{-1}\sum_{j=1}^{m} I\{\boldsymbol{x}_i'\hat{\boldsymbol{\beta}}(\tau_j) > c_H\},$$

where $\hat{\boldsymbol{\beta}}(\tau)$ is the CQR coefficient estimate. The smoothed versions replace $I(\cdot)$ by $K(\cdot)$.

2.5. Partial effects on the probability

Partial effects on the probability can be obtained from Equation 2. Assume that the set of regressors is formed of S continuous variables X and R dummy variables in D. Then, Equation 2 becomes

$$\hat{P}(y = 1|\boldsymbol{x}, \boldsymbol{d}) = m^{-1}\sum_{j=1}^{m} K\left(\frac{\boldsymbol{x}'\hat{\boldsymbol{\beta}}(\tau_j) + \boldsymbol{d}'\hat{\boldsymbol{\gamma}}(\tau_j)}{h_n}\right).$$

Thus partial effects can be calculated as follows:

$$\frac{\partial\hat{P}(y = 1|\boldsymbol{x}, \boldsymbol{d})}{\partial x_s} = m^{-1}\sum_{j=1}^{m}\frac{1}{h_n}K'\left(\frac{\boldsymbol{x}'\hat{\boldsymbol{\beta}}(\tau_j) + \boldsymbol{d}'\hat{\boldsymbol{\gamma}}(\tau_j)}{h_n}\right)\hat{\boldsymbol{\beta}}_{\boldsymbol{s}}(\tau_j) \tag{3}$$

for $s = 1, \ldots, S$ continuous covariates and

$$\Delta_r \hat{P}(y = 1|\boldsymbol{x}, \boldsymbol{d}) = m^{-1} \sum_{j=1}^{m} K \left(\frac{\boldsymbol{x}'\hat{\boldsymbol{\beta}}(\tau_j) + \boldsymbol{d}'_{-r}\hat{\boldsymbol{\gamma}}_{-\boldsymbol{r}}(\tau_j) + \hat{\boldsymbol{\gamma}}_{\boldsymbol{r}}(\tau_j)}{h_n} \right) - m^{-1} \sum_{j=1}^{m} K \left(\frac{\boldsymbol{x}'\hat{\boldsymbol{\beta}}(\tau_j) + \boldsymbol{d}'_{-r}\hat{\boldsymbol{\gamma}}_{-\boldsymbol{r}}(\tau_j)}{h_n} \right) \quad (4)$$

for binary dummy covariates $r = 1, \ldots, R$, and where the subindex $-r$ indicates that the binary regressor r has been excluded.

Then, following the literature we can define the average partial effect (APE)

$$\text{APE}_{x_s} = n^{-1} \sum_{i=1}^{n} \frac{\partial \hat{P}(y = 1|\boldsymbol{x}_i, \boldsymbol{d}_i)}{\partial x_s},$$

$$\text{APE}_{z_r} = n^{-1} \sum_{i=1}^{n} \Delta_r \hat{P}(y = 1|\boldsymbol{x}_i, \boldsymbol{d}_i),$$

and the partial effect at means (PEAM)

$$\text{PEAM}_{x_s} = \frac{\partial \hat{P}(y = 1|\bar{\boldsymbol{x}}, \mathbf{0})}{\partial x_s},$$

$$\text{PEAM}_{z_r} = \Delta_r \hat{P}(y = 1|\bar{\boldsymbol{x}}, \mathbf{0}),$$

where $\bar{x} = n^{-1} \sum_{i=1}^{n} x_i$ and all binary variables are evaluated (by default) at their base value.

Note that by the law of iterated expectations, the APE measures the unconditional effect on the probability of $y = 1$ because we are averaging all the partial effects for each value of the covariates, while in the case of the PEAM it is the change in the conditional probability for a subject with "average characteristics". These are the two most popular ways to report marginal effects for probability models.

2.6. Inference for the estimators

Inference for QR models depends on the estimation of the density of the conditional errors. A common implementation is based on bootstrap resampling, which is an available option for the `qreg` command in Stata, more importantly for the generalizations such as `sqreg`.

Bootstrap resampling for quantile regression is analyzed in Hahn (1995). Buchinsky (1995) study favors bootstrap over other estimation alternatives.

3. The `ldvqreg` syntax

In this section we present the software contribution of the paper. We will use the statistical package Stata, in particular the command-driven mode. Unlike other statistical packages whose routines are programmed by the users in the form of functions, Stata assigns these types of operations in commands. We have programmed the `ldvqreg` command whose functionality is described throughout this section.

3.1. Syntax

The command syntax is:

`ldvqreg` *depvar indepvars* [*if*] [*in*], [`quantile`(#[#[#...]]) `ll`(real) `ul`(real) `reps`(string) `qcen`(string) `pcen`(string) `p1`(string) `pbc` `margins`(type) `xbinary`(varlist) `bwidth`(real) `kernel`(string) `pbwidth`(real)]

where square brackets distinguish optional qualifiers and options from required ones. In this syntax, *depvar* denote the name of the DPV in the data set and *indepvars* denotes a list of covariable names. The *if* and *in* qualifiers are useful to restrict the execution of the command to a subset of rows in the data set.

3.2. Options

`ldvqreg` supports the following options:

- *General:*

  `quantile(#)` estimates # quantile; default is `quantile(50)`.

  `reps(#)` performs # bootstrap replications; default is `reps(50)`.

- *Censoring:*

  `ll(#)` left-censoring limit.

  `ul(#)` right-censoring limit.

  `qcen(newvar)` stores predicted censored quantiles in `newvar_q#`.

  `pcen(newvar)` stores censorship probability in `newvar` and `newvar_s` (smoothed).

- *Binary data:*

  `p1(newvar)` stores probability of *depvar* = 1 in `newvar` and `newvar_s` (smoothed).

  `pbc` indicates that the predicted probability in `p1(newvar)` should be computed with the bias-corrected coefficients.

  `margins(string)` indicates that the partial effects indicated in `type` should be displayed. The options are: `ape` (average partial effect), `peam` (partial effect at means) or `both` (display APE and PEAM).

  `xbinary(varlist)` indicates that the covariates listed in `varlist` are binary. This is necessary to correctly compute the partial effects requested in the option `margins(string)`.

- *Smoothing:*

  `bwidth(real)` specifies the bandwidth to smooth the target function (the default is described in Section 2.3).

  `kernel(string)` specify kernel function; the options are: `gaussian` (the default), `logit`, `epanechnikov` or `biweight`.

  `pbwidth(real)` specifies the bandwidth to smooth the predicted probabilities.

If `ll(#)` and `ul(#)` are not specified and the dependent variable is a dummy variable (which is automatically checked), then the command runs a binary QR. If `ll(#)` and `ul(#)` are not specified but the dependent variable is not a dummy, then the command runs a smoothed QR model.

3.3. Saved results

`ldvqreg` stores the following results in `e()`.

- *Scalars:*

  `e(N)` number of observations.

  `e(reps)` number of replications.

  `e(bwidth)` bandwidth.

- *Macros:*

  `e(title)` Censored or Binary model.

  `e(vcetype)` title used to label Std. Err.

  `e(kernel)` name of kernel.

  `e(properties)` b V.

  `e(depvar)` name of dependent variable.

- *Matrices:*

  `e(b)` coefficient vector.

  `e(V)` bootstrap variance matrix.

  `e(b_bs)` bias-corrected coefficient vector (only for binary model).

- *Functions:*

  `e(sample)` marks estimation sample.

4. Examples

4.1. Simulations

The simulations are based on the so-called location-scale models. Consider the following data generating process (DGP):

$$y_i = \beta_0 + \beta_1 x_i + (\gamma_0 + \gamma_1 x_i)\epsilon_i,\ \epsilon_i \sim iid\ (0,1),$$

where $x > 0$ is a scalar random variable. Here (β_0, β_1) is said to control the location and (γ_0, γ_1) the scale. For these models, the conditional mean model is $E(y|x) = \beta_0 + \beta_1 x$, but for QR models, $Q_\tau(y|x) = (\beta_0 + \gamma_0 Q_\tau(\epsilon)) + (\beta_1 + \gamma_1 Q_\tau(\epsilon))x$. Thus, $\frac{\partial E(y|x)}{\partial x} = \beta_1$ but quantile heterogeneity is obtained by evaluating $\frac{\partial Q_\tau(y|x)}{\partial x} = \beta_1 + \gamma_1 Q_\tau(\epsilon)$.

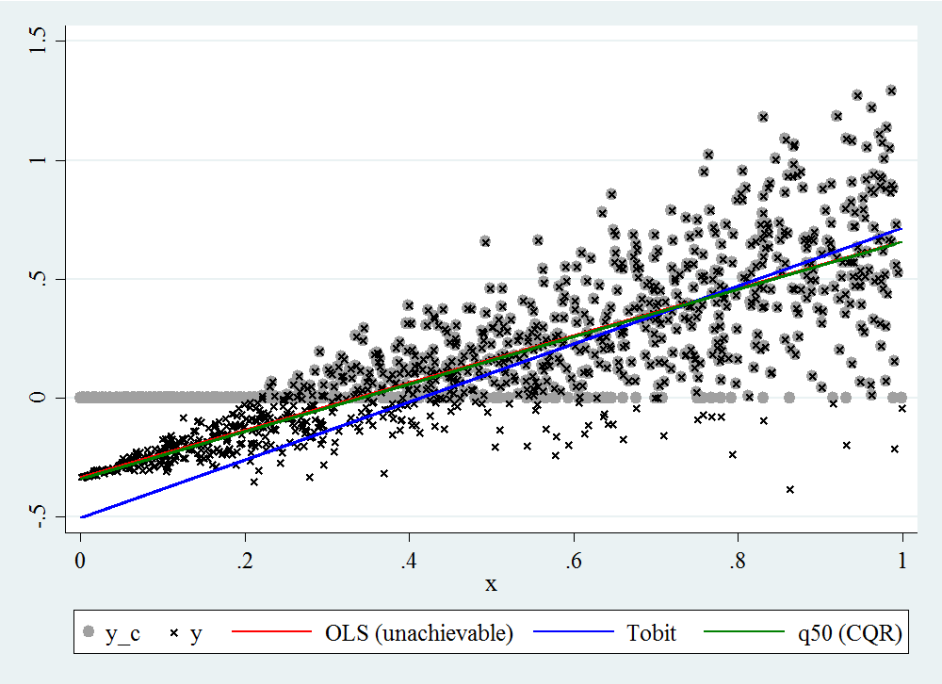

Figure 1: A comparison of tobit and censored quantile regression models. Plot format (colors, font, etc.) depends on the version of Stata.

Censoring

We start with a simple simulation to show that the tobit model can be biased if the homogeneity of the conditional distribution is not satisfied.

```
. set seed 321
. set obs 1000
* number of observations (_N) was 0, now 1,000

. gen x = runiform()
. gen y = -1/3 + x + x * rnormal() / 3
. gen y_c = max(y, 0)
```

In this example the error term is standard Gaussian, but its interaction with `x` determines that it has conditional heteroskedasticity. Figure 1 shows a scatter plot with the latent (unobserved variable) `y` (using points with an x) and the censored variable `y_c` (with grey circles). The graph also has the OLS estimation of the relation between `y` and `x` (which is not feasible as we cannot observe the latent variable), the tobit estimation that controls for lower censoring at 0, which assumes homoskedasticity, and the median regression estimate using the CQR estimate with `ldvqreg` of `y_c` on `x`, which is distribution free (i.e., it does not require to assume homoskedastic Gaussian errors as in the tobit model).

Note that the tobit estimator is clearly different from the true OLS estimate. Nevertheless, the line that corresponds to the CQR estimation is indeed close to the true line. Then, the CQR appears as a useful alternative to the tobit model when the distributional requirements are not satisfied.

Consider now a DGP with two cases (Stata codes omitted). Half the sample has a homoskedastic error structure (`y = x + rnormal()/3`) and the other half has heteroskedastic error structure (`y = x + ( 1 + x )* rnormal()/3`). The former model has that all the QR coefficients are the same across quantiles, i.e., $\beta(\tau) = \beta$, $\forall\tau$, while the second allows for heterogeneity across quantiles. The generated variable `y` has no censoring, and `y_c = min(max(y,0),1)` has lower (0) and upper (1) censoring.

We will first compare the available command `sqreg` with the proposed command `ldvqreg` in the case where there is no censoring, i.e., without specifying `ll(#)` and `ul(#)` (thus the command assumes no censoring). This is done to evaluate if the smoothed implementation works. The results below show that the results are very similar, and thus `ldvqreg` works for the general case.

```
. sqreg y x if heter==0 , q(20 50 80) reps(100)

(fitting base model)

[-OUTPUT OMITTED-]

. estimates store sqr0
. test [q20=q50=q80]: x

( 1)  [q20]x - [q50]x = 0
( 2)  [q20]x - [q80]x = 0

F(  2,  1998) =    1.70
Prob > F =    0.1837

. ldvqreg y x if heter==0 , q(20 50 80) reps(100)

(running cqr_est on estimation sample)

[-OUTPUT OMITTED-]

. estimates store ldv0
. test [q20=q50=q80]: x

( 1)  [q20]x - [q50]x = 0
( 2)  [q20]x - [q80]x = 0

chi2(  2) =    3.23
Prob > chi2 =    0.1992

. sqreg y x if heter==1 , q(20 50 80) reps(100)

(fitting base model)

[-OUTPUT OMITTED-]
```

```
. estimates store sqr1
. test [q20=q50=q80]: x

( 1)  [q20]x - [q50]x = 0
( 2)  [q20]x - [q80]x = 0

F(  2,  1998) =   20.09
Prob > F =    0.0000

. ldvqreg y x if heter==1 , q(20 50 80) reps(100)

(running cqr_est on estimation sample)

[-OUTPUT OMITTED-]

. estimates store ldv1
. test [q20=q50=q80]: x

( 1)  [q20]x - [q50]x = 0
( 2)  [q20]x - [q80]x = 0

chi2(  2) =   84.47
Prob > chi2 =    0.0000

. estimates table sqr0 sqr1 ldv0 ldv1

-------------------------------------------------------------------
Variable     |    sqr0         sqr1         ldv0         ldv1
-------------+-----------------------------------------------------
q20          |
x            |  1.0174711    .77491151    1.0090585    .74898022
_cons        | -.28811283   -.30173431   -.28007639   -.28191605
-------------+-----------------------------------------------------
q50          |
x            |  1.0302213      1.01513    1.0412487    1.0192041
_cons        | -.01484978    .00341644   -.01825961    .00398835
-------------+-----------------------------------------------------
q80          |
x            |  1.0922721    1.3096443    1.0884       1.3203196
_cons        |  .24420286    .28846714    .24572663    .26465872
-------------------------------------------------------------------
```

Second, we show how to estimate the quantiles with the censored variable using the `ldvqreg` command. For this we must use the options `ll(0)` and `ul(1)` to indicate the lower (0) and upper (1) censoring points that correspond to this case. Note that both results are similar to the `sqreg` command with `y` as dependent variable (i.e., no censoring). Therefore, the `ldvqreg`

command allows us to retrieve some of the information about the latent variable distribution. An interesting computational aspect is that the command takes around 60 seconds to run for the homoskedastic case and about 83 seconds in the heterosckdastic case, therefore it has a good performance in terms of its computing speed.[1]

```
. ldvqreg y_c x if heter==0 , q(20 50 80) reps(100) ll(0) ul(1)

(running cqr_est on estimation sample)

[-OUTPUT OMITTED-]

. estimates store ldv0c
. test [q20=q50=q80]: x

( 1)  [q20]x - [q50]x = 0
( 2)  [q20]x - [q80]x = 0

chi2(  2) =    2.10
Prob > chi2 =    0.3501

. ldvqreg y_c x if heter==1 , q(20 50 80) reps(100) ll(0) ul(1)

(running cqr_est on estimation sample)

[-OUTPUT OMITTED-]

. estimates store ldv1c
. test [q20=q50=q80]: x

( 1)  [q20]x - [q50]x = 0
( 2)  [q20]x - [q80]x = 0

chi2(  2) =   32.67
Prob > chi2 =    0.0000

. estimates table ldv0c ldv1c

----------------------------------------
Variable     |   ldv0c        ldv1c
-------------+--------------------------
q20          |
x            |  .95401311    .56308125
_cons        | -.23851249   -.15169441
-------------+--------------------------
```

[1]This calculation includes all 100 bootstrap samples and was measured using Stata 16 MP (64-bit) and Windows 7 operating system (8 GB of RAM and Intel Core i7-3770 processor @ 3.40GHz).

```
q50          |
x            |   1.002916    .97882127
_cons        |  .00109231    .01848998
-------------+----------------------------
q80          |
x            |  1.0908209    1.3702204
_cons        |  .24546395    .25605242
------------------------------------------
```

Third, we show that ignoring censoring in the estimation of conditional quantiles introduces a bias by comparing the results of the `sqreg` and `ldvqreg` commands using the censored dependent variable.[2] Since we only compare point estimates, we run only a few replicates of the bootstrap.

```
. sqreg y_c x , reps(5) q(20 50 80)

(fitting base model)

[-OUTPUT OMITTED-]

. estimates store sqr_c
. ldvqreg y_c x , reps(5) q(20 50 80) ll(0) ul(1)

(running cqr_est on estimation sample)

[-OUTPUT OMITTED-]

. estimates store ldv_c
. estimates table sqr_c ldv_c

---------------------------------------
Variable     |   sqr_c        ldv_c
-------------+-------------------------
q20          |
x            |   .56201746    .8210617
_cons        |  -.06556587   -.22783645
-------------+-------------------------
q50          |
x            |  1.003579     1.0001938
_cons        |   .00202664    .00577996
-------------+-------------------------
q80          |
x            |  .72443049    1.1329151
_cons        |  .39864097     .26838409
---------------------------------------
```

[2] This is a generalization of the bias that occurs for censoring in the mean-based model, which can be studied by the tobit estimator and its comparison to standard OLS.

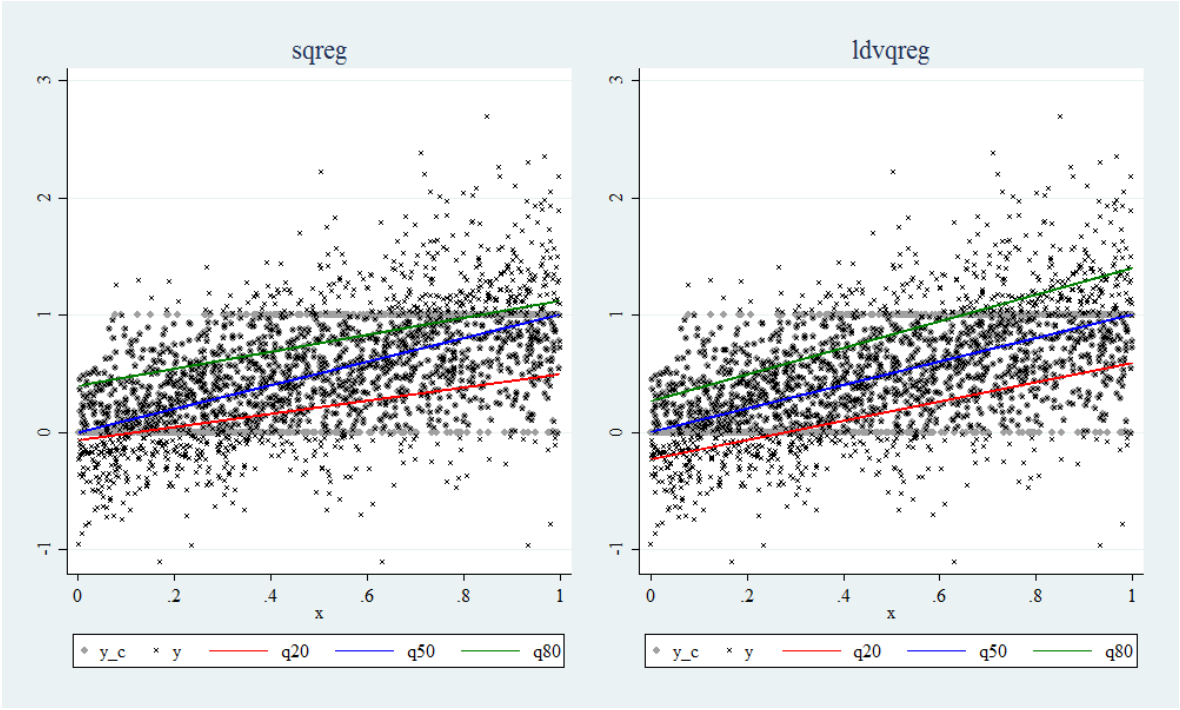

Figure 2: Comparing quantile commands under censorship. Plot formats (colors, font, etc.) depend on the version of Stata.

Note that the coefficients computed by both commands are different. The gap between the two represents the bias for ignoring the censorship process. Figure 2 shows the three lines estimated by both commands. The green line correspond to the $\tau = 0.9$ case, the blue line to the $\tau = 0.5$ one, and finally the red one to $\tau = 0.2$. The points with the symbol "`x`" represent the realizations of the latent (uncensored) variable while the gray circles are those of the observed variable (censored). Clearly, the `sqreg` command underestimates the coefficients of the extreme quantiles as a consequence of the simulated upper and lower censoring while those estimated by the `ldvqreg` command are consistent with the scatter of the latent variable.

The command `ldvqreg` can be used to predict censored quantiles and also to compute the probability of censorship by the options `qcen()` and `pcen()`, respectively. Quantile prediction can be done in an individual way for each τ, but the probability of censorship requires many τs (at least 2). We show now an example code and a graph in Figure 3 with the predicted censured quantiles (left panel) and probability of censoring (right panel).

```
. ldvqreg y_c x , reps(2) q(10 20 30 40 50 60 70 80 90) ///
ll(0) ul(1) qcen(myqcen) pcen(mypcen)

(running cqr_est on estimation sample)

[-OUTPUT OMITTED-]

. summarize
```

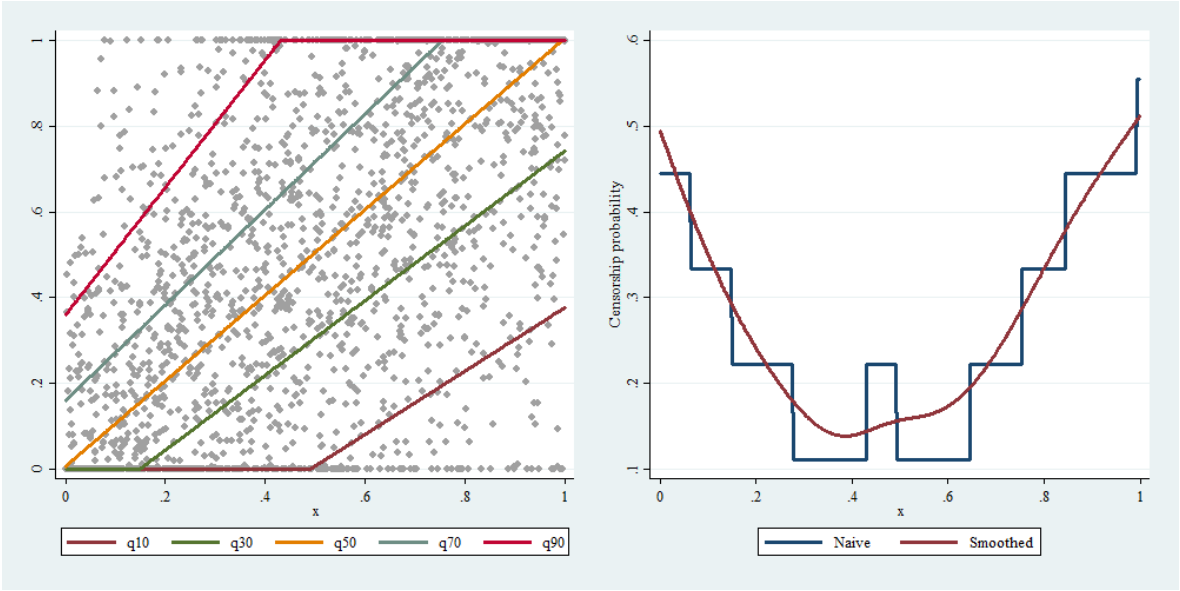

Figure 3: Prediction of quantiles and censorship probability.
Note: plot formats (colors, font, etc.) depends on the version of Stata.

```
Variable     |        Obs        Mean    Std. Dev.       Min        Max
-------------+---------------------------------------------------------
heter         |      4,000          .5    .5000625         0          1
x             |      4,000    .4973596    .2883611      .000018   .9997839
y             |      4,000    .5076568    .5295976    -1.163579   2.778667
y_c           |      4,000    .4922552    .3645911         0          1
mypcen        |      4,000     .25675     .125413    .1111111   .5555556
-------------+---------------------------------------------------------
mypcen_s      |      4,000    .2706427    .1154597    .1384926   .5122733
myqcen_q10    |      4,000    .0946118    .1214869         0    .3766204
myqcen_q20    |      4,000    .2127563    .1962473         0    .5930477
myqcen_q30    |      4,000    .3135726    .2376881         0    .7425613
myqcen_q40    |      4,000    .4120319    .2696362         0    .8849297
-------------+---------------------------------------------------------
myqcen_q50    |      4,000    .5032121    .2883758     .005798          1
myqcen_q60    |      4,000    .5963741    .3047473    .0502407          1
myqcen_q70    |      4,000    .6813506    .2779192    .1597887          1
myqcen_q80    |      4,000    .7620734    .2443681    .2682098          1
myqcen_q90    |      4,000    .8610955    .2007863    .3594372          1
```

It should be noted that in the `summarize` output there are new variables that were generated with the names used in the `ldvqreg` run. On one hand, the variables `mypcen` and `mypcen_s` have the probability with the naïve and the smoothed formulas, respectively. On the other hand, the varlist `myqcen_q10, myqcen_q20,..., myqcen_q90` are the predicted censored quantiles for $\tau \in \{0.10, 0.20, \ldots, 0.90\}$.

Finally, we show here the effect of changing the bandwidth and the kernel function to smooth

the objective function. This corresponds to the options `ker()` and `bw()`, respectively. Given that the main objective is to compare the point estimate we only use a few bootstrap replications.

```
** Kernel changes
. ldvqreg y_c x , q(20 50 80) reps(10) ll(0) ul(1)

(running cqr_est on estimation sample)

[-OUTPUT OMITTED-]

* Kernel: Gaussian
. estimates store cqr_k1
. ldvqreg y_c x , q(20 50 80) reps(10) ll(0) ul(1) ker(logit)

(running cqr_est on estimation sample)

[-OUTPUT OMITTED-]

* Kernel: Logistic
. estimates store cqr_k2
. ldvqreg y_c x , q(20 50 80) reps(10) ll(0) ul(1) ker(epane)

(running cqr_est on estimation sample)

[-OUTPUT OMITTED-]

* Kernel: Epanechnikov
. estimates store cqr_k3
. ldvqreg y_c x , q(20 50 80) reps(10) ll(0) ul(1) ker(biwei)

(running cqr_est on estimation sample)

[-OUTPUT OMITTED-]

* Kernel: Biweight
. estimates store cqr_k4
. estimates table cqr_k*

-------------------------------------------------------------------
Variable     |   cqr_k1       cqr_k2       cqr_k3       cqr_k4
-------------+-----------------------------------------------------
q20          |
x            |  .82106138    .80669786    .82038709    .83939056
_cons        | -.22783621   -.20616172   -.22430489   -.25006252
-------------+-----------------------------------------------------
```

```
q50          |
x            |  1.0001938    .96858031     1.000034    1.0128315
_cons        |  .00577996    .02043529    .00583736   -.00032702
-------------+------------------------------------------------------------
q80          |
x            |  1.1336573    1.1612341    1.1327198    1.1577665
_cons        |  .26818938    .24645125    .26883745    .26697925
------------------------------------------------------------------------------

** Bandwidth changes
. ldvqreg y_c x , q(20 50 80) reps(10) ll(0) ul(1)

(running cqr_est on estimation sample)

[-OUTPUT OMITTED-]

. estimates store cqr_h0
. loc h1 = e(bwidth)/2
. loc h2 = e(bwidth)*2
. ldvqreg y_c x , q(20 50 80) reps(10) ll(0) ul(1) bw(`h1')

(running cqr_est on estimation sample)

[-OUTPUT OMITTED-]

. estimates store cqr_h1
. ldvqreg y_c x , q(20 50 80) reps(10) ll(0) ul(1) bw(`h2')

(running cqr_est on estimation sample)

[-OUTPUT OMITTED-]

. estimates store cqr_h2
. estimates table cqr_h*

----------------------------------------------------
Variable     |   cqr_h0       cqr_h1       cqr_h2
-------------+--------------------------------------
q20          |
x            |  .82106138    .83486195    .79619464
_cons        | -.22783621   -.24567505   -.19608774
-------------+--------------------------------------
q50          |
x            |  1.0001938    1.0119603    .95743738
_cons        |  .00577996    .00044421    .02622289
-------------+--------------------------------------
```

```
q80           |
x             |  1.1336573    1.1419295    1.1844295
_cons         |  .26818938    .26947256    .23250225
------------------------------------------------------

. drop _est*
```

Binary dependent variable

Consider now a binary dependent variable case. The DGP for this case is the following.

```
. drop _all
. set seed 321
. set obs 2000
* number of observations (_N) was 0, now 2,000

. gen x = runiform()*10
. gen y = -2.5 + x + x*(rchi2(1)-1)/sqrt(2)
. gen y_b = (y>0)

. ta y_b

y_b         |      Freq.     Percent        Cum.
------------+-----------------------------------
0           |        883       44.15       44.15
1           |      1,117       55.85      100.00
------------+-----------------------------------
Total       |      2,000      100.00
```

We first compute a `probit` model, then a QR model with the (unobserved) latent variable, and finally the proposed BQR using the developed command. For the last two we consider the median estimate, i.e., $\tau = 0.5$. In order to compare these two we normalize the coefficients of the median QR regression to $\|\boldsymbol{b}\| = 1$. The binary regression model already has this normalization.

```
. quietly probit y_b x
. nlcom (_b[x]/sqrt(_b[x]^2+_b[_cons]^2)) ///
        (_b[_cons]/sqrt(_b[x]^2+_b[_cons]^2)), post

[-OUTPUT OMITTED-]

. estimates store mle_n
. quietly bsqreg y x
. nlcom (_b[x]/sqrt(_b[x]^2+_b[_cons]^2)) ///
        (_b[_cons]/sqrt(_b[x]^2+_b[_cons]^2)), post

[-OUTPUT OMITTED-]
```

```
. estimates store qr_n
. quietly ldvqreg y_b x
. nlcom (_b[x]) (_b[_cons]), post

[-OUTPUT OMITTED-]

. estimates store ldv_b
. estimates table mle_n qr_n ldv_b

-------------------------------------------------
Variable     |   mle_n         qr_n        ldv_b
-------------+-----------------------------------
_nl_1        |   .2230184    .23005033    .23819335
_nl_2        | -.97481424   -.97317873   -.97122184
-------------------------------------------------

. drop _est*
```

Note that in this case the BQR model (column `ldv_b`) show similar results to the QR model with the latent variable (column `qr_n`), but not to the probit model (column `mle_n`). In fact, the probit model should differ from the median estimates given that we are using an asymmetric DGP.

Consider now the comparison of the `ldvqreg` with the standard QR alternatives in `Stata`. In this case we compare it with `bsqreg` (QR with standard errors computed by bootstrap), separately for different quantiles.

```
. * Compare with sqreg
. quietly bsqreg y_b x, q(20)
. nlcom (_b[x]/sqrt(_b[x]^2+_b[_cons]^2)) ///
        (_b[_cons]/sqrt(_b[x]^2+_b[_cons]^2)), post

[-OUTPUT OMITTED-]

. estimates store qr20_b
. quietly ldvqreg y_b x, q(20)
. nlcom (_b[x]) (_b[_cons]), post

[-OUTPUT OMITTED-]

. estimates store ldv20_b
. quietly bsqreg y_b x, q(50)
. nlcom (_b[x]/sqrt(_b[x]^2+_b[_cons]^2)) ///
        (_b[_cons]/sqrt(_b[x]^2+_b[_cons]^2)), post

[-OUTPUT OMITTED-]
```

```
. estimates store qr50_b
. quietly ldvqreg y_b x, q(50)
. nlcom (_b[x]) (_b[_cons]), post

[-OUTPUT OMITTED-]

. estimates store ldv50_b
. estimates table qr20_b qr50_b ldv20_b ldv50_b

-------------------------------------------------------------------
Variable     |   qr20_b       qr50_b      ldv20_b      ldv50_b
-------------+-----------------------------------------------------
_nl_1        |  .33113567     .80311667    .14108016    .23819335
_nl_2        | -.94358315    -.5958218   -.99000605   -.97122184
-------------------------------------------------------------------

. bsqreg y_b x , q(80)

(fitting base model)
convergence not achieved.
convergence not achieved
r(430);
```

Note that the results are very different across estimators. This applies even if we normalize the `bsqreg` coefficients to $\|\boldsymbol{b}\| = 1$. In fact, the QR estimates show convergence problems in several bootstrap simulations (denoted by an `x` in the output), while the `ldvqreg` runs smoothly. Overall this shows that BQR should be implemented with the proposed smoothed version.

Finally, we implement tests of homogeneity and symmetry, comparing the (true) latent variable model with the binary regression case. To evaluate the symmetry of the conditional distribution we use the procedure suggested by Koenker (2005) evaluating the following linear null hypothesis:

$$\mathrm{H}_0 : \frac{1}{2} \cdot \boldsymbol{\beta}\left(\frac{1}{2} - \delta\right) + \frac{1}{2} \cdot \boldsymbol{\beta}\left(\frac{1}{2} + \delta\right) - \boldsymbol{\beta}\left(\frac{1}{2}\right) = \mathbf{0}$$

for some $\delta \in (0, \frac{1}{2})$. This can be easily implemented by a Wald test using the `test` command.

```
* With unobservable data (uncensored)
. sqreg y x , q(10 25 50 75 90) reps(300)

(fitting base model)

[-OUTPUT OMITTED-]

. * Homogeneity
. test [q10=q25=q50=q75=q90]: x
```

```
( 1)  [q10]x - [q25]x = 0
( 2)  [q10]x - [q50]x = 0
( 3)  [q10]x - [q75]x = 0
( 4)  [q10]x - [q90]x = 0

F(  4,  1998) =  118.08
Prob > F =    0.0000

* Symmetry
. test (([q10]x+[q25]x+[q75]x+[q90]x)/4-[q50]x=0) ///
       (([q10]_cons+[q25]_cons+[q75]_cons+[q90]_cons)/4-[q50]_cons = 0)

( 1)  .25*[q10]x + .25*[q25]x - [q50]x + .25*[q75]x + .25*[q90]x = 0
( 2)  .25*[q10]_cons + .25*[q25]_cons - [q50]_cons + .25*[q75]_cons +
>     .25*[q90]_cons = 0

F(  2,  1998) =  176.51
Prob > F =    0.0000

* With observable data (censored)
. ldvqreg y_b x , q(10 25 50 75 90) reps(300)

(running bqr_est on estimation sample)

[-OUTPUT OMITTED-]

* Homogeneity
. test [q10=q25=q50=q75=q90]: x

( 1)  [q10]x - [q25]x = 0
( 2)  [q10]x - [q50]x = 0
( 3)  [q10]x - [q75]x = 0
( 4)  [q10]x - [q90]x = 0

chi2(  4) =  365.80
Prob > chi2 =    0.0000

* Symmetry
. test (([q10]x+[q25]x+[q75]x+[q90]x)/4-[q50]x=0) ///
       (([q10]_cons+[q25]_cons+[q75]_cons+[q90]_cons)/4-[q50]_cons = 0)

( 1)  .25*[q10]x + .25*[q25]x - [q50]x + .25*[q75]x + .25*[q90]x = 0
( 2)  .25*[q10]_cons + .25*[q25]_cons - [q50]_cons + .25*[q75]_cons +
>     .25*[q90]_cons = 0

chi2(  2) =   47.46
Prob > chi2 =    0.0000
```

The results are as expected. We reject the hypotheses of both homoskedasticity and symmetry of the latent variable. Both features should indicate that the assumptions of the probit and logit models are not valid ad we should consider the semi-parametric approach given by `ldvqreg`. Since this example uses boostrap sampling, it is interesting to evaluate the computation time: the `ldvqreg` command takes around 107 seconds to obtain the point estimate and the standard errors, which is quite reasonable compared to the 35 seconds it takes to do the same inference exercises but with the true latent variable (something that is impossible in practice).[3]

Finally, the `ldvqreg` also computes the conditional probabilities of $y = 1$ using the estimated coefficients for a grid of τs by the option `p1()`. We show here an example coding:

```
. probit y_b x

[-OUTPUT OMITTED-]

. predict p_pro

(option pr assumed; Pr(y_b))

. ldvqreg y_b x , reps(2) q(10 20 30 40 50 60 70 80 90) ll(0) ul(1) p1(p_bqr)

(running bqr_est on estimation sample)

[-OUTPUT OMITTED-]

. summarize

Variable     |        Obs        Mean    Std. Dev.       Min        Max
-------------+---------------------------------------------------------
x            |      2,000      4.9855    2.852372   .0001804    9.99784
y            |      2,000    2.408413    6.126037  -2.499892   60.79322
y_b          |      2,000       .5585    .4966901          0          1
y_n          |      2,000    .9553311     2.42998  -.9916178   24.11449
p_pro        |      2,000    .5578556    .3137343   .0558153   .9797236
-------------+---------------------------------------------------------
p_bqr        |      2,000    .5653889    .3212985          0          1
p_bqr_s      |      2,000    .5661044    .3177954   2.72e-13   .9842031
```

Note that new variables appear, that is, `p_bqr` and `p_bqr_s` generated by `ldvqreg` and the variable `p_pro` generated by `probit`. Figure 4 shows these three predicted probabilities of $y = 1|\boldsymbol{x}$. It should be noted that the probit model underestimates the probabilities for the center values and overestimates in the extremes, in particular for low `x`. This is an expected result because of the heterogeneity and asymmetries in the DGP.

[3]This calculation was measured using Stata 16 MP (64-bit) and Windows 7 operating system (8 GB of RAM and Intel Core i7-3770 processor @ 3.40GHz).

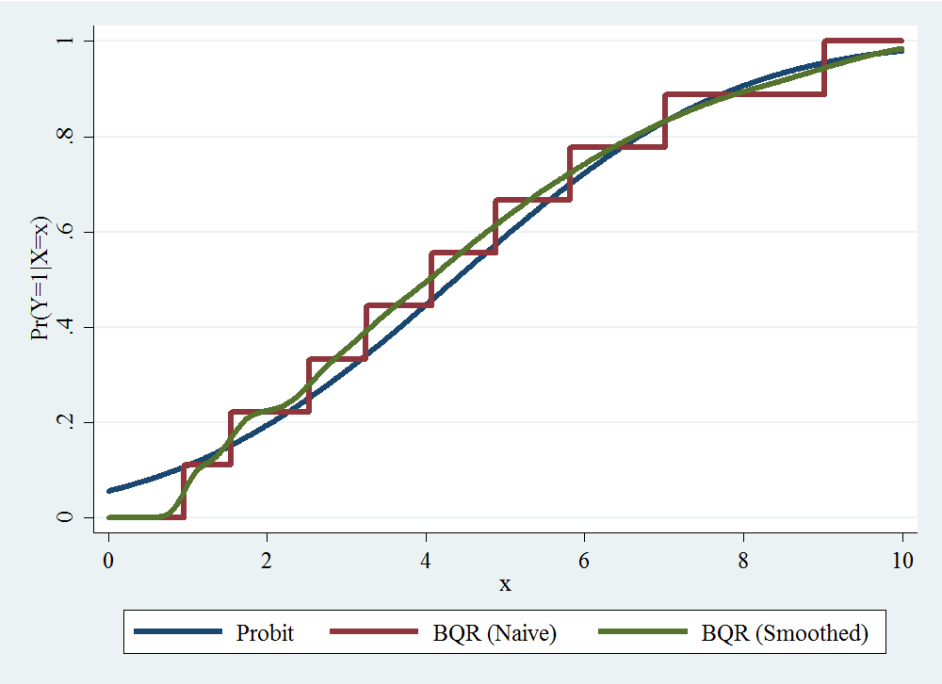

Figure 4: Comparison of predicted probabilities. Plot formats (colors, font, etc.) depend on the version of Stata.

4.2. Real data

Double-censored example

We provide here an empirical illustration of a double-censoring example. Wichert and Wilke (2008) offer an empirical application where the wages are double censored (because of non-zero and a social security ceiling). They use a sample extracted from the IAB-Employment Sample 1975-2001 of the Institute for Employment Research (IAB). For this example the dependent variable is `wage` and the censoring values are 0 and 200 for the lower and upper censoring, respectively. In this case, the censorship is generated by the type of information available in the administrative data.

We thus implement a simple wage regression model for $\tau = 0.20, 0.50, 0.80$, and we compare their results with the double-censored `tobit` command. Table 1 reports the results of estimating the salary equation in the mean (tobit) and in the different quantiles (CQR). For the particular case of censored QR we implement inference of the symmetry across quantiles using the same testing procedure as in the simulations section.

```
. test [q20=q50=q80]

( 1)  [q20]age - [q50]age = 0
( 2)  [q20]female - [q50]female = 0
( 3)  [q20]age - [q80]age = 0
( 4)  [q20]female - [q80]female = 0
```

		Censored QR		
	Tobit	q20	q50	q80
Age	0.568**	0.414**	0.333**	0.846**
	(0.0364)	(0.0488)	(0.0340)	(0.0470)
Female	−24.85**	−31.94**	−20.06**	−19.43**
	(0.510)	(0.905)	(0.466)	(0.582)
Constant	48.78**	36.11**	56.38**	60.53**
	(1.334)	(1.732)	(1.178)	(1.631)
Observations	21,685	21,685	21,685	21,685

Table 1: Wage equations (standard errors in parentheses); * indicates significance at 5% and ** at 1%.

```
chi2(  4) =  341.55
Prob > chi2 =    0.0000
```

```
. test (0.5*_b[q20:age]+0.5*_b[q80:age]=_b[q50:age]) ///
>          (0.5*_b[q20:female]+0.5*_b[q80:female]=_b[q50:female]) ///
>          (0.5*_b[q20:_co]+0.5*_b[q80:_co]=_b[q50:_co])

( 1)  .5*[q20]age - [q50]age + .5*[q80]age = 0
( 2)  .5*[q20]female - [q50]female + .5*[q80]female = 0
( 3)  .5*[q20]_cons - [q50]_cons + .5*[q80]_cons = 0

chi2(  3) =  238.99
Prob > chi2 =    0.0000
```

For both cases, we reject the null of symmetry across quantiles.

In general, both the tobit and CQR model coefficients correspond to the effect of a covariate on the latent variable and therefore not always has an intuitive interpretation. In this particular case, given that the censorship occurs on the salary for administrative reasons, the coefficients measure the wage premium of age and the gender wage gap. For example, women who are in the 20th quantile earn 32 dollars/hour less than men in the same ranking position of the wages and the same age. This gap but for the 80th quantile is just over 19 dollars/hour. Thus, in this example, the greatest gender discrimination on wages occurs among the most disadvantaged workers.

Labor supply models

In this section we show an example of the `ldvqreg` command applied to the study of the probability of having a job in Uruguay, 2015. The data comes from the 2015 Continuous Household Survey (ECH for its acronym in Spanish) prepared by the National Institute of Statistics (INE) and the sample consists of women between 18 and 45 years old who are in the labor force and living in urban areas of Montevideo. Binary probit/logit models are widely

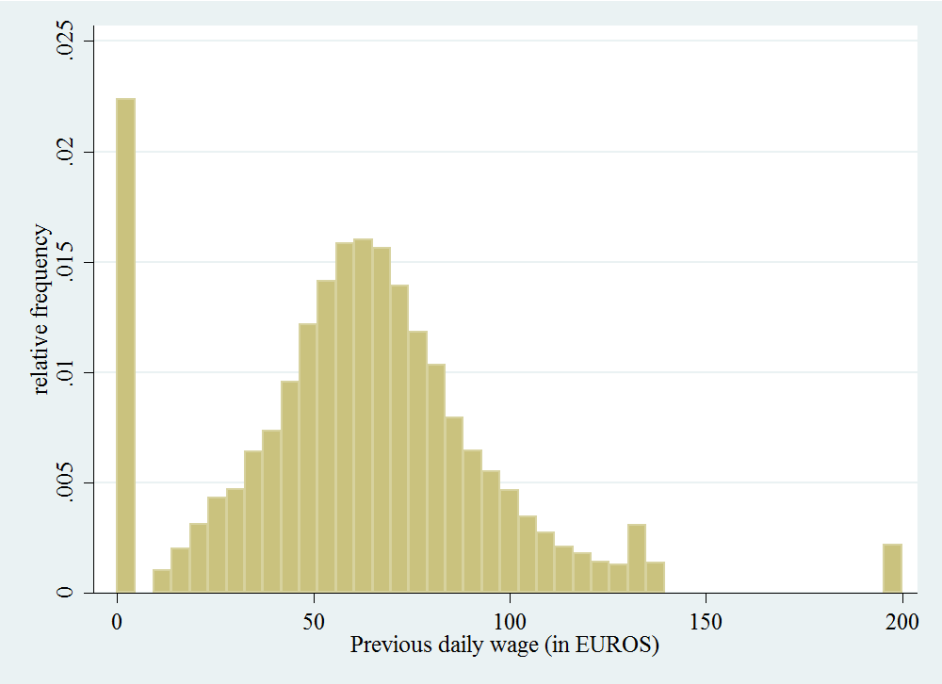

Figure 5: Doubly censored dependent variable (Wichert and Wilke 2008). Plot format (colors, font, etc.) depends on the version of Stata.

used to study the probability of having a job. The `ldvqreg` command is a flexible alternative that allows evaluating some key assumptions of the mentioned maximum likelihood models such as homoskedasticity and symmetry of the conditional distribution.

We analyze the binary regression model using the variable `work` as a dependent variable, which is a dummy variable that indicates with 1 if the woman is working and 0 otherwise. As in the previous example with simulated data, we start by showing the results of the probit model where we normalize the coefficients in such a way that $\|\boldsymbol{b}\| = 1$ and therefore they are comparable with those of the output of the `ldvqreg` command. Again, we estimate the parameters of the conditional quantiles 0.20, 0.50 and 0.80 of the latent variable. For simplicity in exposition, we omit some parts of the code. The main results are in Table 2.

```
. use labordatauy, clear
. describe

Contains data from labordatauy.dta
obs:         9,601
vars:            8                          19 Sep 2021 10:01
size:      307,232
-------------------------------------------------------------------------------------------
storage   display    value
variable name   type    format     label      variable label
-------------------------------------------------------------------------------------------
work            float   %9.0g                 = 1 if it works, = 0 otherwise.
hours           float   %9.0g                 weekly working hours
age             float   %9.0g                 age
educ            float   %9.0g                 years of education
```

```
married         float   %9.0g                 = 1 if married, = 0 otherwise
children        float   %9.0g                 number of children under 6 years of age
couplewrk       float   %9.0g                 = 1 if the couple works, = 0 otherwise
head            float   %9.0g                 = 1 if head of household, = 0 otherwise.
-------------------------------------------------------------------------------------------
```

In this case it is not possible to reject the hypotheses of symmetry at 1% significance level, however, the model does not seem to support the homoskedasticity hypothesis. Thus, a standard model like the probit should not work well.

```
. test [q20=q50=q80]

[-OUTPUT OMITTED-]

chi2( 12) =   41.77
Prob > chi2 =    0.0000

. test (0.5*_b[q20:age]+0.5*_b[q80:age]=_b[q50:age]) ///
>          (0.5*_b[q20:edu]+0.5*_b[q80:edu]=_b[q50:edu]) ///
>          (0.5*_b[q20:mar]+0.5*_b[q80:mar]=_b[q50:mar]) ///
>          (0.5*_b[q20:chi]+0.5*_b[q80:chi]=_b[q50:chi]) ///
>          (0.5*_b[q20:cou]+0.5*_b[q80:cou]=_b[q50:cou]) ///
>          (0.5*_b[q20:hea]+0.5*_b[q80:hea]=_b[q50:hea]) ///
>          (0.5*_b[q20:_co]+0.5*_b[q80:_co]=_b[q50:_co])

[-OUTPUT OMITTED-]

chi2(  7) =    4.02
Prob > chi2 =    0.7771
```

In the case of not rejecting any null hypothesis this might seem that the `ldvqreg` command is somewhat innocuous. However, it should be noted that it serves either as an empirical way of validating or refuting (and possibly replacing) the estimates of a probit/logit model.

In practice, the regression coefficients from models with binary dependent variables do not have a direct quantitative interpretation because they are a normalized version of a latent variable model. However, a qualitative interpretation can be made for probit model coefficients: a positive (negative) sign indicates that the partial effect of the covariate is to increase (decrease) the probability of working. For BQR, the sign indicates the direction of the partial effect on the quantile of the latent variable, but it is not clear how this affects the conditional probability given that the covariate also affects the rest of the conditional quantiles.

It is common in this literature to estimate the partial effects on the probability to make quantitative interpretations. For this purpose we compute the APE and PEAM of the probit and the BQR estimators. Table 3 summarizes the main results. Note that for this exercise we estimated a grid with a finer mesh to achieve better accuracy in the estimation by computing nine quantiles (0.10, 0.20, . . ., 0.90), unlike for the homoskedasticity and symmetry hypothesis tests where we used only three quantiles (0.25, 0.50, 0.75). To get an idea of the trade-off

		Smoothed binary QR		
	Probit	Q20	Q50	Q80
Age	0.0370**	0.0292**	0.0257	0.0713**
	(0.00247)	(0.00353)	(0.0218)	(0.0221)
Years of education	0.0763**	0.0323**	0.172	0.0735
	(0.00504)	(0.00790)	(0.119)	(0.0850)
Children under 6 yrs	−0.0736*	−0.0749*	0.117	−0.0568
	(0.0295)	(0.0406)	(0.160)	(0.108)
Married	0.213**	0.205**	0.478*	0.187
	(0.0454)	(0.0486)	(0.209)	(0.128)
Couple working	−0.0473	−0.122**	−0.199	−0.0403
	(0.0430)	(0.0460)	(0.288)	(0.211)
Household head	0.190**	0.149**	0.262	0.171**
	(0.0404)	(0.0550)	(0.164)	(0.0614)
Constant	−0.951**	−0.956**	−0.787**	−0.959**
	(0.0921)	(0.0210)	(0.164)	(0.122)
Observations	9,601	9,601	9,601	9,601

Table 2: Probability of having a job (standard errors in parentheses); * indicates significance at 5% and ** at 1%; all coefficients are normalized such that $\|\boldsymbol{b}\| = 1$.

	Probit		Smoothed binary QR	
	APE	PEAM	APE	PEAM
Age	0.00669**	0.00766**	0.00569**	0.00456**
	(0.000447)	(0.000498)	(0.000485)	(0.000361)
Years of education	0.0138**	0.0158**	0.00853**	0.00919**
	(0.000910)	(0.00116)	(0.00120)	(0.00111)
Children under 6 yrs	−0.0133*	−0.0153*	−0.0250**	−0.0326**
	(0.00534)	(0.00635)	(0.00544)	(0.00565)
Married	0.0388**	0.0382**	0.0557**	0.0582**
	(0.00830)	(0.00765)	(0.00753)	(0.00777)
Couple working	−0.00852	−0.0101	−0.0217**	−0.0212*
	(0.00770)	(0.00927)	(0.00833)	(0.00871)
Household head	0.0331**	0.0346**	0.0303**	0.0274**
	(0.00678)	(0.00750)	(0.00694)	(0.00717)
Observations	9,601	9,601	9,601	9,601

Table 3: Partial effects on probability of having a job (standard errors in parentheses); * indicates significance at 5% and ** at 1%.

between accuracy and computation speed, the first exercise took about 33 minutes while the second only 12 minutes.[4]

```
* Probit (APE)
. probit work c.age c.educ c.children i.married i.couplewrk i.head

[-OUTPUT OMITTED-]

. margins, dydx(_all) post

[-OUTPUT OMITTED-, see Table 3]

* Probit (PEAM)
. probit work c.age c.educ c.children i.married i.couplewrk i.head

[-OUTPUT OMITTED-]

. margins, dydx(_all) ///
at((mean)age educ children married=0 couplewrk=0 head=0) post

[-OUTPUT OMITTED-, see Table 3]

* Smoothed Binary QR
. ldvqreg work age educ children married couplewrk head , reps(100) ///
q(10 20 30 40 50 60 70 80 90) margins(both) xbin(married children head)

(running bqr_est on estimation sample)

[-OUTPUT OMITTED-, see Table 3]
```

As an example, let us analyze the results in Table 3 for the education covariate. On the one hand, the APE measures how much the unconditional probability of working increases with one additional year of education, keeping everything else constant. Therefore, according to the probit model this change is 1.38 percentage points while BQR estimates almost 0.9 percentage points. On the other hand, the PEAM measures the same partial effects but conditional on an individual who has average characteristics (age, education, etc.). Then, education increases its conditional probability of working by 1.58 percentage points according to probit and just over 0.9 percentage points estimated with BQR.

Finally, we compare the conditional probabilities predicted by both methods. Figure 6 shows a plot of the probability of working as a function of age for an individual with 12 years of education, who is single, has no children and is not the head of the household. Note that the probability predicted by a more flexible methodology such as BQR yields higher probabilities compared to a fully parametric model such as probit. Therefore, using one or the other

[4]Calculations includes all 100 bootstrap samples and was measured using Stata 16 MP (64-bit) and Windows 7 operating system (8 GB of RAM and Intel Core i7-3770 processor @ 3.40GHz).

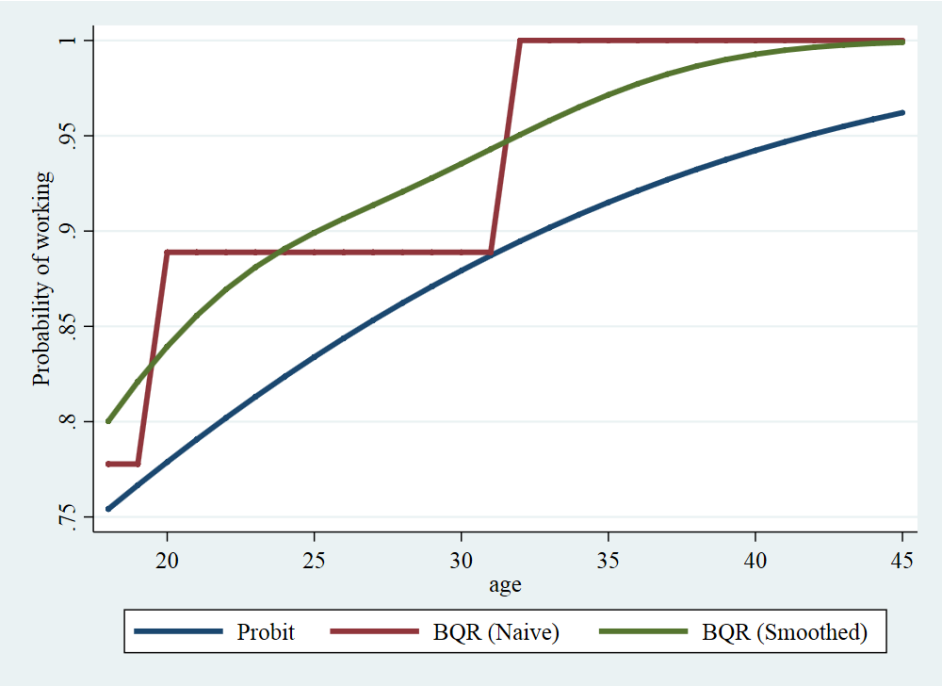

Figure 6: Conditional probability of having a job (Uruguay, ECH, 2015). Plot formats (colors, font, etc.) depend on the version of Stata.

methodology may lead to different predicted probabilities when the latent variable presents heteroskedasticity and/or asymmetries, among others non standard characteristics.

5. Conclusions

This paper proposes a new Stata command `ldvqreg` to estimate censored quantile regression and binary regression models. A key feature of the proposed command is that it implements a smoothed version of the quantile regression model. Thus, it works very well for the case of censored and binary dependent variable.

We illustrate the potential pitfalls of ignoring the censoring mechanisms. In the simulation exercises we compare the standard quantile regression estimates with our corrected procedure. The results highlight that the former may be biased in the usual way. In fact this is the same issue that may appear if we compare the tobit model with standard OLS. The same applies to the binary regression model. In this case, we show that the probit model may be biased in the underlying data generating process is not homoskedastic, asymmetric and/or non-gaussian, while the implementation using standard quantile regression estimates may suffer from convergence problems. In all cases the command `ldvqreg` clearly solves this issues.

References

Baker M (2013). "**mcmccqreg**: Stata Module to Perform Simulation Assisted Estimation of Censored Quantile Regression Using Adaptive Markov Chain Monte Carlo." Statistical

Software Components S457655, Boston College Department of Economics, URL https://ideas.repec.org/c/boc/bocode/s457655.html.

Buchinsky M (1991). "Methodological Issues in Quantile Regression, Chapter 1 of the Theory and Practice of Quantile Regression." Ph.D. dissertation, Harvard University.

Buchinsky M (1995). "Estimating the Asymptotic Covariance Matrix for Quantile Regression Models. A Monte Carlo Study." *Journal of Econometrics*, **68**, 303–338. doi:10.1016/0304-4076(94)01652-g.

Buchinsky M, Hahn J (1998). "An Alternative Estimator for the Censored Regression Model." *Econometrica*, **66**, 653–671. doi:10.2307/2998578.

Chay KY, Powell JL (2001). "Semiparametric Censored Regression Models." *Journal of Economic Perspectives*, **15**(4), 29–42. doi:10.1257/jep.15.4.29.

Chernozhukov V, Fernandez-Val I, Han S, Kowalski AE (2012). "**cqiv**: Stata Module to Perform Censored Quantile Instrumental Variables Regression." Statistical Software Components S457478, Boston College Department of Economics, URL https://ideas.repec.org/c/boc/bocode/s457478.html.

Chernozhukov V, Hong H (2002). "Three-Step Censored Quantile Regression and Extramarital Affairs." *Journal of the American Statistical Association*, **97**, 872–882. doi:10.1198/016214502388618663.

de Castro L, Galvao AF, Kaplan DM, Liu X (2019). "Smoothed GMM for Quantile Models." *Journal of Econometrics*, **213**, 121–144. doi:10.1016/j.jeconom.2019.04.008.

Fitzenberger B (1997). "A Guide to Censored Quantile Regressions." *Handbook of Statistics*, **15**, 405–437. doi:10.1016/s0169-7161(97)15017-9.

Frumento P (2021). ***ctqr**: Censored and Truncated Quantile Regression*. doi:10.32614/CRAN.package.ctqr. R package version 2.0.

Frumento P, Bottai M (2017). "An Estimating Equation for Censored and Truncated Quantile Regression." *Computational Statistics & Data Analysis*, **113**, 53–63. doi:10.1016/j.csda.2016.08.015.

Hahn J (1995). "Bootstrapping Quantile Regression Estimators." *Econometric Theory*, **11**(1), 105–121. doi:10.1017/s0266466600009051.

Horowitz JL (1992). "A Smoothed Maximum Score Estimator for the Binary Response Model." *Econometrica*, **60**, 505–531. doi:10.2307/2951582.

Jolliffe D, Krushelnytskyy B, Semykina A (2000). "Censored Least Absolute Deviations Estimator: CLAD." *Stata Technical Bulletin*, **58**, 13–16. URL https://www.stata.com/products/stb/journals/stb58.pdf.

Kaplan DM, Sun Y (2017). "Smoothed Estimating Equations for Instrumental Variables Quantile Regression." *Econometric Theory*, **33**, 105–157. doi:10.1017/s0266466615000407.

Koenker R (2005). *Quantile Regression*. Cambridge University Press, New York.

Koenker R (2008). "Censored Quantile Regression Redux." *Journal of Statistical Software*, **27**(6), 1–25. doi:10.18637/jss.v027.i06.

Koenker R (2021). ***quantreg**: Quantile Regression*. doi:10.32614/CRAN.package.quantreg. R package version 5.97.

Koenker R, Bassett GW (1978). "Regression Quantiles." *Econometrica*, **46**, 33–49. doi:10.2307/1913643.

Kordas G (2006). "Smoothed Binary Regression Quantiles." *Journal of Applied Econometrics*, **21**(3), 387–407. doi:10.1002/jae.843.

Lin G, He X, Portnoy S (2012). "Quantile Regression with Doubly Censored Data." *Computational Statistics & Data Analysis*, **56**(4), 797–812. doi:10.1016/j.csda.2011.03.009.

Manski C (1975). "Maximum Score Estimation of the Stochastic Utility Model of Choice." *Journal of Econometrics*, **3**(3), 205–228. doi:10.1016/0304-4076(75)90032-9.

Manski C (1985). "Semiparametric Analysis of Discrete Response: Asymptotic Properties of the Maximum Score Estimator." *Journal of Econometrics*, **27**(3), 313–333. doi:10.1016/0304-4076(85)90009-0.

Manski C (1991). "Regression." *Journal of Economic Literature*, **29**, 34–50. doi:10.1016/0168-1591(91)90237-r.

Powell JL (1984). "Least Absolute Deviations Estimation for the Censored Regression Model." *Journal of Econometrics*, **25**, 303–325. doi:10.1016/0304-4076(84)90004-6.

Powell JL (1986). "Censored Regression Quantiles." *Journal of Econometrics*, **32**, 143–155. doi:10.1016/0304-4076(86)90016-3.

R Core Team (2025). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. doi:10.32614/R.manuals. URL https://www.R-project.org/.

StataCorp (2019). *Stata Statistical Software: Release 16*. StataCorp LLC, College Station. URL https://www.stata.com/.

Sun J (2006). *The Statistical Analysis of Interval-Censored Failure Time Data*. Springer-Verlag.

Wichert L, Wilke RA (2008). "Simple Non-Parametric Estimators for Unemployment Duration Analysis." *Journal of the Royal Statistical Society C*, **57**(1), 117–126. doi:10.1111/j.1467-9876.2007.00604.x.

Wooldridge J (2010). *Econometric Analysis of Cross Section and Panel Data, 2nd Edition*. The MIT Press, Cambridge.

Affiliation:

Javier Alejo
Universidad de la República
IECON-UdelaR *and* SNI
Montevideo, Uruguay
E-mail: javier.alejo@fcea.edu.uy

Gabriel Montes-Rojas
CONICET-Universidad de Buenos Aires
Instituto Interdisciplinario de Economía Política
Ciudad Autónoma de Buenos Aires, Argentina
E-mail: gabriel.montes@economicas.uba.ar

Journal of Statistical Software https://www.jstatsoft.org/
published by the Foundation for Open Access Statistics https://www.foastat.org/
September 2025, Volume 114, Issue 8 *Submitted:* 2023-09-21
doi:10.18637/jss.v114.i08 *Accepted:* 2024-09-13