The Stata Journal (*yyyy*) *vv*, Number *ii*, pp. 1–29

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). This model complements the alternative models underlying the commands `cqiv`, `ivqreg2`, and `ivqte`, and the `sivqr` command offers advantages over the apocryphal `ivqreg` command. Computationally, `sivqr` implements the smoothed estimator of Kaplan and Sun (2017), who show the 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` and `svy bootstrap`. I discuss syntax and the underlying methodology. Simulation and empirical examples illustrate the new `sivqr` command.

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

1 Introduction

The new `sivqr` command implements the smoothed instrumental variables quantile regression (IVQR) estimator of Kaplan and Sun (2017) for the model of Chernozhukov and Hansen (2005). Usage is similar to the `ivregress` command (see [R] **ivregress**) but additionally specifying a quantile level. Likewise, usage is similar to the `qreg` command (see [R] **qreg**) but additionally specifying instruments for the endogenous regressors.

The previous command to estimate the Chernozhukov and Hansen (2005) IVQR model has limitations. The `ivqreg` command implements the estimator of Chernozhukov and Hansen (2006), which searches over a grid of possible endogenous coefficient values. This command allows only one endogenous term.[1] Even if only a single regressor is endogenous, this precludes nonlinear and/or interaction terms. Further, it appears `ivqreg` has not been actively supported for several years. In contrast, the smoothed estimation of the new `sivqr` command allows fast computation with many endogenous coefficients and is actively supported.

Other Stata commands that address endogeneity in "quantile regression" are based on different models. None is strictly "better"; they are valid under different assumptions. The `ivqreg2` command estimates the location–scale type IVQR model of Machado and

1. Even beyond this particular implementation, computation time scales poorly with the number of endogenous coefficients; even the original authors' Matlab code at https://voices.uchicago.edu/christianhansen/code-and-data/ only allows two endogenous coefficients.

 st0001

Santos Silva (2019). This model is the most similar to that of Chernozhukov and Hansen (2005), and generally it is "neither more nor less restrictive" (Machado and Santos Silva 2019, p. 152). Machado and Santos Silva (2019, p. 159) also 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," such as ensuring non-crossing of structural quantile functions. Also, as a minor practical difference with `sivqr`, `ivqreg2` does not currently support weights, `by`, or `svy bootstrap`. The `ivqte` command (Frölich and Melly 2010) estimates the "local quantile treatment effect," analogous to the local average treatment effect model of Imbens and Angrist (1994); see also Melly and Wüthrich (2017) for an excellent review. 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 discrete endogenous regressors, as well as simultaneity and reverse causality. For further comparison, see section 9.2.5 of Chernozhukov et al. (2017).

Existing Stata commands can be used to compute standard errors when sampling is not i.i.d. (as assumed for the standard errors reported by `sivqr`). The `bootstrap` command is compatible with `sivqr` and can accommodate clustered and stratified sampling, and `svy boostrap` can handle even more complex sampling designs; see [R] **bootstrap** and [SVY] **svy bootstrap**.

However, even `bootstrap` and `svy bootstrap` standard errors should be viewed cautiously if "weak instruments" are suspected. This is a potential pitfall of all the Stata commands discussed above, including `sivqr`. 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.

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.

Without quantiles, 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 how the τ-quantile of y would change if everybody increased from $x = x_0$ to $x = x_0 + 1$.

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

$$
\begin{aligned}
0 &= \mathrm{E}(v) = \mathrm{E}(y - \beta_0 - \beta_1 x) \\
0 &= \mathrm{E}(zv) = \mathrm{E}[z(y - \beta_0 - \beta_1 x)]
\end{aligned}
$$

The IVQR moment conditions are

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

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 $y - \beta_0 - \beta_1 x$ (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 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. The reported and stored results are similar to those of `ivregress`.

Syntax, options, and stored results are now shown. Prefix `by` is allowed (see [D] **by**), as well as `svy bootstrap` (see [SVY] **svy bootstrap**).

3.1 Syntax

`sivqr` *depvar* [*varlist*$_1$] (*varlist*$_2$ = *varlist*$_{\text{iv}}$) [*if*] [*in*] [*weight*] ,
`quantile(#)` [`bandwidth(#)` `level(#)` `reps(#)` `logiterations`
`noconstant` `seed(#)` `nodots`]

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

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

3.2 Options

`quantile(#)` specifies the quantile level, a real number between 0 and 1 (same as in [R] **qreg**); for example, `quantile(0.5)` specifies the median. 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(20)`.

`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. 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**).

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(q)`	quantile level requested
`e(bwidth_req)`	smoothing bandwidth requested (or plug-in value)		

Macros

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

Matrices

`e(b)`	estimated coefficient vector	`e(V)`	estimated variance–covariance matrix of the estimator, unless `reps(0)`

Functions

`e(sample)`	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

`r(level)`	confidence level

Matrix

`r(table)`	table of results

These are the standard values stored by `ereturn display` that are available after commands like `ivregress` and `qreg`; see [P] **ereturn**.

4 Examples

The examples in this section can all be replicated with the file `sivqr_examples.do`. Some code is omitted here to conserve space.

4.1 Simulated data

The following example uses simulated data. The data-generating process is simple, but the results illustrate multiple points. Specifically, the excluded instrument is $z_i \stackrel{\text{i.i.d.}}{\sim} \mathrm{N}(0,1)$, independent of the rank variable $u_i \stackrel{\text{i.i.d.}}{\sim} \mathrm{Unif}(0,1)$. The endogenous regressor is $x_i = \{z_i + \Phi^{-1}(u_i)\}/2$, and finally the dependent variable is $y_i = 2 + 3x_i + \Phi^{-1}(u_i)$, as seen in the following log, where $\Phi^{-1}(\cdot)$ is the standard normal inverse CDF (quantile function). To connect with the notation in section 2, this is equivalent to $y_i = \beta_0(u_i) + \beta_1(u_i)x_i$ with $\beta_0(u_i) = 2 + \Phi^{-1}(u_i)$ and $\beta_1(u_i) = 3$.

```
. clear
. set obs 1000
number of observations (_N) was 0, now 1,000
. set seed 112358
. gen z = rnormal()
. gen u = runiform()
. gen x = (z+invnormal(u))/2
. gen y = 2 + 3*x + invnormal(u)
```

This example has endogeneity but not heterogeneity. Because there is no heterogeneity (the slope is 3 for every individual), results can be compared with `ivregress`. Because there is substantial endogeneity, `qreg` cannot correctly estimate the slope (the causal effect of x_i on y_i). It instead estimates the slope of the conditional median function $\mathrm{Q}_\tau(y_i \mid x_i)$, which is 4 instead of 3. Without specifying z_i as an instrument for x_i, `sivqr` produces similar results to `qreg` and `bsqreg`; more detailed comparison is below.

First, `sivqr` runs quickly and accurately. Even with 200 bootstrap replications, the following results took 1.3 seconds on a (sub)standard personal computer (3GHz i5 processor, 8GB RAM) running Stata/SE. The slope estimate is close to the true value, which is inside the 95% confidence interval.

```
. sivqr y (x=z) , q(0.5) reps(200)
Computing plug-in bandwidth since bandwidth(-1)<0
Bootstrap replications (200)
----+--- 1 ---+--- 2 ---+--- 3 ---+--- 4 ---+--- 5
..................................................    50
..................................................   100
..................................................   150
..................................................   200
Smoothed instrumental variables quantile regression (SIVQR)
Smoothing bandwidth used = .2524819                Number of obs =       1000
```

y	Coef.	Std. Err.	z	P>|z|	[95% Conf. Interval]	
x	3.031681	.0816529	37.13	0.000	2.871644	3.191717
_cons	1.98397	.0385266	51.50	0.000	1.90846	2.059481

```
Instrumented:  x
Instruments:   z
```

Second, `bootstrap` can also be used for standard errors, but it is multiple times slower. Using `bootstrap` is indispensable when sampling is not i.i.d., but otherwise it just wastes time. The following took 5.6 seconds, over four times slower than above (for the same number of replications), and the confidence interval is nearly the same. The option `r(0)` ensures `sivqr` is not running its own bootstrap within each bootstrap replication (which would just take time without affecting the results).

```
. bootstrap , reps(200) : sivqr y (x=z) , q(0.5) r(0)
(running sivqr on estimation sample)
Bootstrap replications (200)
----+--- 1 ---+--- 2 ---+--- 3 ---+--- 4 ---+--- 5
..................................................    50
..................................................   100
..................................................   150
..................................................   200
Smoothed instrumental variables quantile regression (SIVQR)
Smoothing bandwidth used = .2524819             Number of obs =       1000

------------------------------------------------------------------------------
             |   Observed   Bootstrap                         Normal-based
           y |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
           x |   3.031681   .0808272    37.51   0.000     2.873262    3.190099
       _cons |    1.98397   .0376879    52.64   0.000     1.910103    2.057837
------------------------------------------------------------------------------
Instrumented:  x
Instruments:   z
```

Third, because there is no heterogeneity (as mentioned), `ivregress` produces similar results. The slope estimate is similar; the confidence interval is somewhat narrower (as expected with an additive Gaussian error).

```
. ivregress 2sls y (x=z) , robust
Instrumental variables (2SLS) regression          Number of obs   =      1,000
                                                  Wald chi2(1)    =    2653.65
                                                  Prob > chi2     =     0.0000
                                                  R-squared       =     0.8947
                                                  Root MSE        =     .96378

------------------------------------------------------------------------------
             |               Robust
           y |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
           x |   3.073666   .0596671    51.51   0.000     2.956721    3.190611
       _cons |   1.983035   .0304996    65.02   0.000     1.923257    2.042813
------------------------------------------------------------------------------
Instrumented:  x
Instruments:   z
```

Fourth, due to the endogeneity, `qreg` does not estimate the correct slope. It correctly estimates the slope of the conditional median function (which is 4), but that does not equal the slope on x_i in the structural model.

```
. qreg y x , vce(robust)
Iteration  1:  WLS sum of weighted deviations =  274.74999
Iteration  1: sum of abs. weighted deviations =  274.75018
Iteration  2: sum of abs. weighted deviations =  274.72823
Iteration  3: sum of abs. weighted deviations =  274.71609
Iteration  4: sum of abs. weighted deviations =  274.71316
Median regression                                   Number of obs =      1,000
  Raw sum of deviations 1172.131 (about 1.934894)
  Min sum of deviations 274.7132                    Pseudo R2     =     0.7656
```

```
             |               Robust
           y |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
           x |   4.014932   .0377995    106.22   0.000     3.940756    4.089107
       _cons |   1.985901   .0281678     70.50   0.000     1.930626    2.041176
```

Fifth, as a sanity check, if we were actually interested in the conditional median function, `sivqr` can produce estimates comparable to `bsqreg`. In the first `sivqr` call below, the option `b(0)` tells `sivqr` to use as little smoothing as possible, so it matches the usual QR estimate to at least 6 decimal places, but it takes 17 seconds to run only 10 bootstrap replications because computation is very difficult with so little smoothing. For comparison, `bsqreg` runs 200 replications in only 5.9 seconds. However, the second `sivqr` call with the plug-in bandwidth is in turn five times faster than `bsqreg`, running 200 replications in 1.1 seconds. This smoothing can also improve the statistical accuracy of quantile regression (Kaplan and Sun 2017; Fernandes et al. 2020).

```
. bsqreg y x , reps(200)
(fitting base model)

Bootstrap replications (200)
----+--- 1 ---+--- 2 ---+--- 3 ---+--- 4 ---+--- 5
..................................................    50
..................................................   100
..................................................   150
..................................................   200

Median regression, bootstrap(200) SEs            Number of obs =      1,000
  Raw sum of deviations 1172.131 (about 1.934894)
  Min sum of deviations 274.7132                 Pseudo R2     =     0.7656

------------------------------------------------------------------------------
           y |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
           x |   4.014932   .0392413    102.31   0.000     3.937927    4.091937
       _cons |   1.985901   .0250668     79.22   0.000     1.936711    2.035091
------------------------------------------------------------------------------

. sivqr y x , q(0.5) reps(10) b(0)
Bootstrap replications (10)
----+--- 1 ---+--- 2 ---+--- 3 ---+--- 4 ---+--- 5
..........
Smoothed instrumental variables quantile regression (SIVQR)
Smoothing bandwidth used = 2.51e-09              Number of obs =       1000

------------------------------------------------------------------------------
           y |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
           x |   4.014932   .0396392    101.29   0.000      3.93724    4.092623
       _cons |   1.985901   .0537378     36.96   0.000     1.880577    2.091225
------------------------------------------------------------------------------

(no endogenous regressors)

. sivqr y x , q(0.5) reps(200)
Computing plug-in bandwidth since bandwidth(-1)<0
Bootstrap replications (200)
----+--- 1 ---+--- 2 ---+--- 3 ---+--- 4 ---+--- 5
..................................................    50
..................................................   100
..................................................   150
..................................................   200
Smoothed instrumental variables quantile regression (SIVQR)
Smoothing bandwidth used = .1777643              Number of obs =       1000
```

y	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
x	4.014097	.0396228	101.31	0.000	3.936438	4.091757
_cons	1.999934	.0254458	78.60	0.000	1.950061	2.049806

```
(no endogenous regressors)
```

4.2 Example 1 from ivregress

The following extends Example 1 in [R] **ivregress**. Briefly, it examines the causal relationship between owner-occupied housing values and rent; see [R] **ivregress** for details. For comparison, `ivregress 2sls` is run first. Then, `sivqr` estimates the causal relationship at multiple quantile levels, to see any heterogeneity in unobservables. The 2SLS slope estimate is above the IVQR slope estimates at lower quantiles (0.25, 0.50) but below the IVQR slope estimate at the highest quantile level used (0.75). That is, given higher levels of the unmodeled determinants of rent (the u from section 2.1), the relationship between owner-occupied housing values and rent is more positive.

```
. webuse hsng2 , clear
(1980 Census housing data)
. ivregress 2sls rent pcturban (hsngval = faminc i.region) , vce(robust)
Instrumental variables (2SLS) regression         Number of obs   =         50
                                                 Wald chi2(2)    =      44.98
                                                 Prob > chi2     =     0.0000
                                                 R-squared       =     0.5989
                                                 Root MSE        =     22.166
```

rent	Coef.	Robust Std. Err.	z	P>|z|	[95% Conf.	Interval]
hsngval	.0022398	.000672	3.33	0.001	.0009227	.0035569
pcturban	.081516	.4445938	0.18	0.855	-.789872	.9529039
_cons	120.7065	15.25546	7.91	0.000	90.80636	150.6067

```
Instrumented:  hsngval
Instruments:   pcturban faminc 2.region 3.region 4.region
. sivqr rent pcturban (hsngval = faminc i.region), q(0.25) reps(200)
Computing plug-in bandwidth since bandwidth(-1)<0
Bootstrap replications (200)
----+--- 1 ---+--- 2 ---+--- 3 ---+--- 4 ---+--- 5
..................................................    50
..................................................   100
..................................................   150
..................................................   200
Smoothed instrumental variables quantile regression (SIVQR)
Smoothing bandwidth used = 72.17431                 Number of obs =         50
```

rent	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
hsngval	.0018309	.0004715	3.88	0.000	.0009068	.002755
pcturban	.3463769	.2982586	1.16	0.246	-.2381992	.930953
_cons	85.71129	12.24097	7.00	0.000	61.71942	109.7032

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

. sivqr rent pcturban (hsngval = faminc i.region), q(0.50) reps(200)
Computing plug-in bandwidth since bandwidth(-1)<0
Bootstrap replications (200)
----+--- 1 ---+--- 2 ---+--- 3 ---+--- 4 ---+--- 5
..................................................    50
..................................................   100
..................................................   150
..................................................   200
Smoothed instrumental variables quantile regression (SIVQR)
Smoothing bandwidth used = 7.093792              Number of obs =         50

------------------------------------------------------------------------------
        rent |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
     hsngval |   .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:  hsngval
Instruments:   pcturban faminc i.region

. sivqr rent pcturban (hsngval = faminc i.region), q(0.75) reps(200)
Computing plug-in bandwidth since bandwidth(-1)<0
Bootstrap replications (200)
----+--- 1 ---+--- 2 ---+--- 3 ---+--- 4 ---+--- 5
..................................................    50
..................................................   100
..................................................   150
..................................................   200
Smoothed instrumental variables quantile regression (SIVQR)
Smoothing bandwidth used = 87.15851              Number of obs =         50

------------------------------------------------------------------------------
        rent |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
     hsngval |   .0023771   .0007103     3.35   0.001      .000985    .0037693
    pcturban |   .0962631   .3830058     0.25   0.802    -.6544144    .8469406
       _cons |   157.6744   15.28434    10.32   0.000     127.7177    187.6312
------------------------------------------------------------------------------
Instrumented:  hsngval
Instruments:   pcturban faminc i.region
```

4.3 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. For comparison, `ivregress 2sls` is run first. I focus on the coefficient on job tenure.

To cluster standard errors at the individual level, `bootstrap` is used. When using `bootstrap`, it saves time to specify the `r(0)` option for `sivqr`. At the median level of unobservable determinants of wage, the `sivqr` estimates of the effect of job tenure and the corresponding standard errors are both similar to those from `ivregress 2sls`.

The IVQR estimates across different quantiles show considerable 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)`.

```
. 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 age 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)
------------------------------------------------------------------------------
             |               Robust
     ln_wage |      Coef.   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

. bootstrap , r(20) cluster(idcode) : sivqr ln_wage age c.age#c.age birth_yr g
> rade (tenure = union wks_work msp) , q(0.25) r(0)
(running sivqr on estimation sample)

Bootstrap replications (20)
----+--- 1 ---+--- 2 ---+--- 3 ---+--- 4 ---+--- 5
....................
Smoothed instrumental variables quantile regression (SIVQR)
Smoothing bandwidth used = .0464286                Number of obs =      28534
                              (Replications based on 4,711 clusters in idcode)
------------------------------------------------------------------------------
             |   Observed   Bootstrap                         Normal-based
     ln_wage |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      tenure |   .0876993   .0036316    24.15   0.000     .0805815    .0948171
         age |   .0248632   .0094449     2.63   0.008     .0063515    .0433749
             |
 c.age#c.age |  -.0007687   .0001772    -4.34   0.000    -.0011161   -.0004214
             |
    birth_yr |   -.012388   .0028221    -4.39   0.000    -.0179193   -.0068567
       grade |   .0630836   .0032909    19.17   0.000     .0566336    .0695336
       _cons |    .934314   .2028841     4.61   0.000     .5366685     1.33196
------------------------------------------------------------------------------
Instrumented:  tenure
Instruments:   age c.age#c.age birth_yr grade union wks_work msp

. bootstrap , r(20) cluster(idcode) : sivqr ln_wage age c.age#c.age birth_yr g
> rade (tenure = union wks_work msp) , q(0.50) r(0)
(running sivqr on estimation sample)

Bootstrap replications (20)
----+--- 1 ---+--- 2 ---+--- 3 ---+--- 4 ---+--- 5
....................
```

```
Smoothed instrumental variables quantile regression (SIVQR)
Smoothing bandwidth used = .0514216                Number of obs =      28534
                                (Replications based on 4,711 clusters in idcode)
```

ln_wage	Observed Coef.	Bootstrap Std. Err.	z	P>|z|	Normal-based [95% Conf.	Interval]
tenure	.1082771	.0038132	28.40	0.000	.1008034	.1157507
age	.0065329	.0081236	0.80	0.421	-.0093891	.0224548
c.age#c.age	-.0003668	.0001382	-2.65	0.008	-.0006377	-.0000958
birth_yr	-.0120568	.002564	-4.70	0.000	-.017082	-.0070315
grade	.0656386	.0022977	28.57	0.000	.0611352	.0701421
_cons	1.24776	.1607687	7.76	0.000	.9326592	1.562861

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

. bootstrap , r(20) cluster(idcode) : sivqr ln_wage age c.age#c.age birth_yr g
> rade (tenure = union wks_work msp) , q(0.75) r(0)
(running sivqr on estimation sample)

Bootstrap replications (20)
----+--- 1 ---+--- 2 ---+--- 3 ---+--- 4 ---+--- 5
....................
Smoothed instrumental variables quantile regression (SIVQR)
Smoothing bandwidth used = .0596183                Number of obs =      28534
                                (Replications based on 4,711 clusters in idcode)
```

ln_wage	Observed Coef.	Bootstrap Std. Err.	z	P>|z|	Normal-based [95% Conf.	Interval]
tenure	.1581675	.00888	17.81	0.000	.140763	.175572
age	.0102553	.007577	1.35	0.176	-.0045953	.025106
c.age#c.age	-.0002903	.0001312	-2.21	0.027	-.0005476	-.0000331
birth_yr	-.0075388	.0025393	-2.97	0.003	-.0125159	-.0025618
grade	.0719739	.0027838	25.85	0.000	.0665178	.0774301
_cons	.9373061	.1737396	5.39	0.000	.5967828	1.27783

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

4.4 Example with svy bootstrap

The following extends Example 1 in [SVY] **svy bootstrap**. The point is simply to show how `sivqr` can be used with `svy bootstrap` with complex sampling. Briefly, the goal below is to describe how birth weight variables with mother's age and whether it is a multiple birth, accounting for the sampling weights and complex sampling design; see [SVY] **svy bootstrap** for details on the data.

Although `qreg` does not support `svy bootstrap`, it does support weighted estimates, and the standard errors serve as a sanity check. Indeed, both the estimates and standard errors are similar to `sivqr`.

To save time from re-computing the plug-in bandwidth, `sivqr` is run once without `svy bootstrap` to save the returned plug-in bandwidth. The saved bandwidth is then explicitly specified when running `sivqr` with `svy`.

For comparison, `regress` is run with and without `svy`. As with `sivqr`, the `svy` standard errors are smaller than the heteroskedasticity-robust standard errors.

```
. webuse nmihs_bs , clear
. svyset [pweight=finwgt], bsrweight(bsrw1-bsrw20) vce(bootstrap)
      pweight: finwgt
          VCE: bootstrap
          MSE: off
    bsrweight: bsrw1 bsrw2 bsrw3 bsrw4 bsrw5 bsrw6 bsrw7 bsrw8 bsrw9 bsrw10
               bsrw11 bsrw12 bsrw13 bsrw14 bsrw15 bsrw16 bsrw17 bsrw18
               bsrw19 bsrw20
  Single unit: missing
     Strata 1: <one>
         SU 1: <observations>
        FPC 1: <zero>
. * qreg does not support svy bootstrap
. *   but can use as sanity check
. qreg birthwgt age multiple [pw=finwgt] , q(0.5)
Iteration  1:  WLS sum of weighted deviations =  8.339e+08
Iteration  1: sum of abs. weighted deviations =  8.339e+08
Iteration  2: sum of abs. weighted deviations =  8.335e+08
Iteration  3: sum of abs. weighted deviations =  8.335e+08
Iteration  4: sum of abs. weighted deviations =  8.335e+08
Iteration  5: sum of abs. weighted deviations =  8.334e+08
Iteration  6: sum of abs. weighted deviations =  8.334e+08
Iteration  7: sum of abs. weighted deviations =  8.334e+08
Median regression                                 Number of obs =      9,946
  Raw sum of deviations 8.57e+08 (about 3374)
  Min sum of deviations 8.33e+08                  Pseudo R2     =     0.0280
```

birthwgt	Coef.	Robust Std. Err.	t	P>|t|	[95% Conf. Interval]	
age	11.47368	1.51795	7.56	0.000	8.498196	14.44917
multiple	-969.0526	54.93039	-17.64	0.000	-1076.727	-861.3779
_cons	3092.211	40.05497	77.20	0.000	3013.695	3170.726

```
. * sivqr without svy SE
. sivqr birthwgt age multiple [pw=finwgt] , q(0.5) b(-1)
Computing plug-in bandwidth since bandwidth(-1)<0
Warning: plug-in bandwidth ignores weights (assumes iid)
Warning: coefficient estimates are valid, but standard errors assume iid data
         (usually not true with weights)
Bootstrap replications (20)
----+--- 1 ---+--- 2 ---+--- 3 ---+--- 4 ---+--- 5
....................
Smoothed instrumental variables quantile regression (SIVQR)
Smoothing bandwidth used = 138.7532               Number of obs =       9953
```

birthwgt	Coef.	Std. Err.	z	P>|z|	[95% Conf. Interval]	
age	11.46809	1.38428	8.28	0.000	8.754955	14.18123

```
    multiple | -955.8989   63.20481   -15.12   0.000    -1079.778   -832.0197
       _cons |  3091.695   38.21966    80.89   0.000     3016.786    3166.604
------------------------------------------------------------------------------
(no endogenous regressors)

. * use stored bandwidth to avoid re-computing every bootstrap replication
. local bw = `e(bwidth)´

. * svy SE are very similar in this case
. svy : sivqr birthwgt age multiple , q(0.5) b(`bw´)
(running sivqr on estimation sample)

Bootstrap replications (20)
----+--- 1 ---+--- 2 ---+--- 3 ---+--- 4 ---+--- 5
....................

Smoothed instrumental variables quantile regression (SIVQR)
                                                Number of obs     =      9,953
                                                Population size   =  3,898,922
                                                Replications      =         20

------------------------------------------------------------------------------
             |   Observed   Bootstrap                         Normal-based
    birthwgt |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         age |  11.46881    1.341486     8.55   0.000     8.839547    14.09807
    multiple | -956.7791    59.16788   -16.17   0.000    -1072.746   -840.8122
       _cons |  3091.688    33.92945    91.12   0.000     3025.188    3158.189
------------------------------------------------------------------------------

. * OLS for comparison
. svy , nodots : reg  birthwgt age multiple

Survey: Linear regression                       Number of obs     =      9,946
                                                Population size   =  3,895,562
                                                Replications      =         20
                                                Wald chi2(2)      =     726.17
                                                Prob > chi2       =     0.0000
                                                R-squared         =     0.0627

------------------------------------------------------------------------------
             |   Observed   Bootstrap                         Normal-based
    birthwgt |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         age |  11.30174    1.113209    10.15   0.000     9.119895    13.48359
    multiple | -977.7638    44.08936   -22.18   0.000    -1064.177   -891.3503
       _cons |  3077.936    27.50242   111.92   0.000     3024.032     3131.84
------------------------------------------------------------------------------

. reg birthwgt age multiple [pw=finwgt] , robust
(sum of wgt is   3.8956e+06)

Linear regression                               Number of obs     =      9,946
                                                F(2, 9943)        =     262.84
                                                Prob > F          =     0.0000
                                                R-squared         =     0.0627
                                                Root MSE          =     578.41

------------------------------------------------------------------------------
             |               Robust
    birthwgt |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         age |  11.30174    1.280292     8.83   0.000     8.792112    13.81138
    multiple | -977.7638    46.08628   -21.22   0.000    -1068.102   -887.4254
       _cons |  3077.936    33.51681    91.83   0.000     3012.236    3143.636
------------------------------------------------------------------------------
```

4.5 Example from Chernozhukov and Hansen (2008) and Graddy (1995)

The following example is similar to that of Chernozhukov and Hansen (2008, §5.1). The data come from Graddy (1995), who has data from the Fulton fish market including log prices, log quantity, and weather conditions. The goal is to estimate the demand curve; specifically, because price and quantity are in logs, the slope coefficient is the elasticity of demand. Because price and quantity are determined simultaneously by supply and demand, (quantile) regression cannot estimate the demand curve correctly. Instead, weather conditions are used to instrument for price; the identifying assumption is that weather shifts the supply curve but not the demand curve, providing variation in price that is "exogenous" in the sense of being unrelated to demand shocks. Further, the IVQR model allows a different demand elasticity given different levels of demand shocks. See Section 5.1 of Chernozhukov and Hansen (2008) for details.

After loading the data, some "first-stage" regressions are run to examine the strength of possible instruments. Although Chernozhukov and Hansen (2008) use the binary `stormy` and `mixed` weather variables, the results show `windspd` alone achieves a higher F statistic, 21.4 instead of 15.8. Although this has not been formally shown to measure instrument strength for IVQR (only 2SLS), it should provide a reasonable approximation. Consequently, `windspd` is used as the excluded instrument.

```
. use http://people.brandeis.edu/~kgraddy/datasets/fishdata.dta , clear
. ren price lnp
. ren qty lnq
. reg lnp windspd
      Source |       SS           df       MS      Number of obs   =       111
-------------+----------------------------------   F(1, 109)       =     21.40
       Model |  2.63356341         1  2.63356341   Prob > F        =    0.0000
    Residual |  13.4125776       109   .12305117   R-squared       =    0.1641
-------------+----------------------------------   Adj R-squared   =    0.1565
       Total |   16.046141       110  .145874009   Root MSE        =    .35079

------------------------------------------------------------------------------
         lnp |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
     windspd |   .7416439   .1603121     4.63   0.000     .4239104    1.059377
       _cons |  -2.316782    .460131    -5.04   0.000    -3.228746   -1.404817
------------------------------------------------------------------------------

. reg lnp stormy mixed
      Source |       SS           df       MS      Number of obs   =       111
-------------+----------------------------------   F(2, 108)       =     15.83
       Model |  3.63827422         2  1.81913711   Prob > F        =    0.0000
    Residual |  12.4078668       108  .114887655   R-squared       =    0.2267
-------------+----------------------------------   Adj R-squared   =    0.2124
       Total |   16.046141       110  .145874009   Root MSE        =    .33895

------------------------------------------------------------------------------
         lnp |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      stormy |   .4368179   .0783792     5.57   0.000     .2814567     .592179
       mixed |   .2359673   .0770202     3.06   0.003        .0833    .3886347
       _cons |  -.3918889   .0505278    -7.76   0.000    -.4920438    -.291734
```

First, sivqr is run at three different quantile levels. Although the confidence intervals are wide (as in Chernozhukov and Hansen 2008), the point estimates suggest the demand elasticity depends considerably on the level of unobserved demand shock. The estimate with q(0.25) is −1.5 (a 1% increase in price causes a 1.5% decrease in quantity demanded, all else held equal), whereas the estimates at higher quantile levels are closer to −1. The 2SLS estimate (shown below those) is in between.

```
. sivqr lnq (lnp=windspd) , q(0.25) reps(100)
Computing plug-in bandwidth since bandwidth(-1)<0
Bootstrap replications (100)
----+--- 1 ---+--- 2 ---+--- 3 ---+--- 4 ---+--- 5
..................................................    50
..................................................   100
Smoothed instrumental variables quantile regression (SIVQR)
Smoothing bandwidth used = .3345163              Number of obs =        111
------------------------------------------------------------------------------
         lnq |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         lnp |  -1.508546    .454212    -3.32   0.001    -2.398785   -.6183063
       _cons |   7.658637   .1973858    38.80   0.000     7.271768    8.045506
------------------------------------------------------------------------------
Instrumented:  lnp
Instruments:   windspd
. sivqr lnq (lnp=windspd) , q(0.50) reps(100)
Computing plug-in bandwidth since bandwidth(-1)<0
Bootstrap replications (100)
----+--- 1 ---+--- 2 ---+--- 3 ---+--- 4 ---+--- 5
..................................................    50
..................................................   100
Smoothed instrumental variables quantile regression (SIVQR)
Smoothing bandwidth used = .2999388              Number of obs =        111
------------------------------------------------------------------------------
         lnq |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         lnp |  -.9232779   .6899757    -1.34   0.181    -2.275605    .4290497
       _cons |   8.482092   .1658201    51.15   0.000      8.15709    8.807093
------------------------------------------------------------------------------
Instrumented:  lnp
Instruments:   windspd
. sivqr lnq (lnp=windspd) , q(0.75) reps(100)
Computing plug-in bandwidth since bandwidth(-1)<0
Bootstrap replications (100)
----+--- 1 ---+--- 2 ---+--- 3 ---+--- 4 ---+--- 5
..................................................    50
..................................................   100
Smoothed instrumental variables quantile regression (SIVQR)
Smoothing bandwidth used = .3077761              Number of obs =        111
------------------------------------------------------------------------------
         lnq |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         lnp |  -1.102318   .3106159    -3.55   0.000    -1.711114   -.4935221
       _cons |   8.861603   .0610157   145.23   0.000     8.742015    8.981192
------------------------------------------------------------------------------
Instrumented:  lnp
Instruments:   windspd
```

```
. ivregress 2sls lnq (lnp=windspd) , vce(robust)
Instrumental variables (2SLS) regression         Number of obs   =        111
                                                 Wald chi2(1)    =       6.74
                                                 Prob > chi2     =     0.0094
                                                 R-squared       =          .
                                                 Root MSE        =     .76074

------------------------------------------------------------------------------
             |               Robust
         lnq |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         lnp |  -1.265413   .4875543    -2.60   0.009    -2.221002   -.3098244
       _cons |   8.278343   .1277306    64.81   0.000     8.027996     8.52869
------------------------------------------------------------------------------
Instrumented:  lnp
Instruments:   windspd
```

Second, day-of-week dummy (indicator) variables are added as control variables. Monday is `day1`, Tuesday `day2`, Wednesday `day3`, and Thursday `day4`, with Friday the omitted category. This lowers all the demand elasticity estimates, but the same qualitative pattern across quantiles remains. The standard errors are also lower in some cases. Tuesday and Wednesday have lower demand than other days.

```
. sivqr lnq (lnp=windspd) day1-day4 , q(0.25) reps(100)
Computing plug-in bandwidth since bandwidth(-1)<0
Bootstrap replications (100)
----+--- 1 ---+--- 2 ---+--- 3 ---+--- 4 ---+--- 5
..................................................    50
..................................................   100
Smoothed instrumental variables quantile regression (SIVQR)
Smoothing bandwidth used = .2863885              Number of obs =        111
------------------------------------------------------------------------------
         lnq |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         lnp |  -1.108158   .5590716    -1.98   0.047    -2.203918   -.0123977
        day1 |  -.1330531   .3078762    -0.43   0.666    -.7364793    .4703731
        day2 |  -.6891695   .3610322    -1.91   0.056     -1.39678    .0184406
        day3 |    -.55762   .2965491    -1.88   0.060    -1.138845    .0236055
        day4 |   .2560015   .3659021     0.70   0.484    -.4611535    .9731565
       _cons |       8.05   .4248509    18.95   0.000     7.217307    8.882692
------------------------------------------------------------------------------
Instrumented:  lnp
Instruments:   day1 day2 day3 day4 windspd
. sivqr lnq (lnp=windspd) day1-day4 , q(0.50) reps(100)
Computing plug-in bandwidth since bandwidth(-1)<0
Bootstrap replications (100)
----+--- 1 ---+--- 2 ---+--- 3 ---+--- 4 ---+--- 5
..................................................    50
..................................................   100
Smoothed instrumental variables quantile regression (SIVQR)
Smoothing bandwidth used = .2451134              Number of obs =        111
------------------------------------------------------------------------------
         lnq |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         lnp |  -.7263921   .5290964    -1.37   0.170    -1.763402    .3106177
        day1 |  -.0296199   .2650984    -0.11   0.911    -.5492033    .4899634
        day2 |   -.512335   .2926166    -1.75   0.080    -1.085853     .061183
        day3 |  -.5757288   .2819132    -2.04   0.041    -1.128269    -.023189
```

```
        day4   .0676082   .2051011     0.33   0.742    -.3343825    .4695989
       _cons   8.681582   .1521717    57.05   0.000     8.383331    8.979833
------------------------------------------------------------------------------
Instrumented:  lnp
Instruments:   day1 day2 day3 day4 windspd
. sivqr lnq (lnp=windspd) day1-day4 , q(0.75) reps(100)
Computing plug-in bandwidth since bandwidth(-1)<0
Bootstrap replications (100)
----+--- 1 ---+--- 2 ---+--- 3 ---+--- 4 ---+--- 5
..................................................    50
..................................................   100
Smoothed instrumental variables quantile regression (SIVQR)
Smoothing bandwidth used = .2805696                Number of obs =        111
------------------------------------------------------------------------------
         lnq       Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
------------------------------------------------------------------------------
         lnp   -.9305434   .3408895    -2.73   0.006    -1.598675   -.2624121
        day1    .1484225   .2219872     0.67   0.504    -.2866645    .5835095
        day2    -.346092   .1905604    -1.82   0.069    -.7195834    .0273994
        day3   -.4181707   .1680462    -2.49   0.013    -.7475352   -.0888061
        day4   -.0011363   .1383564    -0.01   0.993      -.27231    .2700373
       _cons     8.99673   .0942567    95.45   0.000     8.811991     9.18147
------------------------------------------------------------------------------
Instrumented:  lnp
Instruments:   day1 day2 day3 day4 windspd
```

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, $\boldsymbol{x}$ a column vector of regressors, and u a scalar unobserved variable. Let $\boldsymbol{d}$ be a subset of $\boldsymbol{x}$ containing all the endogenous regressors. Let $\boldsymbol{z}$ be a column vector of all exogenous variables: both exogenous regressors in $\boldsymbol{x}$ as well as the excluded instruments (instrumental variables). The scalar random variable u is unobserved. Following *Stata Journal* convention, vectors are lowercase bold (like $\boldsymbol{x}$) 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 $\boldsymbol{\beta}$ in the structural model

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

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

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

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

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

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 $\mathrm{E}(y \mid \boldsymbol{z})$ in (2) with a conditional τ-quantile restriction for some $0 < \tau < 1$:

$$0 = \mathrm{Q}_\tau(u \mid \boldsymbol{z}) \tag{3}$$

That is, conditional on any value of $\boldsymbol{z}$, 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 $\boldsymbol{z}$. Thus, (3) is equivalent to

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

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 $\mathrm{E}[1\{\cdot\}]$ and applying the law of iterated expectations as in (2),

$$\begin{aligned} \tau = \mathrm{E}[1\{y - \boldsymbol{x}'\boldsymbol{\beta} \le 0\} \mid \boldsymbol{z}] &\implies 0 = \mathrm{E}[1\{y - \boldsymbol{x}'\boldsymbol{\beta} \le 0\} - \tau \mid \boldsymbol{z}] \\ &\implies \mathbf{0} = \mathrm{E}[\boldsymbol{z}(1\{y - \boldsymbol{x}'\boldsymbol{\beta} \le 0\} - \tau)] \end{aligned} \tag{4}$$

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 $1\{\cdot\}$, 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 $1\{v \le 0\}$, $\tilde{I}(v)$ is a function of v that smoothly decreases from 1 to 0 over $-1 \le v \le 1$ instead of decreasing discontinuously from 1 to 0 at $v = 0$.[2] Adding a bandwidth h allows $\tilde{I}(v/h)$ to decrease more steeply over $-h \le v \le h$. The estimator $\widehat{\boldsymbol{\beta}}$ solves the "smoothed estimating equations" (or "smoothed moment conditions")

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

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:

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

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 $v \le -1$ and $\tilde{I}(v) = 0$ for $v \ge 1$, 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 ($\boldsymbol{x} = \boldsymbol{z} = 1$). 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 $y_i - \boldsymbol{x}_i'\widehat{\boldsymbol{\beta}}$, just as in the mean IV moment conditions; if $\tau \ne 0.5$, 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 $\boldsymbol{z}$ is used in (5). Specifically, it is replaced by the linear projection of the regressor vector $\boldsymbol{x}$ onto the instruments $\boldsymbol{z}$, which has the same dimension as $\boldsymbol{x}$ and thus $\boldsymbol{\beta}$. This linear projection is motivated by the two-stage least squares estimator, which can also be written as the mean IV estimator when replacing $\boldsymbol{z}$ with the linear projection of $\boldsymbol{x}$ onto $\boldsymbol{z}$. 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 their own coefficient vector $\boldsymbol{b}$. This is a "random coefficient" model, meaning $\boldsymbol{b}$ can

2. This notation is simpler and equivalent to (but different than) that in Kaplan and Sun (2017).
3. See also Chapter 6 of Kaplan (2020) for an introductory discussion.

differ by individual the same way that y, $\boldsymbol{x}$, and $\boldsymbol{z}$ do:

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

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 $\boldsymbol{b}$ can be written as a deterministic (but unknown) vector-valued function $\boldsymbol{\beta}(\cdot)$ applied to a scalar unobserved u:

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

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 $\boldsymbol{\beta}(\cdot)$. Assume a "monotonicity" property that given any value of $\boldsymbol{x}$, $\boldsymbol{x}'\boldsymbol{\beta}(u)$ is an increasing function of u, and like before assume the instruments are independent of u. Then,

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

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 $q_\tau(\boldsymbol{x}) = \boldsymbol{x}'\boldsymbol{\beta}(\tau)$ 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 $v \equiv y - \boldsymbol{x}'\boldsymbol{\beta}(\tau)$ 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 propositon 2 of Kaplan and Sun (2017) with $v \perp\!\!\!\perp \boldsymbol{z}$,

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

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 $\boldsymbol{\beta}(\tau)$.

The `sivqr` plug-in bandwidth is the minimum of a few different values. Taking the minimum errs on the side 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 $\widehat{v}_i$ are then recomputed from the new $\widehat{\boldsymbol{\beta}}(\tau)$, 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 $\widehat{\boldsymbol{\beta}}(\tau)$ must be used to construct residuals

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

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, §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:

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

Following the Gaussian reference approach of Silverman (1986, §3.4.2), assume $f(\cdot)$ is the density of a $\mathrm{N}(\mu,\sigma^2)$ 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

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

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

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

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,[5]

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

4. `D[PDF[NormalDistribution[\[Mu], \[Sigma]], x], {x,2}]` at http://www.wolframalpha.com for example; then plug in $x=0$ and $-\mu/\sigma=\Phi^{-1}(\tau)$.
5. 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^*,

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

Following Silverman (1986, eq. (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

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

where now b is the bandwidth (to avoid confusion with h and s). The asymptotic mean squared error for this estimator at the point zero is from Wand and Jones (1994),[6]

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

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 zero), this 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 $K(\cdot)$,

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

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

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

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

yields

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

Again, to get a feasible bandwidth $\widehat{b}$, σ is replaced by 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 plug-in bandwidth $\widehat{h}$.

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 is[8]

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

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 $\widehat{v}_i$ or the sample interquartile range of the $\widehat{v}_i$ 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` and `svy bootstrap` can be used; see

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

[R] **bootstrap** and [SVY] **svy bootstrap**.) This is a 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). In each replication, weights are set as $w_i = \xi_i/\overline{\xi}$ where the ξ_i are i.i.d. standard exponential random variables and $\overline{\xi}$ is their average; equivalently, the vector $(w_1, \ldots, w_n)/n$ 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 $\widehat{\boldsymbol{\beta}}$.

6 Conclusion

The new `sivqr` command offers smoothed estimation of the instrumental variables quantile regression model of Chernozhukov and Hansen (2005). The `sivqr` command improves greatly upon `ivqreg`, while complementing commands for alternative models (`ivqreg2`, `cqiv`, and `ivqte`). The new `sivqr` uses smoothing to improve computation and accuracy, as well as offering fast Bayesian bootstrap standard errors. Further, `sivqr` is compatible with `by` and weights, as well as both `bootstrap` and `svy bootstrap`.

Potential applications abound. For example, although the assumptions underlying `sivqr` may be valid even if those underlying `ivregress` are violated, and vice versa, often `sivqr` can be applied to the same data with the same instruments as `ivregress`, to learn more about heterogeneity of effects.

7 Acknowledgements

There would be no `sivqr` Stata command without the work of Yixiao Sun on Kaplan and Sun (2017). Thanks to Graddy (1995) for making her data public.

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.

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. 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.

Graddy, K. 1995. Testing for imperfect competition at the Fulton fish market. *RAND Journal of Economics* 26(1): 75–92. https://www.jstor.org/stable/2556036.

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 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.

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.

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).