

sivqr: Smoothed IV quantile regression

David M. Kaplan
Department of Economics
University of Missouri
Columbia, MO
kaplandm@missouri.edu

Abstract. In this article, I introduce the `sivqr` command, which estimates the coefficients of the instrumental variables (IV) quantile regression model introduced by Chernozhukov and Hansen (2005). The `sivqr` command offers several advantages over the existing `ivqreg` and `ivqreg2` commands for estimating this IV quantile regression model, which complements the alternative “triangular model” behind `cqiv` and the unconditional “local quantile treatment effect” model of `ivqte`. Computationally, `sivqr` implements the smoothed estimator of Kaplan and Sun (2017), who show that smoothing improves both computation time and statistical accuracy. Standard errors are computed by Bayesian bootstrap; for non-i.i.d. sampling, `sivqr` is compatible with `bootstrap`. I discuss syntax and the underlying methodology, and I compare `sivqr` with other commands in examples.

Keywords: st0001, `sivqr`, endogeneity, instrumental variables, quantile regression

1 Introduction

The instrumental variables quantile regression (IVQR) model of Chernozhukov and Hansen (2005) has become popular for modeling causal effects that differ for unobserved reasons, but the existing IVQR Stata commands `ivqreg` and `ivqreg2` have several limitations, as seen in the examples in section 4. First, `ivqreg` (implementing Chernozhukov and Hansen 2006) only allows a single endogenous term. Besides excluding multiple endogenous regressors, this also excludes interaction or quadratic terms involving the endogenous regressor, for example. Second, `ivqreg2` (implementing Machado and Santos Silva 2019) imposes a location–scale model that seems to cause problems in several of the examples. Sometimes the estimates are an order of magnitude closer to zero (compared to `sivqr` and `ivqreg`), with or without issuing “WARNING: some fitted values of the scale function are negative”; sometimes standard errors are missing; sometimes it takes 30 minutes to run instead of 30 seconds for `sivqr`. Machado and Santos Silva (2019, p. 159) write, “In these linear models, the validity of [our model] depends on assumptions that are stronger than those required by the IVQR but, when these assumptions are valid, [our estimator] has some potential advantages,” like ensuring non-crossing of structural quantile functions. Ideally, these two estimators could be combined using a “model averaging” approach like in Cheng et al. (2019) or Liu (2019); meanwhile, simply running both may be insightful. Third, `ivqreg` and `ivqreg2` share some minor inconveniences, like not supporting syntax for factor variables and interaction terms. Fourth, `ivqreg` is seemingly not actively supported and has additional minor problems, like incorrect coefficient labels when using an asterisk in the regressor

list (like `xvar*`) and leaving the dataset with extra rows of missing values if the original number of observations is below 200.

In contrast to these limitations, `sivqr` can handle models with multiple endogenous terms and produce reasonable estimates in a reasonable amount of time. It implements the smoothed IVQR estimator of Kaplan and Sun (2017). Syntax is similar to the `ivregress` command (see [R] `ivregress`) but additionally specifying a quantile level. Likewise, syntax is similar to the `qreg` command (see [R] `qreg`) but additionally specifying instruments for the endogenous regressors. With non-i.i.d. sampling, the `bootstrap` prefix can be used for standard errors; see [R] `bootstrap` and the example in section 4.2.

Other Stata commands that address endogeneity in “quantile regression” are based on different models. These alternative models have neither “stronger” nor “weaker” assumptions; they may be more plausible in certain settings and less plausible in others. The `ivqte` command (Frölich and Melly 2010) estimates the unconditional (default) or conditional (option `aai`) “local quantile treatment effect,” analogous to the local average treatment effect model of Imbens and Angrist (1994); see Frölich and Melly (2013) for details, as well as section 10.5 of Melly and Wüthrich (2017) for a comparison with the IVQR model and Wüthrich (2020) for certain robustness properties of the IVQR estimator (like `sivqr`). Compared to `ivqte`, the main advantage of `sivqr` is the ability to handle more than a single, binary endogenous regressor. The `cqiv` command (Chernozhukov et al. 2019) uses a control function estimator based on a triangular model (Lee 2007), also allowing for censoring (Chernozhukov et al. 2015). Compared to `cqiv`, the main advantages of `sivqr` are handling multiple and/or non-continuous endogenous regressors, as well as allowing simultaneity and reverse causality. For further comparison of the IVQR and triangular models, see section 9.2.5 of Chernozhukov et al. (2017).

Unfortunately, it seems all these commands require “strong” instruments for valid standard errors. In the future, it would be valuable to have Stata implementations of IVQR inference methods robust to weak instruments, such as those of Chernozhukov and Hansen (2008), Chernozhukov et al. (2009), Chernozhukov et al. (2017, §9.3.3), and references therein. See also Kaplan and Liu (2021) for a “*k*-class” IVQR estimator that can reduce estimation bias from weak instruments, as well as simulation evidence about the usefulness of the conventional “first-stage” *F* statistic for assessing instrument strength in IVQR.

Section 2 discusses the methodology at a relatively intuitive level. Section 3 describes syntax and usage of `sivqr`. Section 4 provides examples that can be replicated with the provided do-file. Section 5 shows some of the theoretical foundations before concluding. Abbreviations are used for instrumental variables (IV), quantile regression (QR), IV quantile regression (IVQR), and two-stage least squares (2SLS).

2 A gentle introduction to methodology

This section discusses methodology at a relatively non-technical level (compared to section 5).

2.1 Parameter interpretation

First, consider interpretation of the parameters being estimated. For simplicity, imagine a single regressor x and outcome variable y , both scalars. Scalars u and v represent unobserved variables.

Usually there is a structural model like $y = \beta_0 + \beta_1 x + v$. Here, β_0 and β_1 are unknown constants and v is everything besides x that causally determines y . Here, β_1 has a causal interpretation as some effect of x on y , but in reality we rarely believe such an effect is the same for all individuals (or firms, or schools, etc.). There are two approaches: either let differences in effects go into v and interpret β_1 as some sort of average, or try to learn about the effect heterogeneity.

Consider a structural model that allows individuals to each have their own intercept and slope. Since the parameters are now individual-specific, they are not constants like β_0 and β_1 were, but rather random variables in a “random coefficients” model. The additive error term v from before is simply absorbed into the random intercept, so the structural model is $y = b_0 + b_1 x$. That is, each individual has their own (y, x, b_0, b_1) , but only (y, x) is observable.

Now imagine the random coefficients can each be written as deterministic functions of a scalar unobservable u : $b_0 = \beta_0(u)$ and $b_1 = \beta_1(u)$. Because $\beta_0(\cdot)$ and $\beta_1(\cdot)$ are unrestricted, the distribution of u can be normalized to uniform over the unit interval $[0, 1]$. The functions $\beta_0(\cdot)$ and $\beta_1(\cdot)$ are unknown but deterministic; evaluated at a fixed $0 < \tau < 1$, $\beta_0(\tau)$ and $\beta_1(\tau)$ are unknown constants, just like β_0 and β_1 were before. The differences across individuals are driven by u . Each individual has their own (y, x, u) , with $y = \beta_0(u) + \beta_1(u)x$, whereas functions $\beta_0(\cdot)$ and $\beta_1(\cdot)$ are not specific to any individual.

A special case of this random coefficient model is the usual structural model with constant (non-random) coefficients. Let $\beta_1(u) = \beta_1$, a constant that does not depend on u . Define $v \equiv \beta_0(u) - \beta_0$, so $\beta_0(u) = \beta_0 + v$. The function $\beta_0(\cdot)$ can be interpreted as the inverse CDF (quantile function) of v , shifted by β_0 ; for example, with $\Phi(\cdot)$ as the standard normal CDF, $v = \Phi^{-1}(u)$ has a standard normal distribution. Then

$$y = \beta_0(u) + \beta_1(u)x = \beta_0 + v + \beta_1 x = \beta_0 + \beta_1 x + v.$$

Some additional restrictions are required in order to learn about the structural model. Imagine further that given x , $\beta_0(u) + \beta_1(u)x$ is increasing in u . This is known as a “monotonicity” assumption. It also explains why u is often called the “rank variable”: it describes how somebody’s y would rank in the population if everyone were forced to have the same x . For example, somebody with $u = 0.5$ would have median y ,

and somebody with $u = 0.9$ would have 90th percentile y . If everybody keeps the same “rank” regardless of the x value, then “rank invariance” holds. A weaker assumption called “rank similarity” allows the ranking to differ across x as long as the differences are exogenous.

Even if x is endogenous, IVQR can estimate $\beta_0(\tau)$ and $\beta_1(\tau)$ for any $0 < \tau < 1$ if an instrument z is available that is related to x but independent of u (Chernozhukov and Hansen 2005). The interpretation of these parameters depends partly on the rank assumption. If rank invariance holds, then $\beta_0(\tau) + \beta_1(\tau)x_0$ is the y value that somebody with rank $u = \tau$ would have if we assign them to have value $x = x_0$. Even with the weaker rank similarity assumption, this is the τ -quantile structural function of Imbens and Newey (2009, §3.1): given any $x = x_0$, it provides the τ -quantile of $\beta_0(u) + \beta_1(u)x_0$ over the unconditional population distribution of u (uniform over $[0, 1]$), which is $\beta_0(\tau) + \beta_1(\tau)x_0$ due to monotonicity. Similarly, $\beta_1(\tau)$ can be interpreted as a τ -quantile treatment effect, capturing the difference in the τ -quantile of y between the counterfactual “untreated” distribution for which everyone has $x = x_0$ and the counterfactual “treated” distribution for which everyone has $x = x_0 + 1$.

2.2 Estimation

Chernozhukov and Hansen (2005) show how to derive moment conditions (or “estimating equations”) to characterize the parameters $\beta_0(\tau)$ and $\beta_1(\tau)$ given a valid instrument and the assumptions discussed above. For comparison, with $v \equiv y - \beta_0 - \beta_1x$ the standard IV moment conditions are

$$\begin{aligned} 0 &= \mathbb{E}(v) = \mathbb{E}(y - \beta_0 - \beta_1x) \\ 0 &= \mathbb{E}(zv) = \mathbb{E}[z(y - \beta_0 - \beta_1x)] \end{aligned}$$

The IVQR moment conditions are

$$\begin{aligned} 0 &= \mathbb{E}[1\{y - \beta_0(\tau) - \beta_1(\tau)x \leq 0\} - \tau] \\ 0 &= \mathbb{E}[z(1\{y - \beta_0(\tau) - \beta_1(\tau)x \leq 0\} - \tau)] \end{aligned} \tag{1}$$

where $1\{\cdot\}$ is the indicator function defined as $1\{\mathcal{A}\} = 1$ if \mathcal{A} is true and otherwise $1\{\mathcal{A}\} = 0$. This is implied by a conditional quantile restriction on $y - \beta_0(\tau) - \beta_1(\tau)x$ (given z), mirroring how the standard IV moments are implied by a conditional mean restriction on $y - \beta_0 - \beta_1x$ (given z).

Smoothing solves the computational difficulties inherent in (1). Unlike the standard IV moment conditions, the IVQR moment conditions cannot be solved explicitly for the parameters, nor are the sample moment conditions smooth (differentiable) functions of the parameters. This computational challenge is addressed by “smoothing” the indicator function: replacing it with a continuously differentiable version that smoothly (if quickly) decreases from 1 to 0 rather than discontinuously jumping from 1 to 0. This smoothing allows the sample system of equations defining the estimator to be solved by standard numerical methods like those available in Mata. As a bonus, smoothing also improves the statistical properties of the estimator in theory and simulations; see sections 5 and 7 of Kaplan and Sun (2017).

3 The `sivqr` command

The `sivqr` command estimates the coefficients in an instrumental variables quantile regression (IVQR) model, as well as standard errors.

Syntax, options, and stored results are now shown. Prefix `by` is allowed (see [D] `by`).

3.1 Syntax

```
sivqr devar [varlist1] (varlist2 = varlistiv) [if] [in] [weight] ,
    quantile(#) [bandwidth(#) level(#) reps(#) logiterations
    noconstant seed(#) nodots]
```

As in `ivregress`: *varlist*₁ has exogenous regressors (or control variables), *varlist*₂ has endogenous regressors, and *varlist*_{iv} has excluded instruments (exogenous variables that instrument for *varlist*₂).

`pweights`, `iweights`, and `fweights` are allowed; see [U] **11.1.6 weight**.

3.2 Options

`quantile`(#) specifies the quantile level as in [R] `qreg`: a real number strictly between 0 and 1, or alternatively a number between 1 and 100 interpreted as a percentile. For example, `quantile(0.5)` specifies the median, as does `quantile(50)`. To prevent inadvertent mistakes, there is no default; this must be specified explicitly.

`bandwidth`(#) specifies the desired smoothing bandwidth. Any negative value invokes a plug-in bandwidth based on Kaplan and Sun (2017), which is the default. If the desired bandwidth is too small to find a numerical solution, then it is increased until a solution is found. For example, specifying `bandwidth(0)` requests the smallest bandwidth that is computationally feasible.

`level`(#) specifies the confidence level, as a percentage, for confidence intervals. The default is `level(95)` or as set by `set level`; see [U] **20.7 Specifying the width of confidence intervals**.

`reps` specifies the number of bootstrap replications. The default is `reps(20)`, which is usually fine for exploratory analysis but should be increased for final results. To reduce computation time by omitting standard errors altogether, use `reps(0)`.

`logiterations` prints each iteration of the numerical solver; see [M-5] `solvenl()`, in particular `solvenl_init_iter_log()`. The default is not to print such information, which usually only helps debugging and troubleshooting.

`noconstant` omits the intercept term that otherwise is included automatically.

`seed` sets the random-number seed, to make results replicable; default is `seed(112358)`.

The current seed is restored at the end of execution.

`nodots` suppresses display of the replication dots (see [R] `bootstrap`).

3.3 Stored results

`sivqr` stores the following in `e()`:

Scalars

<code>e(N)</code>	number of observations	<code>e(reps)</code>	number of bootstrap replications
<code>e(bwidth)</code>	smoothing bandwidth used	<code>e(q)</code>	quantile level requested
<code>e(bwidth_req)</code>	smoothing bandwidth requested (or plug-in value)		

Macros

<code>e(cmd)</code>	<code>sivqr</code>	<code>e(constant)</code>	<code>noconstant</code> if specified
<code>e(instd)</code>	instrumented variable(s)	<code>e(insts)</code>	instrument(s)
<code>e(depvar)</code>	name of dependent variable	<code>e(exogr)</code>	exogenous regressors
<code>e(wtype)</code>	weight type	<code>e(wexp)</code>	weight expression
<code>e(properties)</code>	<code>b V</code> or else only <code>b</code> if <code>reps(0)</code>	<code>e(title)</code>	title in estimation output

Matrices

<code>e(b)</code>	estimated coefficient vector	<code>e(V)</code>	estimated variance-covariance matrix of the estimator, unless <code>reps(0)</code>
-------------------	------------------------------	-------------------	--

Functions

<code>e(sample)</code>	marks estimation sample
------------------------	-------------------------

Almost all of the above are the same as for the familiar commands `ivregress` and/or `qreg` (or `bsqreg`). The only exceptions are for the smoothing bandwidth. If the user specifies a negative value for the bandwidth option, like `bandwidth(-1)`, then the plug-in bandwidth is computed and returned in `e(bwidth_req)`. If the numerical solver cannot find the solution with `e(bwidth_req)` because it is too close to zero, then the bandwidth is increased until the numerical solver finds the solution. This feasible bandwidth is returned in `e(bwidth)`. There is nothing wrong with these values being different; for example, specifying `bandwidth(0)` makes `e(bwidth_req)` zero and `e(bwidth)` the smallest possible bandwidth for which the numerical solver finds a solution.

`sivqr` stores the following in `r()`:

Scalar

<code>r(level)</code>	confidence level
-----------------------	------------------

Matrix

<code>r(table)</code>	table of results
-----------------------	------------------

These are the standard values stored by `ereturn display`; see [P] `ereturn`.

4 Examples

All examples can be replicated with the provided `sivqr_examples.do` file. I ran them in Stata/SE 14.2 on Windows 10, on a few-years-old standard-issue Dell desktop computer (Intel i5-8500 3GHz processor, 8GB RAM).

The primary focus is comparison with `ivqreg` and `ivqreg2`, in terms of estimates, speed, and capabilities. Some recurring patterns are:

1. `sivqr` with a small bandwidth like `b(0)` produces estimates very similar to `ivqreg`;
2. `ivqreg2` often has estimates that differ greatly from `ivqreg` and `sivqr` (and other estimators, when applicable);
3. `ivqreg` does not allow multiple endogenous regressors;
4. `ivqreg2` can be significantly slower than `sivqr` to compute an estimate (like 30 minutes instead of 30 seconds).

4.1 Example 1 from `ivregress`

The following extends the familiar Example 1 in [R] `ivregress`. Briefly, it examines the causal relationship between owner-occupied housing values and rent; see [R] `ivregress` for details. Here, the quantile level represents the level of unmodeled determinants of rent (the u from section 2.1), normalized to be uniform over the $[0, 1]$ interval.

For the median, `sivqr` produces the following results with the plug-in bandwidth and 200 bootstrap replications in five seconds.

```
. webuse hsg2 , clear
(1980 Census housing data)
. sivqr rent pcturban (hsgval = faminc i.region), q(0.50) reps(200)
Computing plug-in bandwidth because bandwidth(-1)<0
Bootstrap replications (200)
-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
..... 50
..... 100
..... 150
..... 200
Smoothed instrumental variables quantile regression (SIVQR)   Quantile = .5
Smoothing bandwidth used = 7.093792                        Number of obs = 50
```

rent	Coef.	Std. Err.	z	P> z	[95% Conf. Interval]	
hsgval	.0018274	.0003946	4.63	0.000	.001054	.0026008
pcturban	.5394268	.346328	1.56	0.119	-.1393636	1.218217
_cons	109.0715	14.45037	7.55	0.000	80.74933	137.3937

```
Instrumented: hsgval
Instruments: pcturban faminc i.region
```

The format of the output is essentially the same as for `ivregress 2sls`, besides additional information about the quantile level, smoothing bandwidth, and bootstrap replications.

Omitting the various `timer` and matrix storage commands (and output), the following code produces the values seen in table 1. Region dummies were generated before running `ivqreg2` and `ivqreg` because they do not support the `i.region` factor variable syntax used above with `sivqr`.

```
. forv i = 2/4 {
2.   gen dregion`i' = (region==`i')
3. }
. forv i = 1/3 {
2.   local q = `i'/4
3.   local b = 5 + 20*abs(`i'-2)
4.   sivqr rent pcturban (hsngval = faminc dregion*), q(`q') reps(0) b(`b')
5.   capture ivqreg rent pcturban (hsngval = faminc dregion*), q(`q') // error...
6. }
. ivqreg2 rent hsngval pcturban , q(.25 .5 .75) inst(faminc dregion* pcturban)
. forv i = 1/3 {
2.   local q = 25*`i'
3.   cqiv rent pcturban (hsngval = faminc dregion*) , uncensored q(`q')
6. }
```

Table 1: Estimates of `hsngval` coefficient and runtimes (seconds).

Quantile	2SLS	<i>sivqr</i>	<i>ivqreg</i>	<i>ivqreg2</i>	<i>cqiv</i>
0.25	0.002240	0.001545	(error)	0.000064	0.001490
0.50	0.002240	0.001800	(error)	0.000044	0.002092
0.75	0.002240	0.002536	(error)	0.000008	0.003518
Runtime	0.3	0.2	22.0	18.0	4.7
100 reps		17.7			

Table 1 compares results across quantile levels and estimation commands. The familiar 2SLS estimate of the coefficient on `hsngval` is 0.0022. This value is simply copied into each row because it does not depend on the quantile level. Each of the three IVQR commands is discussed in turn. For `sivqr`, to help minimize bias, the specified bandwidths are 25, 5, and 25, respectively (across the three quantiles), which are roughly the smallest for which `solven1` can find a numerical solution.

Table 1 shows that the `sivqr` estimates are similar to 2SLS but show an increasing pattern: 0.0015 (below 2SLS), 0.0018, 0.0025 (above 2SLS). That is, given higher levels of the unmodeled determinants of rent, the relationship between owner-occupied housing values and rent is more positive.

Table 1 shows `ivqreg2` estimates all 30–300 times smaller than those of 2SLS, and similarly much smaller than the `sivqr` or `cqiv` estimates. Although the IVQR coefficient at a particular quantile level may differ greatly from 2SLS, it should not be systematically much closer to zero at all quantiles. This possible bias toward zero appears in other examples, too. The computation time is similar to `sivqr` with 100 bootstrap replications, but much slower than `sivqr` with zero bootstrap replications. For final published

results, even more than 100 bootstrap replications may be desired, but for exploratory work where interest is more in coefficient estimates than standard errors, using `sivqr` would be faster in this example. One caveat is that because `ivqreg2` uses a location-scale model, the additional time to compute additional quantile levels is negligible, so if we wanted to run $q = 0.01, 0.02, \dots, 0.99$ every time (as well as hundreds of bootstrap replications), then `ivqreg2` would be faster. That said, the same location-scale model is probably the cause of the estimates being seemingly incorrect. Also, the decreasing pattern of `ivqreg2` estimates across quantile levels contradicts the increasing pattern shared by `sivqr` and `cqiv`.

Table 1 also shows that `ivqreg` simply failed to compute any estimates in this example. In fairness, it is a relatively difficult example due to the very small sample size (50 observations). This difficulty is also reflected by `sivqr` not being able to find a numerical solution with even the plug-in bandwidth in some cases. Usually, it is helpful to run `sivqr` with a bandwidth much smaller than the plug-in (like 5 or 50 times smaller, or more), to see if the plug-in bandwidth may be causing excessive bias. Despite the difficulty, `sivqr` was able to compute seemingly reasonable results, whereas `ivqreg` could not produce any result and the `ivqreg2` results do not seem reasonable.

4.2 Example 4 from `ivregress`

The following extends Example 4 in [R] `ivregress`. Briefly, it hopes to estimate a structural wage model, treating job tenure as endogenous; see [R] `ivregress` for details. I focus on the coefficient on job tenure.

For comparison, `ivregress 2sls` is run first. This provides a helpful “sanity check”: even if there are big differences across quantile levels, the IVQR coefficient estimates and signs should still be reasonably similar overall, especially at the median.

```
. webuse nlswork , clear
(National Longitudinal Survey. Young Women 14-26 years of age in 1968)
. xtset idcode
      panel variable: idcode (unbalanced)
. ivregress 2sls ln_wage c.age##c.age birth_yr grade (tenure = union wks_wo
> rk msp) , vce(cluster idcode)
Instrumental variables (2SLS) regression           Number of obs   =   18,625
                                                    Wald chi2(5)     =   1600.81
                                                    Prob > chi2      =    0.0000
                                                    R-squared        =    .
                                                    Root MSE        =    .48441

                                                    (Std. Err. adjusted for 4,110 clusters in idcode)
```

ln_wage	Coef.	Robust Std. Err.	z	P> z	[95% Conf. Interval]	
tenure	.1060832	.0044835	23.66	0.000	.0972957	.1148707
age	.0162345	.0069467	2.34	0.019	.0026193	.0298497
c.age#c.age	-.0005309	.0001155	-4.59	0.000	-.0007573	-.0003044
birth_yr	-.0091139	.0022602	-4.03	0.000	-.0135438	-.0046839

grade	.070454	.0031158	22.61	0.000	.0643471	.0765609
_cons	.9079537	.1681148	5.40	0.000	.5784548	1.237453

```
Instrumented: tenure
Instruments: age c.age#c.age birth_yr grade union wks_work msp
```

The following shows results from `sivqr` at the median level. To cluster standard errors at the individual level, the `bootstrap` prefix is used. With `bootstrap`, it saves time to specify the `reps(0)` option for `sivqr`. The `sivqr` coefficient estimates (other than the intercept) as well as the corresponding standard errors are both similar to those from `ivregress 2sls`. Only 20 bootstrap replications are used here (to reduce computation time); this is sufficient for exploratory analysis, but should ideally be increased when running final results.

```
. bootstrap , reps(20) cluster(idcode) seed(112358) : sivqr ln_wage c.age#c.age
> birth_yr grade (tenure = union wks_work msp) , q(0.50) reps(0)
(running sivqr on estimation sample)
```

```
Bootstrap replications (20)
```

```
-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 1 | 2 | 3 | 4 | 5 |
.....
```

```
Smoothed instrumental variables quantile regression (SIVQR)   Quantile = .5
Smoothing bandwidth used = .0600669                       Number of obs = 18,625
(Replications based on 4,110 clusters in idcode)
```

ln_wage	Observed Coef.	Bootstrap Std. Err.	z	P> z	Normal-based [95% Conf. Interval]	
tenure	.1076941	.0036231	29.72	0.000	.1005931	.1147952
age	.0060803	.0080756	0.75	0.451	-.0097476	.0219083
c.age#c.age	-.0003585	.0001304	-2.75	0.006	-.000614	-.0001029
birth_yr	-.011967	.0021917	-5.46	0.000	-.0162627	-.0076714
grade	.065723	.0027007	24.34	0.000	.0604298	.0710162
_cons	1.255389	.1922238	6.53	0.000	.8786375	1.632141

```
Instrumented: tenure
Instruments: age c.age#c.age birth_yr grade union wks_work msp
```

Omitting the various `timer` and matrix storage commands (and output), the following code produces the values seen in tables 2 and 3. A small bandwidth of 0.001 is used to make `sivqr` estimates more directly comparable with `ivqreg`; the plug-in bandwidths are all above 0.05. The squared age variable is generated manually because the other IVQR commands do not support the factorial operator `##` used above with `sivqr`.

```
. gen agesq = age^2
(24 missing values generated)
. forv i = 1/3 {
  2. local q = `i'/4
  3. local p = 25*`i'
  4. sivqr ln_wage age agesq birth_yr grade (tenure = union wks_work msp) , q(`q`)
> reps(0) b(0.001)
  5. ivqreg ln_wage age agesq birth_yr grade (tenure = union wks_work msp) , q(`q`)
  6. cqiv ln_wage age agesq birth_yr grade (tenure = union wks_work msp) , q(`p`)
```

```

> uncensored
7. }
. ivqreg2 ln_wage age agesq birth_yr grade tenure , q(.25 .5 .75) inst(union
> wks_work msp age agesq birth_yr grade)

```

Table 2: Estimates of `tenure` coefficient.

Quantile	2SLS	<code>sivqr</code>	<code>ivqreg</code>	<code>ivqreg2</code>	<code>cqiv</code>
0.25	0.106	0.086	0.086	0.023	0.045
0.50	0.106	0.108	0.103	0.017	0.044
0.75	0.106	0.156	0.158	0.011	0.039

Table 3: Runtimes (seconds) of IVQR commands on `nlswork` data.

Quantile	Linear in <code>tenure</code>			Quadratic	
	<code>sivqr</code>	<code>ivqreg</code>	<code>ivqreg2</code>	<code>sivqr</code>	<code>ivqreg2</code>
0.25	59	106	↑		
0.50	13	85	2524	27	2996
0.75	27	129	↓		

Table 2 shows estimates from `sivqr` and `ivqreg` that seem reasonable and exhibit economically significant heterogeneity in the effect of job tenure on wage. The estimated coefficient on job tenure is nearly twice as large with `q(0.75)` as with `q(0.25)`; the former is above the 2SLS estimate, while the latter is below. The `sivqr` runtime without bootstrap in table 3 is a few times faster, so it may be faster for exploratory analysis, but slower than `ivqreg` when running many bootstrap replications.

Table 2 shows that the `ivqreg2` estimates are again much closer to zero than 2SLS, `sivqr`, and `ivqreg`. Also similar to table 1, the `ivqreg2` estimates follow the opposite pattern of `sivqr` and `ivqreg`, decreasing instead of increasing with the quantile level. Further, sometimes the sign (+ or -) differs for the `ivqreg2` estimates of other coefficients (not shown in the table); for example, at the 0.5 and 0.75 quantile levels, the `ivqreg2` coefficient on education (`grade`) is negative, whereas it is positive (around 0.07) for 2SLS, `sivqr`, and `ivqreg` alike. Again, the most likely explanation of all these discrepancies is that the location-scale model is incorrect here. Comparing the combined runtime for all three quantiles, `ivqreg2` takes over 40 minutes, compared to under 2 minutes for `sivqr`. This makes `sivqr` more useful for exploratory analysis, though again `ivqreg2` can run additional quantile levels in negligible time and does not require additional time to compute bootstrap standard errors. Runtime aside, the primary concern is the large difference between the `ivqreg2` estimates and those of `ivqreg`, `sivqr`, and 2SLS.

To include a quadratic term, only `sivqr` and `ivqreg2` can be used, not `ivqreg` (or `cqiv`). In the following code, a model is run with `tenure-squared` as another endogenous regressor, to allow the tenure effect to depend on the initial tenure value. In the code, additional instruments are generated as squares and interactions of the original instruments. Table 4 shows that at the median level, the `sivqr` estimates are qualitatively similar to 2SLS (especially at four years or less tenure, which is 74% of the dataset), whereas the `ivqreg2` estimates are again much closer to zero. Further, as seen in table 3, whereas `sivqr` takes around 30 seconds to run, `ivqreg2` takes nearly 50 minutes and then reports “Warning: variance matrix is nonsymmetric or highly singular” with missing values for all the standard errors. Computing bootstrap standard errors is possible in that case, but would take over 50 minutes per replication, so even 30 replications would take over 24 hours to run, as opposed to 15 minutes for `sivqr`.

```
. gen tenuresq = tenure^2
. gen wks_worksq = wks_work^2
. gen unionmsp = union*msp
. ivregress 2sls ln_wage age agesq birth_yr grade (tenure tenuresq = union wks_work
> msp wks_worksq unionmsp ) , vce(cluster idcode)
. sivqr ln_wage age agesq birth_yr grade (tenure tenuresq = union wks_work msp
> wks_worksq unionmsp ) , q(0.5) reps(0) b(0.01)
. ivqreg2 ln_wage tenure tenuresq age agesq birth_yr grade , q(0.5) inst(age agesq
> birth_yr grade union wks_work msp wks_worksq unionmsp )
```

Table 4: Quadratic estimates, median.

Variable	2SLS	<code>sivqr</code>	<code>ivqreg2</code>
<code>tenure</code>	0.039	0.028	0.013
<code>tenuresq</code>	0.006	0.014	0.001

Finally, the provided code shows some additional important considerations about computation time. Most of the `sivqr` computation time is from calling the `solven1` numerical solver. Naturally, some numerical problems are more difficult to solve quickly. In particular, difficulty is higher with less smoothing or when there is less data near the quantile of interest (roughly speaking), which most often happens farther into the tails at quantile levels near zero or one. Table 5 illustrates this by running `sivqr` over $q = 0.10, 0.15, \dots, 0.90$, using either the plug-in bandwidth (always between 0.05 and 0.08) or a smaller bandwidth 0.005. First, although there are other factors, runtime is generally longer in the tails: with the plug-in bandwidth, runtime is over 10 seconds at quantile levels $\{0.10, 0.80, 0.85, 0.90\}$ but under 10 seconds in between (besides one anomaly at 0.35), and the general pattern is similar with `b(0.005)`. Second, the runtime is longer with the smaller bandwidth at 14 out of 17 quantile levels. Also, even with the same original dataset, some bootstrap datasets are naturally more difficult than others, so bootstrap runtime can depend on the random seed. With `q(0.5)`, out of 100 runs with different seeds, runtimes for the built-in `sivqr` bootstrap with 20 replications

ranged from 12 seconds to 46 seconds, with lower and upper quartiles 15 seconds and 23 seconds. Among five runs with the `bootstrap` prefix (also 20 replications), runtimes ranged from 122 to 142 seconds. However, it cannot be known beforehand which seeds are “faster” for a particular dataset, so using the default seed is recommended.

Table 5: Runtimes (seconds) across quantile levels, `nlswork` data.

Quantile	plug-in	b(0.005)
0.10	21	25
0.15	5	16
0.20	2	4
0.25	4	13
0.30	6	2
0.35	12	38
0.40	6	11
0.45	7	8
0.50	4	7
0.55	5	12
0.60	7	8
0.65	5	7
0.70	6	10
0.75	4	23
0.80	72	23
0.85	49	84
0.90	16	10

4.3 Other examples

The other examples in `sivqr_examples.do` are briefly noted here.

The first additional example revisits the quantile demand curve estimation from Chernozhukov and Hansen (2008) using local weather as an exogenous source of variation in the local fish market supply. Here, it is clearly inappropriate to use `cqiv` (due to simultaneity) or `ivqte` (due to continuous regressor). Unfortunately, standard errors are very large.

The second additional example estimates the effect of 401(k) retirement plan participation on net total financial assets, using 401(k) eligibility to instrument for participation. Both `sivqr` and `ivqreg` estimate conditional quantile treatment effects similar to those of Chernozhukov and Hansen (2004). In contrast, `ivqreg2` displays “WARNING: some fitted values of the scale function are negative” and actually reports negative effects at quantile levels below 0.7, and at seven of nine quantile levels reporting missing standard errors with “Warning: variance matrix is nonsymmetric or highly singular,”

after running for over an hour (versus under a minute for *sivqr*). There is also code to show the difference between these conditional quantile treatment effects and the unconditional effects estimated by *ivqte* (which can be used since the treatment and instrument are binary).

The third additional example estimates Engel curves for household alcohol expenditure as in the *cqiv* help file. Despite another “WARNING: some fitted values of the scale function are negative,” the *ivqreg2* estimates seem reasonably similar to the *sivqr* and *cqiv* estimates with a linear model. However, with a quadratic model, the *ivqreg2* elasticities are very near zero at all expenditure levels (despite no “WARNING”) and very slightly increasing, whereas the *sivqr* and *cqiv* estimates start positive, then near zero, then negative as total household expenditure increases. Also, because it can only handle one endogenous regressor, *cqiv* (or *ivqreg*) must pretend squared log expenditure is exogenous while treating log expenditure as exogenous, whereas *sivqr* can more appropriately model both as endogenous.

5 Methods and formulas

This section contains additional theoretical details. After introducing notation in section 5.1, section 5.2 concerns the difficulty of estimation and the solution of Kaplan and Sun (2017), while section 5.3 has some of the identification arguments from Chernozhukov and Hansen (2005) that characterize causal parameters as solutions to moment conditions (estimating equations). Section 5.4 describes various bandwidths. Section 5.5 has brief notes on the Bayesian bootstrap used to compute standard errors.

These details may provide a deeper understanding for some readers, but they may also be skipped without hindering successful application of *sivqr* in practice.

5.1 Notation

Notationally, let y be the scalar outcome variable, \mathbf{x} a column vector of regressors, and u a scalar unobserved variable. Let \mathbf{d} be a subset of \mathbf{x} containing all the endogenous regressors. Let \mathbf{z} be a column vector of all exogenous variables: both exogenous regressors in \mathbf{x} as well as the excluded instruments (instrumental variables). The scalar random variable u is unobserved. Following *Stata Journal* convention, vectors are lowercase bold (like \mathbf{x}) and scalars are lowercase plain (like y) since uppercase is reserved for matrices, but with some luck, random variables can be distinguished from their possible non-random realizations.

Additional notation is introduced as needed below.

5.2 Estimation

First, as a helpful reference point for intuition, recall the standard (“mean”) instrumental variables (IV) model and estimator. The goal is to estimate the non-random

parameter vector β in the structural model

$$y = \mathbf{x}'\beta + u$$

If the instruments \mathbf{z} are independent of the structural error u , then

$$\mathbf{0} = E(u | \mathbf{z}) \implies \mathbf{0} = E(\mathbf{z}u) = E[\mathbf{z}(y - \mathbf{x}'\beta)] \quad (2)$$

where the \implies follows from the law of iterated expectations. If the dimensions of \mathbf{z} and \mathbf{x} are the same (“exact identification”), then this can be solved explicitly for β :

$$E(\mathbf{z}y) = E(\mathbf{z}\mathbf{x}'\beta) \implies E(\mathbf{z}y) = E(\mathbf{z}\mathbf{x}')\beta \implies \beta = [E(\mathbf{z}\mathbf{x}')]^{-1} E(\mathbf{z}y)$$

and (with i.i.d. data) the expectations are replaced by sample averages to yield the familiar IV estimator.

Superficially, imagine replacing the conditional mean restriction $E(y | \mathbf{z})$ in (2) with a conditional τ -quantile restriction for some $0 < \tau < 1$:

$$\mathbf{0} = Q_\tau(u | \mathbf{z}) \quad (3)$$

That is, conditional on any value of \mathbf{z} , the conditional distribution of u has its τ -quantile equal to zero. By definition, the τ -quantile of a distribution is the value with τ probability below that value, and the same is true conditional on \mathbf{z} . Thus, (3) is equivalent to

$$\tau = \Pr(u \leq 0 | \mathbf{z}) = \Pr(y - \mathbf{x}'\beta \leq 0 | \mathbf{z})$$

Let $1\{\cdot\}$ be the indicator function such that $1\{\mathcal{A}\} = 1$ if \mathcal{A} is true and otherwise $1\{\mathcal{A}\} = 0$. Rewriting $\Pr(\cdot)$ as $E[1\{\cdot\}]$ and applying the law of iterated expectations as in (2),

$$\begin{aligned} \tau = E[1\{y - \mathbf{x}'\beta \leq 0\} | \mathbf{z}] &\implies 0 = E[1\{y - \mathbf{x}'\beta \leq 0\} - \tau | \mathbf{z}] \\ &\implies \mathbf{0} = E[\mathbf{z}(1\{y - \mathbf{x}'\beta \leq 0\} - \tau)] \end{aligned} \quad (4)$$

Despite some similarities, computing an IVQR estimator based on (4) is much more difficult than computing the mean IV estimator based on (2). With mean IV, it was possible to solve for β and replace the mean (expectation) with the sample average. Here, the β is stuck inside the indicator function $1\{\cdot\}$, so it cannot be solved for explicitly. Further, it may be impossible to solve the equation exactly after substituting in the sample average.

These difficulties are both addressed by “smoothing” the indicator function in (4). Replacing $1\{\cdot\}$ with a very similar but continuously differentiable function allows the sample system of equations to be solved quickly by standard numerical methods. Specifically, replacing $1\{v \leq 0\}$, $\tilde{I}(v)$ is a function of v that smoothly decreases from 1 to 0 over $-1 \leq v \leq 1$ instead of decreasing discontinuously from 1 to 0 at $v = 0$.¹ Adding a

1. This notation is simpler and equivalent to (but different than) that in Kaplan and Sun (2017).

bandwidth h allows $\tilde{I}(v/h)$ to decrease more steeply over $-h \leq v \leq h$. The estimator $\widehat{\beta}$ solves the “smoothed estimating equations” (or “smoothed moment conditions”)

$$\mathbf{0} = \frac{1}{n} \sum_{i=1}^n z_i [\tilde{I}((y_i - \mathbf{x}'_i \widehat{\beta})/h) - \tau] \quad (5)$$

As a bonus, this smoothing also improves estimation precision, as shown both theoretically and in simulations by Kaplan and Sun (2017, §§5,7). Weights w_i can be inserted into (5) as needed:

$$\mathbf{0} = \frac{1}{n} \sum_{i=1}^n w_i z_i [\tilde{I}((y_i - \mathbf{x}'_i \widehat{\beta})/h) - \tau] \quad (6)$$

Specifically, `sivqr` uses a piecewise linear $\tilde{I}(\cdot)$ that connects the smoothed IVQR estimator to other estimators. Instead of jumping down from 1 to 0 discontinuously, $\tilde{I}(\cdot)$ has $\tilde{I}(v) = 1$ for $v \leq -1$ and $\tilde{I}(v) = 0$ for $v \geq 1$, but transitions linearly with $\tilde{I}(v) = (1 - v)/2$ for $-1 < v < 1$. Kaplan and Sun (2017, §2.2, p. 111) show how this choice produces the Winsorized mean estimator of the type in Huber (1964, ex. (iii), p. 79) in the special case with $\tau = 0.5$ and an intercept-only model ($\mathbf{x} = \mathbf{z} = 1$). They also show (in Section 2.2) how using a very large amount of smoothing (h) turns the smoothed IVQR estimator into the usual (mean) IV estimator, with an adjusted intercept. Intuitively, if h is large enough that every observation is smoothed, then $\tilde{I}(\cdot)$ is a linear function of the residuals $y_i - \mathbf{x}'_i \widehat{\beta}$, just as in the mean IV moment conditions; if $\tau \neq 0.5$, then the intercept is different, but the slope estimates are identical to the IV slope estimates for any τ . Of course, in practice h is small, but this shows that the worst-case effect of choosing h too large is that you simply get the usual IV slope estimates.

If there are more instruments than parameters (overidentification), then a different \mathbf{z} is used in (5). Specifically, it is replaced by the linear projection of the regressor vector \mathbf{x} onto the instruments \mathbf{z} , which has the same dimension as \mathbf{x} and thus β . This linear projection is motivated by the two-stage least squares estimator, which can also be written as the mean IV estimator when replacing \mathbf{z} with the linear projection of \mathbf{x} onto \mathbf{z} . Although theoretically more efficient estimators may exist, this estimator remains consistent, reliable, and fast to compute regardless of the degree of overidentification.

5.3 Identification

It remains to motivate the conditional quantile restriction in (3) from a causal model, as originally done by Chernozhukov and Hansen (2005).² Instead of assuming every individual (or firm, or school, etc.) has the same coefficients β , imagine each individual has their own coefficient vector \mathbf{b} . This is a “random coefficient” model, meaning \mathbf{b} can differ by individual the same way that y , \mathbf{x} , and \mathbf{z} do:

$$y = \mathbf{x}'\mathbf{b}$$

² See also Chapter 6 of Kaplan (2020) for an introductory discussion.

An additional error term would be redundant; for example, if $y = b_0 + b_1x + v$, then the random intercept can simply absorb v to become $b_0 + v$. To make the model more tractable empirically, imagine \mathbf{b} can be written as a deterministic (but unknown) vector-valued function $\beta(\cdot)$ applied to a scalar unobserved u :

$$y = \mathbf{x}'\beta(u) \tag{7}$$

Assuming u is continuous, it can be normalized to have a $\text{Unif}(0, 1)$ distribution (uniformly distributed between 0 and 1): any transformation is simply absorbed into $\beta(\cdot)$. Assume a “monotonicity” property that given any value of \mathbf{x} , $\mathbf{x}'\beta(u)$ is an increasing function of u , and like before assume the instruments are independent of u . Then,

$$\begin{aligned} \Pr(y \leq \mathbf{x}'\beta(\tau) \mid \mathbf{z}) &= \Pr(\mathbf{x}'\beta(u) \leq \mathbf{x}'\beta(\tau) \mid \mathbf{z}) && \text{by (7)} \\ &= \Pr(u \leq \tau \mid \mathbf{z}) && \text{by monotonicity} \\ &= \Pr(u \leq \tau) && \text{by } u \perp \mathbf{z} \\ &= \tau && \text{by } u \sim \text{Unif}(0, 1) \end{aligned}$$

That is, for any $u = \tau$, $\beta(\tau)$ is identified by the conditional quantile restriction from (3), which can be estimated by the `sivqr` command using (5). Given τ , the function $q_\tau(\mathbf{x}) = \mathbf{x}'\beta(\tau)$ is also the τ -quantile structural function introduced by Imbens and Newey (2009, §3.1).

5.4 Plug-in bandwidth

This section only applies to the plug-in bandwidth, specified by a negative value for the `bandwidth` option of the `sivqr` command; if instead the user-specified bandwidth is non-negative, then none of the following applies.

Kaplan and Sun (2017, prop. 2, p. 117) provide a theoretical “optimal” smoothing bandwidth. Specifically, it minimizes the asymptotic mean squared error of the smoothed estimating equations (moment conditions) themselves. It also minimizes the asymptotic mean squared error of a particular linear combination of the estimated coefficients, although it does not do so for every possible linear combination; see section 5 of Kaplan and Sun (2017).

A plug-in (data-dependent) version of the optimal bandwidth from Kaplan and Sun (2017) is implemented in `sivqr` as follows. First, as in their proposition 2, the bandwidth is simplified by assuming $v \equiv y - \mathbf{x}'\beta(\tau)$ is independent of the full vector of instruments. Second, the smoothed indicator function is piecewise linear, as mentioned in section 5.2. In the notation of Kaplan and Sun (2017),

$$G(v) = \max\{0, \min\{1, (v + 1)/2\}\} \tag{8}$$

Third, the smallest amount of underlying smoothness in the data-generating process is assumed, $r = 2$ (see Kaplan and Sun 2017, ass. 3).

With the chosen $G(\cdot)$ in (8) and $r = 2$, simplifying the optimal bandwidth h^* from

proposition 2 of Kaplan and Sun (2017) with $v \perp z$,

$$\begin{aligned}
1 - \int_{-1}^1 [G(v)]^2 dv &= 1 - \int_{-1}^1 (v+1)^2/4 dv = 1/3 \\
\left[\int_{-1}^1 G'(v)v^r dv \right]^2 &= \left[\int_{-1}^1 v^2/2 dv \right]^2 = 1/9 \\
h^* &= \left(\frac{(r!)^2 \left[1 - \int_{-1}^1 [G(v)]^2 dv \right] f_v(0) d}{2r \left[\int_{-1}^1 G'(v)v^r dv f_v^{(r-1)}(0) \right]^2 n} \right)^{1/(2r-1)} \\
&= n^{-1/3} \left(d \frac{4(1/3)f_v(0)}{4(1/9)[f'_v(0)]^2} \right)^{1/3} = n^{-1/3} (3df_v(0)/[f'_v(0)]^2)^{1/3}
\end{aligned}$$

where $f_v(\cdot)$ is the probability density function of v and $f''_v(\cdot)$ its second derivative, and d is the dimension (length) of $\beta(\tau)$.

The `sivqr` plug-in bandwidth is the minimum of a few different values. By taking the minimum (instead of average or maximum), mistakes tend to be in the direction of undersmoothing compared to the “optimal” smoothing that minimizes mean squared error. This results in lower bias (but higher variance) than is optimal, because smoothing reduces variance at the cost of increased bias. The minimum is taken among a nonparametrically estimated version of h^* , a Gaussian reference version, and the Silverman rule of thumb bandwidth (Silverman 1986, §3.4.2) for estimating the density of v , as suggested by Fernandes et al. (2020) for non-IV smoothed QR. The former two are detailed further below.

After finally using the plug-in bandwidth to compute the smoothed IVQR estimator, the residuals \hat{v}_i are then recomputed from the new $\hat{\beta}(\tau)$, leading to an updated plug-in bandwidth, and the resulting IVQR estimate is reported by `sivqr`.

Nonparametrically estimated bandwidth

One approach is to replace $f_v(0)$ and $f'_v(0)$ in h^* by nonparametric kernel estimates. Since v is unobserved, an initial estimate $\hat{\beta}(\tau)$ must be used to construct residuals

$$\hat{v}_i = y_i - \mathbf{x}'_i \hat{\beta}(\tau)$$

For $f(0)$ (dropping the v subscript for now), given the \hat{v}_i , the usual kernel density estimator is

$$\frac{1}{ns} \sum_{i=1}^n K(-\hat{v}_i/s) \quad (9)$$

where s is the kernel bandwidth (not to be confused with the IVQR smoothing bandwidth h) and the kernel function $K(\cdot)$ is chosen to be the Gaussian kernel. For s , a pointwise version of Silverman’s rule of thumb (Silverman 1986, §3.4.2) is used because

interest is in $f(0)$, not the full function $f(\cdot)$, so it is better to minimize the pointwise mean squared error than the integrated mean squared error. The (asymptotic) pointwise mean squared error and optimal bandwidth can be found in DasGupta (2008, Ch. 32, p. 536), for example:

$$s^* = n^{-1/5} \left(\frac{f(0)}{\{f''(0)\}^2} \right)^{1/5} \left(\frac{\int_{\mathbb{R}} \{K(v)\}^2 dv}{\{\int_{\mathbb{R}} K(v)v^2 dv\}^2} \right)^{1/5}$$

Following the Gaussian reference approach of Silverman (1986, §3.4.2), assume $f(\cdot)$ is the density of a $N(\mu, \sigma^2)$ distribution; that is, $f(x) = \phi((x - \mu)/\sigma)/\sigma$, and the corresponding cumulative distribution function is $F(x) = \Phi((x - \mu)/\sigma)$, where $\phi(\cdot)$ and $\Phi(\cdot)$ are respectively the density and distribution functions of the standard normal distribution. Given the restriction that the τ -quantile of v is zero,

$$\tau = \Phi((0 - \mu)/\sigma) \implies -\mu/\sigma = \Phi^{-1}(\tau)$$

Thus, the density at zero is

$$f(0) = \phi((0 - \mu)/\sigma)/\sigma = \phi(\Phi^{-1}(\tau))/\sigma \quad (10)$$

Similarly, assuming normality allows us to express $\hat{f}''(0)$ in terms of $\hat{\sigma}$. The second derivative of the $N(\mu, \sigma^2)$ density evaluated at zero is³

$$\begin{aligned} f''(0) &= f(0)\{(0 - \mu)^2/\sigma^4 - 1/\sigma^2\} = \overbrace{\phi(\Phi^{-1}(\tau))\sigma^{-1}}^{=f(0) \text{ by (10)}} \sigma^{-2}[\{\Phi^{-1}(\tau)\}^2 - 1] \\ &= \sigma^{-3}\phi(\Phi^{-1}(\tau))[\{\Phi^{-1}(\tau)\}^2 - 1] \end{aligned}$$

Thus,

$$\frac{f(0)}{\{f''(0)\}^2} = \frac{\phi(\Phi^{-1}(\tau))\sigma^{-1}}{\sigma^{-6}\{\phi(\Phi^{-1}(\tau))\}^2[\{\Phi^{-1}(\tau)\}^2 - 1]^2} = \frac{\sigma^5}{\{\phi(\Phi^{-1}(\tau))\}[\{\Phi^{-1}(\tau)\}^2 - 1]^2}$$

Also, for the Gaussian kernel,⁴

$$\begin{aligned} \int_{\mathbb{R}} \{K(v)\}^2 dv &= \frac{1}{2\pi} \int_{\mathbb{R}} \exp(-v^2) dv = 1/(2\sqrt{\pi}), \\ \int_{\mathbb{R}} K(v)v^2 dv &= 1 \end{aligned} \quad (11)$$

3. D[PDF[NormalDistribution[[Mu], \[Sigma]], x], {x,2}] at <http://www.wolframalpha.com> for example; then plug in $x = 0$ and $-\mu/\sigma = \Phi^{-1}(\tau)$.

4. Unfortunately, the value in table 32.1 in DasGupta (2008, p. 537) is incorrectly stated as $1/\sqrt{2\pi}$.

Finally, plugging everything into the formula for s^* ,

$$\begin{aligned} s^* &= n^{-1/5} \left(\frac{f(0)}{\{f''(0)\}^2} \right)^{1/5} \left(\frac{\int_{\mathbb{R}} \{K(v)\}^2 dv}{\{\int_{\mathbb{R}} K(v)v^2 dv\}^2} \right)^{1/5} \\ &= n^{-1/5} \sigma [\phi(\Phi^{-1}(\tau))\{\{\Phi^{-1}(\tau)\}^2 - 1\}^2]^{-1/5} \left(\frac{1/(2\sqrt{\pi})}{1^2} \right)^{1/5} \\ &= 0.776n^{-1/5} \sigma [\phi(\Phi^{-1}(\tau))\{\{\Phi^{-1}(\tau)\}^2 - 1\}^2]^{-1/5} \end{aligned}$$

Following Silverman (1986, eq. (3.30)), to get a feasible bandwidth \hat{s} , the unknown σ is replaced by either the sample standard deviation of the \hat{v}_i or their sample interquartile range divided by 1.349 (the standard normal interquartile range), whichever is smaller. This \hat{s} is then used to compute $\hat{f}_v(0)$ using (9).

For the density derivative $f'_v(0)$ (here simply $f'(0)$), the approach is the same: use a nonparametric kernel estimator with a Silverman-type bandwidth. The usual estimator is

$$\frac{1}{nb^2} \sum_{i=1}^n K'(-\hat{v}_i/b) \quad (12)$$

where now b is the bandwidth (to avoid confusion with h and s). The asymptotic mean squared error for this estimator at the point zero is from Wand and Jones (1994),⁵

$$n^{-1}b^{-3}f(0) \int_{\mathbb{R}} \{K'(v)\}^2 dv + (1/4)b^4\{f'''(0)\}^2 \left\{ \int_{\mathbb{R}} K(v)v^2 dv \right\}^2$$

where $\int_{\mathbb{R}} K(v)v^2 dv = 1$ as in (11). Solving the first-order condition (setting the derivative with respect to b equal to zero), this leads to the optimal bandwidth

$$b^* = n^{-1/7} \left(\frac{3f(0) \int_{\mathbb{R}} \{K'(v)\}^2 dv}{\{f'''(0)\}^2} \right)^{1/7}$$

Plugging in the standard normal density for $K(\cdot)$,

$$\int_{\mathbb{R}} \{K'(v)\}^2 dv = 1/(4\sqrt{\pi})$$

Plugging in $f(0)$ from (10) and⁶

$$f'''(0) = \sigma^{-3}f(0)\Phi^{-1}(\tau) [3 - \{\Phi^{-1}(\tau)\}^2]$$

5. Unfortunately, their (2.34) in section 2.12 has a typo, but the correct version of (2.34) can be derived from the bias and variance expressions in their exercise 2.6(a) on pages 52–53.

6. D[PDF[NormalDistribution[μ , σ], x], { x , 3}] at <http://www.wolframalpha.com> for example; then plug in $x = 0$ and $-\mu/\sigma = \Phi^{-1}(\tau)$.

yields

$$\begin{aligned} b^* &= n^{-1/7} \left(\frac{3f(0)/(4\sqrt{\pi})}{\sigma^{-6}\{f(0)\}^2\{\Phi^{-1}(\tau)\}^2[3 - \{\Phi^{-1}(\tau)\}^2]^2} \right)^{1/7} \\ &= n^{-1/7} \left(\frac{3/(4\sqrt{\pi})}{\sigma^{-6}\sigma^{-1}\phi(\Phi^{-1}(\tau))\{\Phi^{-1}(\tau)\}^2[3 - \{\Phi^{-1}(\tau)\}^2]^2} \right)^{1/7} \\ &= n^{-1/7}\sigma \left(\frac{0.423}{\phi(\Phi^{-1}(\tau))\{\Phi^{-1}(\tau)\}^2[3 - \{\Phi^{-1}(\tau)\}^2]^2} \right)^{1/7} \end{aligned}$$

Again, to get a feasible bandwidth \hat{b} , σ is replaced by the sample standard deviation or interquartile range divided by 1.349, whichever is smaller. This \hat{b} is then used to compute $\hat{f}'_v(0)$ using (12).

Finally, $\hat{f}_v(0)$ and $\hat{f}'_v(0)$ are plugged into the expression for h^* to get the feasible plug-in bandwidth \hat{h} .

Gaussian reference bandwidth

Here, the Gaussian reference approach is used directly for h^* , to simplify the density and density derivative of v .

From (10), assuming normality of v yields $f(0) = \sigma^{-1}\phi(\Phi^{-1}(\tau))$.

Assuming normality, the first derivative of the density of v evaluated at zero is⁷

$$\begin{aligned} f'(0) &= -\sigma^{-1}(-\mu/\sigma)f(0) \\ f(0)/\{f'(0)\}^2 &= \frac{1}{\sigma^{-2}\{\Phi^{-1}(\tau)\}^2\sigma^{-1}\phi(\Phi^{-1}(\tau))} = \frac{\sigma^3}{\{\Phi^{-1}(\tau)\}^2\phi(\Phi^{-1}(\tau))} \end{aligned}$$

Plugging this into the expression for h^* yields

$$h^* = n^{-1/3}\sigma \left(\frac{3d}{\{\Phi^{-1}(\tau)\}^2\phi(\Phi^{-1}(\tau))} \right)^{1/3}$$

and as usual σ is replaced by the smaller of the sample standard deviation of the \hat{v}_i or the sample interquartile range of the \hat{v}_i divided by 1.349.

5.5 Bayesian bootstrap

Standard errors are computed by Bayesian bootstrap (Rubin 1981), assuming i.i.d. sampling. (With non-i.i.d. sampling, `bootstrap` can be used; see [R] `bootstrap`.) This is a

⁷ D[PDF[NormalDistribution[μ , σ], x], { x , 1}] at <http://www.wolframalpha.com> for example; then plug in $x = 0$ and $-\mu/\sigma = \Phi^{-1}(\tau)$.

particular type of frequentist exchangeable bootstrap (for example, van der Vaart and Wellner 1996, ex. 3.6.9, p. 354) that also has a nonparametric Bayesian interpretation (for example, Chamberlain and Imbens 2003, who advocate for Bayesian bootstrap and show IV and QR applications). In each replication, weights are set as $w_i = \xi_i/\bar{\xi}$ where the ξ_i are i.i.d. standard exponential random variables and $\bar{\xi}$ is their average; equivalently, the vector $(w_1, \dots, w_n)/n$ follows a Dirichlet distribution with every parameter equal to one. Using these w_i , the estimator is computed by solving (6). This is repeated many times; the reported standard error is the standard deviation of the different $\hat{\beta}$.

6 Conclusion

The `sivqr` command offers smoothed estimation of the instrumental variables quantile regression model of Chernozhukov and Hansen (2005). The smoothing improves computation and accuracy, helping `sivqr` to overcome several limitations of `ivqreg` and `ivqreg2`, while complementing commands for alternative quantile models with endogeneity (`cqiv` and `ivqte`). The `sivqr` command can help Stata users reliably estimate heterogeneous effects across a variety of settings and datasets.

7 Acknowledgements

There would be no `sivqr` Stata command without the work of Yixiao Sun on Kaplan and Sun (2017). Thanks to Xin Liu for research assistance.

8 References

- Chamberlain, G., and G. W. Imbens. 2003. Nonparametric Applications of Bayesian Inference. *Journal of Business & Economic Statistics* 21(1): 12–18. <https://www.jstor.org/stable/1392346>.
- Cheng, X., Z. Liao, and R. Shi. 2019. On Uniform Asymptotic Risk of Averaging GMM Estimators. *Quantitative Economics* 10(3): 931–979. <https://doi.org/10.3982/QE711>.
- Chernozhukov, V., I. Fernández-Val, S. Han, and A. Kowalski. 2019. Censored quantile instrumental-variable estimation with Stata. *Stata Journal* 19(4): 768–781. <https://doi.org/10.1177/1536867X19893615>.
- Chernozhukov, V., I. Fernández-Val, and A. E. Kowalski. 2015. Quantile regression with censoring and endogeneity. *Journal of Econometrics* 186(1): 201–221. <https://doi.org/10.1016/j.jeconom.2014.06.017>.
- Chernozhukov, V., and C. Hansen. 2004. The Effects of 401(k) Participation on the Wealth Distribution: An Instrumental Quantile Regression Analysis. *Review of Economics and Statistics* 86(3): 735–751. <https://doi.org/10.1162/0034653041811734>.

- . 2005. An IV Model of Quantile Treatment Effects. *Econometrica* 73(1): 245–261. <https://www.jstor.org/stable/3598944>.
- . 2006. Instrumental quantile regression inference for structural and treatment effect models. *Journal of Econometrics* 132(2): 491–525. <https://doi.org/10.1016/j.jeconom.2005.02.009>.
- . 2008. Instrumental Variable Quantile Regression: A Robust Inference Approach. *Journal of Econometrics* 142(1): 379–398.
- Chernozhukov, V., C. Hansen, and M. Jansson. 2009. Finite sample inference for quantile regression models. *Journal of Econometrics* 152(2): 93–103. <https://doi.org/10.1016/j.jeconom.2009.01.004>.
- Chernozhukov, V., C. Hansen, and K. Wüthrich. 2017. Instrumental Variable Quantile Regression. In *Handbook of Quantile Regression*, ed. R. Koenker, V. Chernozhukov, X. He, and L. Peng, chap. 9, 119–144. CRC/Chapman-Hall. <https://www.routledgehandbooks.com/doi/10.1201/9781315120256>.
- DasGupta, A. 2008. *Asymptotic Theory of Statistics and Probability*. New York: Springer. <https://doi.org/10.1007/978-0-387-75971-5>.
- Fernandes, M., E. Guerre, and E. Horta. 2020. Smoothing Quantile Regressions. *Journal of Business & Economic Statistics* XXX(XXX): XXX. <https://doi.org/10.1080/07350015.2019.1660177>.
- Frölich, M., and B. Melly. 2010. Estimation of quantile treatment effects with Stata. *Stata Journal* 10(3): 423–457. <https://www.stata-journal.com/article.html?article=st0203>.
- . 2013. Unconditional quantile treatment effects under endogeneity. *Journal of Business & Economic Statistics* 31(3): 346–357.
- Huber, P. J. 1964. Robust Estimation of a Location Parameter. *Annals of Mathematical Statistics* 35(1): 73–101. <https://www.jstor.org/stable/2238020>.
- Imbens, G. W., and J. D. Angrist. 1994. Identification and Estimation of Local Average Treatment Effects. *Econometrica* 62(2): 467–475. <https://www.jstor.org/stable/2951620>.
- Imbens, G. W., and W. K. Newey. 2009. Identification and Estimation of Triangular Simultaneous Equations Models Without Additivity. *Econometrica* 77(5): 1481–1512. <https://www.jstor.org/stable/25621368>.
- Kaplan, D. M. 2020. Distributional and Nonparametric Econometrics. Textbook draft. <https://faculty.missouri.edu/kaplandm/teach.html>.
- Kaplan, D. M., and X. Liu. 2021. k -class IV quantile regression. Working paper available at <https://faculty.missouri.edu/kaplandm>.

- Kaplan, D. M., and Y. Sun. 2017. Smoothed Estimating Equations for Instrumental Variables Quantile Regression. *Econometric Theory* 33(1): 105–157. <https://doi.org/10.1017/S0266466615000407>.
- Lee, S. 2007. Endogeneity in quantile regression models: A control function approach. *Journal of Econometrics* 141(2): 1131–1158. <https://doi.org/10.1016/j.jeconom.2007.01.014>.
- Liu, X. 2019. Averaging estimation for instrumental variables quantile regression. Working paper available at <https://ideas.repec.org/p/umc/wpaper/1907.html>.
- Machado, J. A. F., and J. M. C. Santos Silva. 2019. Quantiles via moments. *Journal of Econometrics* 213(1): 145–173. <https://doi.org/10.1016/j.jeconom.2019.04.009>.
- Melly, B., and K. Wüthrich. 2017. Local Quantile Treatment Effects. In *Handbook of Quantile Regression*, ed. R. Koenker, V. Chernozhukov, X. He, and L. Peng, chap. 10, 145–164. CRC/Chapman-Hall. <https://www.routledgehandbooks.com/doi/10.1201/9781315120256>.
- Rubin, D. B. 1981. The Bayesian Bootstrap. *Annals of Statistics* 9(1): 130–134. <https://projecteuclid.org/euclid.aos/1176345338>.
- Silverman, B. W. 1986. *Density Estimation for Statistics and Data Analysis*, vol. 26 of *Monographs on Statistics and Applied Probability*. London: Chapman & Hall/CRC.
- van der Vaart, A. W., and J. A. Wellner. 1996. *Weak Convergence and Empirical Processes: With Applications to Statistics*. Springer Series in Statistics, New York: Springer. <https://doi.org/10.1007/978-1-4757-2545-2>.
- Wand, M. P., and M. C. Jones. 1994. *Kernel Smoothing*, vol. 90 of *Monographs on Statistics and Applied Probability*. Chapman & Hall/CRC. <https://www.worldcat.org/title/kernel-smoothing/oclc/711630787/editions>.
- Wüthrich, K. 2020. A Comparison of Two Quantile Models With Endogeneity. *Journal of Business & Economic Statistics* 38(2): 443–456. <https://doi.org/10.1080/07350015.2018.1514307>.

About the author

David M. Kaplan is an associate professor in the Department of Economics at the University of Missouri. His primary research interest is econometric methodology. In particular, he enjoys creating and advancing methods for understanding changes and treatment effects on entire distributions (instead of just averages).