Smoothed instrumental variables 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 quantile regression model introduced by Chernozhukov and Hansen (2005, *Econometrica* 73: 245–261). The `sivqr` command offers several advantages over the existing `ivqreg` and `ivqreg2` commands for estimating this instrumental variables quantile regression model, which complements the alternative "triangular model" behind `cqiv` and the "local quantile treatment effect" model of `ivqte`. Computationally, `sivqr` implements the smoothed estimator of Kaplan and Sun (2017, *Econometric Theory* 33: 105–157), who show that smoothing improves both computation time and statistical accuracy. Standard errors are computed analytically or by Bayesian bootstrap; for nonindependent and identically distributed sampling, `sivqr` is compatible with `bootstrap`. I discuss syntax and the underlying methodology, and I compare `sivqr` with other commands in an example.

Keywords: st0676, 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 across individuals (or firms, countries, etc.) because of unobserved variables (rather than interactions among observed variables), but the existing IVQR commands, `ivqreg` (Kwak 2010a) and `ivqreg2` (Machado and Santos Silva 2018), have several limitations. First, `ivqreg` (implementing the estimator of Chernozhukov and Hansen [2006]) allows only 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 the estimator of Machado and Santos Silva [2019]) imposes a location–scale model that can help if correct but that can result in inconsistency if misspecified. Machado and Santos Silva (2019, 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 noncrossing of structural quantile functions. Ideally, `sivqr` and `ivqreg2` could be combined using a "model averaging" approach as in Cheng, Liao, and Shi (2019) or Liu (2019); meanwhile, simply running both may be insightful. Third, the existing commands' computation time can be long. Fourth, `ivqreg` and `ivqreg2` share

some minor inconveniences, like not supporting syntax for factor variables and interaction terms. Fifth, `ivqreg` is seemingly not actively supported and has additional issues; for example, the newest version 1.0.1 (02dec2010) fixes a problem from version 1.0.0 displaying coefficient labels when using an asterisk in the regressor list (like `xvar*`), but it introduces an error when there is overidentification (more excluded instruments than endogenous regressors).

To address these limitations, I introduce the `sivqr` command. The `sivqr` command allows models with multiple endogenous terms, supports convenient syntax like that for factor variables and interaction terms, and computes a consistent estimator of the IVQR parameters in a reasonable amount of time. The run time in the section 4.1 example is around 25 seconds for `sivqr`, as opposed to a few minutes (`ivqreg`) or over 20 minutes (`ivqreg2`). Although these differences become trivial on small datasets, where everything computes quickly, they become even more important when using `bootstrap` for cluster–robust standard errors, which requires computing the estimator many times.

The `sivqr` command implements the smoothed IVQR estimator of Kaplan and Sun (2017), using familiar syntax. Syntax is similar to the `ivregress` command (see [R] **ivregress**) but additionally specifies a quantile level. Likewise, syntax is similar to the `qreg` command (see [R] **qreg**) but additionally specifies instruments for the endogenous regressors. With nonindependent and identically distributed sampling, the `bootstrap` prefix can be used for standard errors; see [R] **bootstrap** and the example in section 4.

Other Stata commands that address endogeneity in quantile models are based on different models than that of Chernozhukov and Hansen (2005). 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). This conditional (`aai`) estimand and the IVQR estimand are different but closely related, as explained in section 10.5 of Melly and Wüthrich (2018) and detailed in Wüthrich (2020), both of whom compare the models more generally, too. See also Frölich and Melly (2013) for details. Practically, compared with `ivqte` with the `aai` option, the main advantage of `sivqr` is the ability to handle more than a single, binary endogenous regressor. The `cqiv` command (Chernozhukov et al. 2019, 2012) uses a control function estimator based on a triangular model (Lee 2007), also allowing for censoring (Chernozhukov, Fernández-Val, and Kowalski 2015). The estimand is the same as for IVQR (see section 2.1). Practically, compared with `cqiv`, the main advantages of `sivqr` are handling multiple or noncontinuous endogenous regressors, or both, as well as allowing simultaneity and reverse causality. For further comparison of the IVQR and triangular models, see section 9.2.5 of Chernozhukov, Hansen, and Wüthrich (2018).

Unfortunately, all these commands (including `sivqr`) 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, Hansen, and Jansson (2009), Chernozhukov, Hansen, and Wüthrich (2018, sec. 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), IVQR, and two-stage least squares (2SLS).

2 A gentle introduction to methodology

This section discusses methodology at a relatively nontechnical level (compared with 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. Because 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 his or her 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 (nonrandom) 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 cumulative distribution function (quantile function) of v, shifted by β_0; for example, with $\Phi(\cdot)$ as the standard normal cumulative distribution function, $v = \Phi^{-1}(u)$ 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 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 him or her 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, sec. 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 $\beta_0(\tau) + \beta_1(\tau)x_0$ because of 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 $v \equiv y - \beta_0 - \beta_1 x$ the standard IV moment conditions are

$$0 = E(v) = E(y - \beta_0 - \beta_1 x)$$
$$0 = E(zv) = E\{z(y - \beta_0 - \beta_1 x)\}$$

The IVQR moment conditions are

$$0 = E\left[\mathbf{1}\{y - \beta_0(\tau) - \beta_1(\tau)x \leq 0\} - \tau\right]$$
$$0 = E\left(z\left[\mathbf{1}\{y - \beta_0(\tau) - \beta_1(\tau)x \leq 0\} - \tau\right]\right) \tag{1}$$

where $\mathbf{1}\{\cdot\}$ is the indicator function defined as $\mathbf{1}\{\mathcal{A}\} = 1$ if $\mathcal{A}$ is true and otherwise $\mathbf{1}\{\mathcal{A}\} = 0$. 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 $y - \beta_0 - \beta_1 x$ (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 function that decreases continuously from 1 to 0 rather than discontinuously jumping from 1 to 0. The smoothed function differs from the indicator function only near the jump point; they are identical outside of a window whose width is controlled by a bandwidth (which asymptotically goes to zero). 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 IVQR model, and it provides standard errors.

Syntax, options, and stored results are now shown. The `by` prefix is allowed (see [D] **by**). After running, postestimation commands like `test` and `predict` (see [R] **test** and [R] **predict**) can be used as usual.

3.1 Syntax

`sivqr` *depvar* $\big[$*varlist*$_1$$\big]$ (*varlist*$_2$ = *varlist*$_{\mathrm{iv}}$) $\big[$*if*$\big]$ $\big[$*in*$\big]$ $\big[$*weight*$\big]$, <u>q</u>uantile(#)
   $\big[$<u>band</u>width(#) <u>l</u>evel(#) <u>r</u>eps(#) <u>logi</u>terations <u>nocon</u>stant seed(#)
   nodots <u>init</u>ial(*matname*) $\big]$

As in `ivregress`, *varlist*$_1$ has exogenous regressors (or control variables), *varlist*$_2$ has endogenous regressors, and *varlist*$_{\mathrm{iv}}$ has excluded instruments (exogenous variables that instrument for *varlist*$_2$).

`pweight`s, `fweight`s, and `iweight`s are allowed; see [U] **11.1.6 weight**.

3.2 Options

quantile(#) specifies the quantile level as in qreg (see [R] **qreg**): a real number strictly
between 0 and 1 or, alternatively, a number between 1 and 100 interpreted as a per-
centile. 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.
quantile() is required.

bandwidth(#) manually specifies the desired smoothing bandwidth. Leaving it unspec-
ified invokes a plugin bandwidth based on Kaplan and Sun (2017) as the default.
If the desired bandwidth is too small to find a numerical solution, then it is in-
creased until a solution is found. For example, specifying bandwidth(0) requests
the smallest bandwidth that is computationally feasible. Generally, the default plu-
gin bandwidth is recommended; compared with an arbitrarily chosen bandwidth,
the plugin usually produces a more accurate estimator (lower mean squared error).
However, for a final analysis, it is prudent to also try a smaller bandwidth like
bandwidth(0) as a sensitivity check (results should be qualitatively similar, though
not identical). A smaller bandwidth tends to yield an estimator with larger variance
but smaller (closer to zero) bias.

level(#); see [R] **Estimation options**.

reps(#) specifies the number of bootstrap replications for estimating the variance–
covariance matrix and standard errors. reps(0), which is the default, reports hetero-
skedasticity-robust analytic standard errors. Otherwise, the Bayesian bootstrap of
Rubin (1981) is used; it is a valid frequentist bootstrap that also has a nonparametric
Bayesian interpretation (see section 5.6).

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 helps only debugging and troubleshooting.

noconstant; see [R] **Estimation options**.

seed(#) sets the random-number seed to make results replicable. The default is
seed(112358). The current seed is restored at the end of execution.

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

initial(*matname*) sets the initial coefficient values for the numerical search, using the
values stored in the matrix (row vector) named *matname*. By default, qreg is used
to generate initial values. This option is particularly helpful when fitting the same
model across many quantile levels; examples are given in sivqr_examples_init.do.

3.3 Stored results

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

Scalars
e(N)	number of observations
e(reps)	number of bootstrap replications
e(bwidth)	smoothing bandwidth used
e(bwidth_req)	smoothing bandwidth requested (or plugin value)
e(bwidth_max)	maximum plugin bandwidth
e(q)	quantile level requested

Macros
e(cmd)	sivqr
e(depvar)	name of dependent variable
e(wtype)	weight type
e(wexp)	weight expression
e(title)	title in estimation output
e(vcetype)	Robust or Bootstrap
e(properties)	b V or else only b if reps(0)
e(instd)	instrumented variables
e(insts)	instruments
e(constant)	noconstant if specified
e(exogr)	exogenous regressors

Matrices
e(b)	coefficient vector
e(V)	variance–covariance matrix of the estimators, unless reps(0)

Functions
e(sample)	marks estimation sample

Almost all the above are the same as for the familiar commands `ivregress` and `qreg` (or `bsqreg`). The only exceptions are for the smoothing bandwidth. By default, the plugin bandwidth is computed and returned in `e(bwidth_req)`. More specifically, three plugin bandwidths are computed (see section 5.4), of which `e(bwidth_req)` is the minimum and `e(bwidth_max)` is the maximum. 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.

If you choose to use the default plugin bandwidth and if `e(bwidth)` is greater than `e(bwidth_max)`, then there may be deeper problems such as weak instruments. Although there is currently no weak instrument test specific to IVQR, it would be helpful to run a weak instrument test for 2SLS in this case, such as running `ivregress` and then `estat firststage`. If `e(bwidth)` is above `e(bwidth_req)` but less than `e(bwidth_max)`, then it may be simply that the plugin bandwidth was chosen too conservatively (see section 5.4), but it may still be worth running `ivregress` and `estat firststage`.

sivqr also stores the following in r():

Scalar
  r(level)   confidence level

Matrices
  r(table)   table of results

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

4 Example: Example 4 from ivregress

All results (besides run times) can be replicated with the provided sivqr_examples.do file. I ran them in Stata/MP 17.0 with 2 cores on Windows 10, on a few-years-old standard-issue Dell desktop computer (Intel i5-8500 3GHz processor, 8GB RAM). Besides using version 1.1.0 (22jan2022) of sivqr, I used version 2.6 (29sep2021) of ivqreg2 from Statistical Software Components Archive (Machado and Santos Silva 2018) and version 1.0.0 (28jul2010) of ivqreg (Kwak 2010a). The most recent version 1.0.1 (02dec2010) from Kwak (2010b) unfortunately gives an error when trying to use an overidentified model (as in section 4.1), "st_numscalar(): 3204 matrix found where scalar required." Other changes in version 1.0.1 are both positive and negative; when I modify the section 4.1 model to have only union as an excluded instrument, then version 1.0.1 reports standard errors, whereas version 1.0.0 does not (instead reporting "Warning: variance matrix is nonsymmetric or highly singular"), although the version 1.0.1 run time is over 10 times longer.

The primary focus is comparison with ivqreg and ivqreg2, in terms of estimates, speed, and capabilities, including the following:

1. sivqr with a small bandwidth produces estimates very similar to ivqreg;

2. ivqreg2 can differ from ivqreg and sivqr, though is qualitatively similar;

3. ivqreg2 can be significantly slower than sivqr.

As noted earlier, ivqreg does not allow multiple endogenous regressors, which precludes even a quadratic function of the endogenous regressor, and the factorial operator ## is supported only by sivqr.

4.1 Main comparison

The following revisits example 4 in [R] **ivregress**. Briefly, it hopes to fit 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 differences across quantile levels, the IVQR coefficient estimates and signs should still be similar overall, especially around the median.

```
. webuse nlswork
(National Longitudinal Survey of Young Women, 14-24 years old in 1968)

. ivregress 2sls ln_wage c.age##c.age birth_yr grade
> (tenure = union wks_work 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	Coefficient	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, we use the `bootstrap` prefix after using `xtset` appropriately.[1] With `bootstrap`, it would only waste time (without changing the reported estimates or standard errors) to specify the `reps()` option (with a nonzero number) for `sivqr`. At the median, the `sivqr` coefficient estimates (other than the intercept) as well as the corresponding cluster–robust standard errors are similar to those from `ivregress 2sls`.

---

1. For example, see https://www.stata.com/support/faqs/statistics/bootstrap-with-panel-data/.

```
. generate newid = idcode

. xtset newid year
Panel variable: newid (unbalanced)
 Time variable: year, 68 to 88, but with gaps
         Delta: 1 unit

. bootstrap, reps(100) cluster(idcode) idcluster(newid) seed(112358): sivqr
> ln_wage c.age##c.age birth_yr grade (tenure = union wks_work msp),
> quantile(0.50)
(running sivqr on estimation sample)

Bootstrap replications (100)
─────┼─── 1 ───┼── 2 ───┼── 3 ───┼── 4 ───┼── 5
.................................................      50
.................................................     100

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)
```

	Observed	Bootstrap			Normal-based	
ln_wage	coefficient	std. err.	z	P>|z|	[95% conf. interval]	
tenure	.1076941	.0046079	23.37	0.000	.0986627	.1167255
age	.0060803	.0073372	0.83	0.407	-.0083004	.020461
c.age#c.age	-.0003585	.0001212	-2.96	0.003	-.000596	-.0001209
birth_yr	-.011967	.0021773	-5.50	0.000	-.0162344	-.0076997
grade	.065723	.0030378	21.63	0.000	.059769	.071677
_cons	1.25539	.1643566	7.64	0.000	.9332567	1.577523

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

The following code produces the results in tables 1 and 2, besides the cluster–bootstrap standard errors computed above (that are repeated for other quantiles). Compared with the provided do-file, below I omit various `timer` and matrix storage commands (and output). I run `sivqr` with both the default plugin bandwidth as well as `bandwidth(0)`, which tries to find the smallest computationally feasible bandwidth. The squared age variable is generated manually because `ivqreg` and `ivqreg2` do not support the factorial operator `##`.

```
. generate agesq = age^2
(24 missing values generated)

. forvalues i = 1/3 {
  2. local q = `i'/4
  3. sivqr ln_wage age agesq birth_yr grade (tenure = union wks_work msp),
> quantile(`q')
  4. sivqr ln_wage age agesq birth_yr grade (tenure = union wks_work msp),
> quantile(`q') bandwidth(0)
  5. ivqreg ln_wage age agesq birth_yr grade (tenure = union wks_work msp),
> q(`q') robust
  6. }
  (output omitted)

. ivqreg2 ln_wage tenure age agesq birth_yr grade, quantile(0.25 0.5 0.75)
> instruments(union wks_work msp age agesq birth_yr grade)
  (output omitted)
```

Table 1. Estimates of `tenure` coefficient; clustered standard errors

Quantile	sivqr	sivqr, b(0)	ivqreg	ivqreg2
0.25	0.0865756	0.0860257	0.0861186	0.0933732
	(0.0031621)			
0.50	0.1076941	0.1080343	0.1032661	0.1084423
	(0.0046079)			
0.75	0.1565857	0.1553029	0.1584188	0.1250514
	(0.0103318)			

NOTE: 2SLS estimate (std. err.) is 0.1061 (0.0045).

Table 1 shows the different `tenure` coefficient estimates at different quantiles, with the 2SLS results for comparison. To help gauge the magnitude of the differences, table 1 also shows cluster–robust standard errors from 2SLS and from using `bootstrap` with `sivqr`. (None of the commands produces cluster–robust standard errors by itself, so the default standard errors are not reported; they are generally too small here, as expected.)

At the median, all estimates are similar. All are well within one standard error of the 2SLS estimate. (The 2SLS estimate can differ greatly from median IVQR estimates, just as the sample mean and median can differ greatly, but they are often similar.) The cluster–bootstrap `sivqr` standard error is also similar to the 2SLS cluster–robust standard error.

Away from the median, `sivqr` (either bandwidth) and `ivqreg` remain similar to each other, but other differences arise. First, the IVQR estimates are all increasing with the quantile level, differing significantly from the 2SLS estimate both economically and statistically. Compared with the 0.25 quantile estimates, most of the 0.75 quantile estimates are nearly twice as large, suggesting substantial heterogeneity. This could be interpreted as individuals with higher levels of "ability" (and other unobserved wage determinants) having higher returns to job tenure.

Second, `ivqreg2` differs from the other IVQR estimators. This is especially seen at the 0.75 quantile: the `ivqreg2` estimate of the return to tenure is below the other three estimates by around 0.03 (three percentage points), which is around a three standard-error difference. It is possible that the location–scale model of `ivqreg2` is correctly specified and that this difference is only due to larger estimation error of `sivqr` and `ivqreg`, but given the consistency of both `sivqr` and `ivqreg` along with the difference being three standard errors, it seems likely that at least part of the difference is explained by misspecification of the `ivqreg2` location–scale model.

As a side note, these `ivqreg2` results were using the regressor order that gave the most economically plausible estimates. If we keep everything the same but with `tenure` last in the regressor list, the estimates of the `tenure` coefficient become significantly negative, and two warning messages appear: "WARNING: .03758389% of the fitted

values of the scale function are not positive" and "Warning: variance matrix is nonsymmetric or highly singular" (with missing standard errors).

Table 2. Runtime (seconds) of IVQR commands on `nlswork` data

Quantile	bootstrap:sivqr	sivqr	sivqr, b(0)	ivqreg	ivqreg2
0.25	861	16	98	130	↑
0.50	500	5	68	68	1304
0.75	1026	4	47	153	↓

Table 2 shows the run time of the different commands at different quantiles. The fastest is `sivqr` with the default plugin bandwidth and analytic standard errors, taking around 25 seconds to compute at all 3 quantiles. Using `bandwidth(0)` is significantly slower (a few minutes total, instead of seconds) because it requires many more calls to the numerical solver, trying to find the very smallest usable bandwidth. The `ivqreg` time is even slower than `bandwidth(0)`, also taking multiple minutes total. Naturally, running a cluster bootstrap with 100 replications takes even longer, even using the fast `sivqr` command with plugin bandwidth, taking around 40 minutes total. Even without bootstrap, `ivqreg2` takes over 20 minutes, an order of magnitude more than the 25 seconds taken by `sivqr`.

4.2   More computation times

The provided do-file illustrates some additional important considerations about computation time. Most of the `sivqr` computation time is from calling the `solvenl()` 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 3 illustrates this by running `sivqr` over quantiles $0.10, 0.15, \ldots, 0.90$, using either the plugin bandwidth (always between 0.05 and 0.08) or `bandwidth(0)` (always between 0.00001 and 0.00012, except at the 0.9 quantile). First, although there are other factors, run time is generally longer in the tails: with the plugin bandwidth, run time is at least 8 seconds at quantile levels $\{0.10, 0.80, 0.85, 0.90\}$ but under 8 seconds in between. The pattern is less clear for `bandwidth(0)` because it depends not only on the run time per `solvenl()` call but also on the number of such calls. Second, the run time is longer with the smaller bandwidth at 16 out of 17 quantile levels.

Table 3. Runtime (seconds) across quantile levels, `nlswork` data

Quantile	plug-in	bandwidth(0)
0.10	16	99
0.15	3	51
0.20	6	49
0.25	5	88
0.30	2	82
0.35	2	54
0.40	3	46
0.45	4	25
0.50	4	61
0.55	4	29
0.60	3	27
0.65	2	29
0.70	5	38
0.75	4	44
0.80	13	36
0.85	53	44
0.90	8	43

Also, even with the same original dataset, some bootstrap datasets are naturally more difficult than others, so bootstrap run time can depend on the random seed. With `quantile(0.5)`, out of 100 runs with different seeds, run times for the built-in `sivqr` bootstrap with 20 replications ranged from 9 seconds to 30 seconds, with lower and upper quartiles 11 seconds and 16 seconds. Among 5 runs with the `bootstrap` prefix (also 20 replications), run times ranged from 79 to 90 seconds. (I conjecture this is much slower because it resamples a new dataset rather than merely reweighting the original dataset like the Bayesian bootstrap of `sivqr`, but I have not broken down the run time by function to investigate further.) However, it cannot be known beforehand which seeds are "faster" for a particular dataset, so using the default seed is recommended, which also avoids any appearance of trying to manipulate results.

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 shows how to compute analytic standard errors. Section 5.6 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 (IV). 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) because uppercase is reserved for matrices, but with some luck, random variables can be distinguished from their possible nonrandom realizations.

Additional notation is introduced as needed below.

5.2 Estimation

As a helpful reference point for intuition, recall the standard ("mean") IV model and estimator. The goal is to estimate the nonrandom parameter vector $\boldsymbol{\beta}$ in the structural model

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

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

$$\mathbf{0} = E(u \mid \mathbf{z}) \implies \mathbf{0} = E(\mathbf{z}u) = E\left\{\mathbf{z}(y - \mathbf{x}'\boldsymbol{\beta})\right\} \tag{2}$$

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 $\boldsymbol{\beta}$,

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

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

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

$$0 = Q_\tau(u \mid \mathbf{z}) \tag{3}$$

That is, conditional on any value of $\mathbf{z}$, the conditional distribution of u has its τ-quantile equal to 0. 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

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

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

$$\tau = E[\mathbf{1}\{y - \mathbf{x}'\boldsymbol{\beta} \leq 0\} \mid \mathbf{z}] \implies 0 = E[\mathbf{1}\{y - \mathbf{x}'\boldsymbol{\beta} \leq 0\} - \tau \mid \mathbf{z}]$$
$$\implies \mathbf{0} = E\left(\mathbf{z}\left[\mathbf{1}\{y - \mathbf{x}'\boldsymbol{\beta} \leq 0\} - \tau\right]\right) \tag{4}$$

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 $\boldsymbol{\beta}$ and replace the mean (expectation) with the sample average. Here the $\boldsymbol{\beta}$ is stuck inside the indicator function $\mathbf{1}\{\cdot\}$, 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 $\mathbf{1}\{\cdot\}$ with a very similar but continuously differentiable function allows the sample system of equations to be solved quickly by standard numerical methods. Specifically, replacing $\mathbf{1}\{v \leq 0\}$, $\widetilde{I}(v)$ 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 $v = 0$.[2] Adding a bandwidth h allows $\widetilde{I}(v/h)$ to decrease more steeply over $-h \leq v \leq h$. The estimator $\widehat{\boldsymbol{\beta}}$ solves the "smoothed estimating equations" (or "smoothed moment conditions")

$$\mathbf{0} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{z}_i \left[\widetilde{I}\left\{\left(y_i - \mathbf{x}_i'\widehat{\boldsymbol{\beta}}\right)/h\right\} - \tau\right] \tag{5}$$

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

$$\mathbf{0} = \frac{1}{n}\sum_{i=1}^{n} w_i \mathbf{z}_i \left[\widetilde{I}\left\{\left(y_i - \mathbf{x}_i'\widehat{\boldsymbol{\beta}}\right)/h\right\} - \tau\right] \tag{6}$$

Specifically, `sivqr` uses a piecewise linear $\widetilde{I}(\cdot)$ that connects the smoothed IVQR estimator to other estimators. Instead of jumping down from 1 to 0 discontinuously, $\widetilde{I}(\cdot)$ has $\widetilde{I}(v) = 1$ for $v \leq -1$ and $\widetilde{I}(v) = 0$ for $v \geq 1$, but transitions linearly with $\widetilde{I}(v) = (1 - v)/2$ for $-1 < v < 1$. Kaplan and Sun (2017, sec. 2.2, 111) show how this choice produces the Winsorized mean estimator of the type in Huber (1964, ex. (iii), 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 $\widetilde{I}(\cdot)$ is a linear function of the residuals $y_i - \mathbf{x}_i'\widehat{\boldsymbol{\beta}}$, 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

---

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

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 $\boldsymbol{\beta}$. This linear projection is motivated by the 2SLS 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).[3] Instead of assuming every individual (or firm, or school, etc.) has the same coefficients $\boldsymbol{\beta}$, imagine each individual has his or her 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:

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

An additional error term would be redundant; for example, if $y = b_0 + b_1 x + v$, 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 $\boldsymbol{\beta}(\cdot)$ applied to a scalar unobserved u:

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

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

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

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

---

3. See also chapter 6 of Kaplan (2021) for an introductory discussion.

5.4 Plugin bandwidth

This section applies only to the plugin bandwidth. If instead the `bandwidth()` option is manually specified, then none of the following applies.

Kaplan and Sun (2017, prop. 2, 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 plugin (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 $v \equiv y - \mathbf{x}'\boldsymbol{\beta}(\tau)$ 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 $v \perp \mathbf{z}$,

$$1 - \int_{-1}^{1} \{G(v)\}^2 \mathrm{d}v = 1 - \int_{-1}^{1} (v+1)^2/4 \mathrm{d}v = 1/3$$

$$\left(\int_{-1}^{1} G'(v)v^r \mathrm{d}v\right)^2 = \left(\int_{-1}^{1} v^2/2 \mathrm{d}v\right)^2 = 1/9$$

$$h^* = \left(\frac{(r!)^2 \left[1 - \int_{-1}^{1}\{G(v)\}^2 \mathrm{d}v\right] f_v(0)}{2r \left\{\int_{-1}^{1} G'(v)v^r \mathrm{d}v f_v^{(r-1)}(0)\right\}^2} \frac{d}{n}\right)^{1/(2r-1)}$$

$$= n^{-1/3} \left(d \frac{4(1/3)f_v(0)}{4(1/9)\{f'_v(0)\}^2}\right)^{1/3} = n^{-1/3} \left[3df_v(0)/\{f'_v(0)\}^2\right]^{1/3}$$

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

The `sivqr` plugin bandwidth is the minimum of a few different values. If one takes the minimum (instead of average or maximum), mistakes tend to be in the direction of undersmoothing compared with 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, sec. 3.4.2) for estimating the density of

v, as suggested by Fernandes, Guerre, and Horta (2021) for non-IV smoothed QR. The first two are detailed further below.

After one finally uses the plugin bandwidth to compute the smoothed IVQR estimator, the residuals $\widehat{v}_i$ are then recomputed from the new $\widehat{\boldsymbol{\beta}}(\tau)$, leading to an updated plugin bandwidth, and the resulting IVQR estimate is reported by `sivqr`.

5.4.1 Nonparametrically estimated bandwidth

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

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

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

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

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, sec. 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, 536), for example:

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

Following the Gaussian reference approach of Silverman (1986, sec. 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 0,

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

Thus, the density at 0 is

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

Similarly, assuming normality allows us to express $\widehat{f}''(0)$ in terms of $\widehat{\sigma}$. The second derivative of the $N(\mu, \sigma^2)$ density evaluated at 0 is[4]

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

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,[5]

$$\int_{\mathbb{R}}\{K(v)\}^2\mathrm{d}v = \frac{1}{2\pi}\int_{\mathbb{R}}\exp(-v^2)\mathrm{d}v = 1/(2\sqrt{\pi})$$
$$\int_{\mathbb{R}}K(v)v^2\mathrm{d}v = 1 \tag{11}$$

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

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

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

$$\frac{1}{nb^2}\sum_{i=1}^{n}K'\left(-\widehat{v}_i/b\right) \tag{12}$$

where now b is the bandwidth (to avoid confusion with h and s). The asymptotic mean squared error for this estimator at the point 0 is from Wand and Jones (1995),[6]

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

---

4. Enter `D[PDF[NormalDistribution[\[Mu], \[Sigma]], x], {x,2}]` at https://www.wolframalpha.com, for example; then, plug in $x = 0$ and $-\mu/\sigma = \Phi^{-1}(\tau)$.
5. Unfortunately, the value in table 32.1 in DasGupta (2008, 537) is incorrectly stated as $1/\sqrt{2\pi}$.
6. 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.

where $\int_{\mathbb{R}} K(v)v^2 \mathrm{d}v = 1$ as in (11). Solving the first-order condition (setting the derivative with respect to b equal to 0) leads to the optimal bandwidth

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

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

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

Plugging in $f(0)$ from (10) and[7]

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

yields

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

$$= n^{-1/7} \left(\frac{3/(4\sqrt{\pi})}{\sigma^{-6}\sigma^{-1}\phi\{\Phi^{-1}(\tau)\}\{\Phi^{-1}(\tau)\}^2 \left[3 - \{\Phi^{-1}(\tau)\}^2 \right]^2} \right)^{1/7}$$

$$= n^{-1/7}\sigma \left(\frac{0.423}{\phi\{\Phi^{-1}(\tau)\}\{\Phi^{-1}(\tau)\}^2 \left[3 - \{\Phi^{-1}(\tau)\}^2 \right]^2} \right)^{1/7}$$

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

Finally, $\widehat{f_v}(0)$ and $\widehat{f'_v}(0)$ are plugged into the expression for h^* to get the feasible plugin bandwidth $\widehat{h}$.

5.4.2   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 is[8]

$$f'(0) = -\sigma^{-1}(-\mu/\sigma)f(0)$$

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

---

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

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 $\widehat{v}_i$ or the sample interquartile range of the $\widehat{v}_i$ divided by 1.349.

5.5 Standard errors

By default or with `reps(0)`, analytic standard errors are computed as follows; otherwise, bootstrap is used as in section 5.6. The formulas here are special cases of (3.11) of Chernozhukov and Hansen (2006); (18), (23), and theorem 5 of de Castro et al. (2019); or theorem 7 of Kaplan and Sun (2017). For notational simplicity, I leave the dependence on τ implicit. Define an additive error term

$$\epsilon \equiv y - \mathbf{x}'\boldsymbol{\beta}$$

The first-order asymptotic distribution of the `sivqr` estimator is

$$\sqrt{n}\left(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}\right) \xrightarrow{d} \mathrm{N}\left\{\mathbf{0}, (\mathbf{J}'\mathbf{S}^{-1}\mathbf{J})^{-1}\right\}, \ \mathbf{J} \equiv E\left\{f_{\epsilon|\mathbf{z},\mathbf{x}}(0 \mid \mathbf{z}, \mathbf{x})\mathbf{z}\mathbf{x}'\right\}, \ \mathbf{S} \equiv \tau(1-\tau)E(\mathbf{z}\mathbf{z}')$$

where $f_{\epsilon|\mathbf{z},\mathbf{x}}(0 \mid \mathbf{z}, \mathbf{x})$ is the conditional probability density functions of ϵ given $(\mathbf{z}, \mathbf{x})$ evaluated at $\epsilon = 0$. The expression $\mathbf{J}^{-1}\mathbf{S}(\mathbf{J}')^{-1}$ is mathematically equivalent but requires inverting the asymmetric matrix $\mathbf{J}$ by itself. The estimated asymptotic covariance matrix is

$$\left(\widehat{\mathbf{J}}'\widehat{\mathbf{S}}^{-1}\widehat{\mathbf{J}}\right)^{-1}, \quad \widehat{\mathbf{S}} = \tau(1-\tau)\frac{1}{n}\sum_{i=1}^{n}\mathbf{z}_i\mathbf{z}_i', \quad \widehat{\mathbf{J}} = \frac{1}{nh}\sum_{i=1}^{n}\phi\left(\frac{\mathbf{y}_i - \mathbf{x}_i'\widehat{\boldsymbol{\beta}}}{h}\right)\mathbf{z}_i\mathbf{x}_i'$$

using a conventional kernel density estimator for $\widehat{\mathbf{J}}$ with a Gaussian kernel $\phi(\cdot)$ and bandwidth h being Silverman's rule of thumb (Silverman 1986, sec. 3.4.2), specifically $1.06n^{-1/5}$ times the minimum of either the sample standard deviation of the $\mathbf{y}_i - \mathbf{x}_i'\widehat{\boldsymbol{\beta}}$ or their sample interquartile range divided by 1.349. Alternatively, the bandwidth in (A.8) of Liu (2019) would also work well. Note $\widehat{\mathbf{J}}$ is a Powell (1986)-type estimator, similar to (3.14) of Chernozhukov and Hansen (2006) but here with a Gaussian instead of uniform kernel. The reported standard errors are then the square roots of the diagonals of the matrix $(\widehat{\mathbf{J}}'\widehat{\mathbf{S}}^{-1}\widehat{\mathbf{J}})^{-1}/n$.

5.6 Bayesian bootstrap

When `reps(#)` specifies multiple bootstrap replications, 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 particular type of frequentist exchangeable bootstrap (for example, van der Vaart and Wellner [1996, ex. 3.6.9,

354]) that also has a nonparametric Bayesian interpretation (for example, Chamberlain and Imbens [2003] 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 $(w_1, \ldots, w_n)/n$ follows a Dirichlet distribution with every parameter equal to 1. Using these w_i, one computes the estimator by solving (6). This is repeated many times; the reported standard error is the standard deviation of the different $\widehat{\boldsymbol{\beta}}$.

6 Conclusion

The `sivqr` command offers smoothed estimation of the IVQR 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 Acknowledgments

8 Programs and supplemental materials

To install a snapshot of the corresponding software files as they existed at the time of publication of this article, type

```
. net sj 22-2
. net install st0676      (to install program files, if available)
. net get st0676          (to install ancillary files, if available)
```

9 References

de Castro, L., A. F. Galvao, D. M. Kaplan, and X. Liu. 2019. Smoothed GMM for quantile models. *Journal of Econometrics* 213: 121–144. https://doi.org/10.1016/j.jeconom.2019.04.008.

Chamberlain, G., and G. W. Imbens. 2003. Nonparametric applications of Bayesian inference. *Journal of Business and Economic Statistics* 21: 12–18. https://doi.org/10.1198/073500102288618711.

Cheng, X., Z. Liao, and R. Shi. 2019. On uniform asymptotic risk of averaging GMM estimators. *Quantitative Economics* 10: 931–979. https://doi.org/10.3982/QE711.

Chernozhukov, V., I. Fernández-Val, S. Han, and A. Kowalski. 2012. cqiv: Stata module to perform censored quantile instrumental variables regression. Statistical Software Components S457478, Department of Economics, Boston College. https://ideas.repec.org/c/boc/bocode/s457478.html.

———. 2019. Censored quantile instrumental-variable estimation with Stata. *Stata Journal* 19: 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: 201–221. https://doi.org/10.1016/j.jeconom.2014.06.017.

Chernozhukov, V., and C. Hansen. 2005. An IV model of quantile treatment effects. *Econometrica* 73: 245–261. https://doi.org/10.1111/j.1468-0262.2005.00570.x.

———. 2006. Instrumental quantile regression inference for structural and treatment effect models. *Journal of Econometrics* 132: 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: 379–398. https://doi.org/10.1016/j.jeconom.2007.06.005.

Chernozhukov, V., C. Hansen, and M. Jansson. 2009. Finite sample inference for quantile regression models. *Journal of Econometrics* 152: 93–103. https://doi.org/10.1016/j.jeconom.2009.01.004.

Chernozhukov, V., C. Hansen, and K. Wüthrich. 2018. Instrumental variable quantile regression. In *Handbook of Quantile Regression*, ed. R. Koenker, V. Chernozhukov, X. He, and L. Peng, 119–143. Handbooks of Modern Statistical Methods, Boca Raton, FL: Chapman & Hall/CRC. https://doi.org/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. 2021. Smoothing quantile regressions. *Journal of Business and Economic Statistics* 39: 338–357. https://doi.org/10.1080/07350015.2019.1660177.

Frölich, M., and B. Melly. 2010. Estimation of quantile treatment effects with Stata. *Stata Journal* 10: 423–457. https://doi.org/10.1177/1536867X1001000309.

———. 2013. Unconditional quantile treatment effects under endogeneity. *Journal of Business and Economic Statistics* 31: 346–357. https://doi.org/10.1080/07350015.2013.803869.

Huber, P. J. 1964. Robust estimation of a location parameter. *Annals of Mathematical Statistics* 35: 73–101.

Imbens, G. W., and J. D. Angrist. 1994. Identification and estimation of local average treatment effects. *Econometrica* 62: 467–475. https://doi.org/10.2307/2951620.

Imbens, G. W., and W. K. Newey. 2009. Identification and estimation of triangular simultaneous equations models without additivity. *Econometrica* 77: 1481–1512. https://doi.org/10.3982/ECTA7108.

Kaplan, D. M. 2021. Distributional and Nonparametric Econometrics. Digital book. https://drive.google.com/file/d/1QtZNe7xtxpIsZxexEiwGzWg4B1czaAMj/view.

Kaplan, D. M., and X. Liu. 2021. *k*-class instrumental variables quantile regression. https://drive.google.com/file/d/10Glmp7gOHJsc-Kp2cztGYWDMM1OP5Nmq/view.

Kaplan, D. M., and Y. Sun. 2017. Smoothed estimating equations for instrumental variables quantile regression. *Econometric Theory* 33: 105–157. https://doi.org/10.1017/S0266466615000407.

Kwak, D. W. 2010a. ivqreg.ado (Stata code) version 1.0.0.

———. 2010b. ivqreg.ado (Stata code) version 1.0.1. https://sites.google.com/site/dwkwak/dataset-and-code.

Lee, S. 2007. Endogeneity in quantile regression models: A control function approach. *Journal of Econometrics* 141: 1131–1158. https://doi.org/10.1016/j.jeconom.2007.01.014.

Liu, X. 2019. Averaging estimation for instrumental variables quantile regression. Working Paper 1907, Department of Economics, University of Missouri. https://ideas.repec.org/p/umc/wpaper/1907.html.

Machado, J. A. F., and J. M. C. Santos Silva. 2018. ivqreg2: Stata module to provide structural quantile function estimation. Statistical Software Components S458571, Department of Economics, Boston College. https://ideas.repec.org/c/boc/bocode/s458571.html.

———. 2019. Quantiles via moments. *Journal of Econometrics* 213: 145–173. https://doi.org/10.1016/j.jeconom.2019.04.009.

Melly, B., and K. Wüthrich. 2018. Local quantile treatment effects. In *Handbook of Quantile Regression*, ed. R. Koenker, V. Chernozhukov, X. He, and L. Peng, 145–164. Handbooks of Modern Statistical Methods, Boca Raton, FL: Chapman & Hall/CRC. https://doi.org/10.1201/9781315120256.

Powell, J. L. 1986. Censored regression quantiles. *Journal of Econometrics* 32: 143–155. https://doi.org/10.1016/0304-4076(86)90016-3.

Rubin, D. B. 1981. The Bayesian bootstrap. *Annals of Statistics* 9: 130–134. https://doi.org/10.1214/aos/1176345338.

Silverman, B. W. 1986. *Density Estimation for Statistics and Data Analysis*. Boca Raton, FL: Chapman & Hall/CRC.

van der Vaart, A. W., and J. A. Wellner. 1996. *Weak Convergence and Empirical Processes: With Applications to Statistics*. New York: Springer. https://doi.org/10.1007/978-1-4757-2545-2.

Wand, M. P., and M. C. Jones. 1995. *Kernel Smoothing*. New York: Chapman & Hall/CRC.

Wüthrich, K. 2020. A comparison of two quantile models with endogeneity. *Journal of Business and Economic Statistics* 38: 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), also a focus of his free open-source textbook and his first community-contributed command, `distcomp`.