Flexible and fast estimation of quantile treatment effects: The `rqr` and `rqrplot` commands

Nicolai T. Borgen
University of Oslo
Oslo, Norway
n.t.borgen@isp.uio.no

Andreas Haupt
Karlsruhe Institute of Technology
Karlsruhe, Germany
andreas.haupt@kit.edu

Øyvind Wiborg
University of Oslo
Oslo, Norway
o.n.wiborg@sosge.uio.no

Abstract. Using quantile regression models to estimate quantile treatment effects is becoming increasingly popular. This paper introduces the `rqr` command that can be used to estimate residualized quantile regression (RQR) coefficients and the `rqrplot` postestimation command that can be used to effortlessly plot the coefficients. The main advantages of the `rqr` command compared to other Stata commands that estimate (unconditional) quantile treatment effects are that it can include high-dimensional fixed effects and that it is considerably faster than the other commands.

Keywords: residualized quantile regression model, `rqr`, `rqrplot`, quantile regression, fixed effects

1 Introduction

Quantile regression models have become increasingly popular in the last couple of decades and considerable methodological developments have occurred within the same time frame. One such development is the residualized quantile regression (RQR) model, which can be used to identify unconditional quantile treatment effects (QTEs) (Borgen et al. 2021). This paper introduces the `rqr` command that estimates RQR coefficients and the `rqrplot` postestimation command that effortlessly plots RQR coefficients.

Quantile regression models share the fact that they are interested in quantiles of the outcome variable rather than simply the mean. However, various quantile regression models have different aims and interpretations. Therefore, let us begin by clarifying how the RQR model, and the corresponding `rqr` command, relate to other quantile regression approaches and Stata commands.

There are three overall classes of quantile regression models. First, we have the conditional quantile regression (CQR) model (Koenker 2005), which estimates group-specific quantile differences. The CQR model can be estimated in Stata using the official `qreg` command, and variants of the model can be estimated using user-written commands such as `xtqreg` (Machado and Silva 2018b, 2019), `ivqreg2` (Machado and Silva 2019, 2018a), `sivqr` (Kaplan 2020), and `qmodel` (Bottai and Orsini 2019a). The seminal work of Koenker and colleagues in the 1970s marked the beginning of quantile regressions, and quantile regressions have historically been synonymous with CQR. However, since

treatment effects are based on the conditional distribution of the outcome variable, the CQR model is ill-suited to answer many research questions (Borgen et al. 2020; Wenz 2019; Firpo 2007; Powell 2020; Frölich and Melly 2010), especially in models including fixed effects Killewald and Bearak (2014).

The second class of quantile regression models concerns estimating distributional effects, where the unconditional quantile regression (UQR) model developed by Firpo et al. (2009) is one example. The UQR model can be estimated in Stata using the user-written commands `rifreg`¹, `xtrifreg` (Borgen 2016), and `rifhdreg` (Rios-Avila 2020). The UQR model was cleverly developed to draw conclusions on how independent variables influenced unconditional quantile values, which are different from the individual-level treatment effects studied in the CQR model and the QTE models described below. Unfortunately, UQR coefficients have been widely used to study unconditional QTEs within a range of disciplines, as discussed by Borgen et al. (2020).

Finally, the last class of quantile regression models is QTE models, in which the RQR model belongs. QTE models identify unconditional QTEs, which resemble average treatment effects more than the conditional quantile differences estimated by the CQR model. One classic example of QTEs is whether high-wage women bear a larger motherhood wage penalty than low-wage women (Killewald and Bearak 2014; Budig and Hodges 2010). In the aftermath of the COVID-19 pandemic, another example where QTEs will be targeted is studies that estimate the influence of school closures on low- and high-achieving students.

Firpo (2007) provided an early proposal for estimating unconditional QTEs in his seminal propensity score approach, which can be estimated in Stata using the user-written command `ivqte` (Frölich and Melly 2010). Frölich and Melly (2013) extended Firpo’s approach allowing for instrumental variables. However, Firpo (2007)’s propensity score approach only handles binary treatment variables, and including fixed effects is problematic. Powell (2020) has developed the generalized quantile regression (GQR) model that allows for non-binary treatment variables and instrumental variables. The GQR model can be estimated in Stata using the `genqreg` command. The main limitation of this approach is that including high-dimensional fixed effects is challenging or practically impossible and that the model is computationally demanding.

The RQR model is a flexible approach that complements these existing QTE models. Compared to much of the literature on quantile regressions, the RQR model is transparent and conceptually easy to understand. RQR coefficients are estimated through a two-step approach. In the first step, the treatment is regressed on control variables using ordinary least squares (OLS), and the residuals of the treatment variable is obtained. This step decomposes the variance of the treatment variable into a piece explained by the observed control variables and a residual piece that is orthogonal to the observed controls. Then, in the second step, the outcome is regressed on the residualized treatment variable using CQR algorithms. Theoretical arguments and data simulations that prove that this simple approach indeed works can be found in Borgen et al. (2021).

1. The command can be downloaded at <https://sites.google.com/view/nicole-m-fortin/data-and-programs>

There are several advantages of using the `rqr` command rather than the existing `ivqte` and `genqreg` commands to estimate unconditional QTEs. First, unlike `ivqte` and `genqreg`, the `rqr` command can easily handle high-dimensional fixed effects. Second, the `rqr` command is computationally efficient, irrespective of whether fixed effects are included or not. This is important since the computational burden of quantile regressions may discourage researchers from using these approaches (Chernozhukov et al. 2020a; Fortin et al. 2011). Third, the RQR framework can draw on the already extensive literature on CQR, including models to estimate parametric quantile regressions (Bottai and Orsini 2019b), methods to deal with count data (Machado and Silva 2005), and fast algorithms. Concerning algorithms, computational time can be reduced considerably by applying newly developed quantile regression algorithms for Stata (Chernozhukov et al. 2020a,b).

The contribution of this paper is to introduce the `rqr` command that estimates the RQR model and the `rqrplot` postestimation command that plots RQR coefficients. To be sure, it is straightforward to implement the RQR estimator in all software that provides a package for CQR or linear programming, making it accessible to most researchers even without a Stata command. Yet, a user-friendly Stata command streamlines the process, making the research process more efficient and less prone to coding errors, especially when bootstrapping the standard errors.

In the following, we define the unconditional QTE, followed by a brief explanation of the RQR model. Then we introduce the `rqr` command and the `rqrplot` plots along with some examples of use before providing a brief conclusion. To avoid cluttering the text with too many examples, we focus the main text on the primary usage of the `rqr` and `rqrplot` commands. Additional examples of the commands are included in Appendix A and referenced briefly in the main text. Finally, Appendix B compares RQR to other QTE approaches.

2 Quantile treatment effects

The goal of the RQR model is to identify quantile differences, which under the assumption of ignorability can be interpreted as QTEs. Consider a binary treatment variable T_i and let Y_{i1} and Y_{i2} be the values of the outcome for individual i under the treatment and no-treatment conditions. Further, let Q_{Y1}^τ and Q_{Y0}^τ be the values of the quantile τ under the potential outcomes given the treatment and no-treatment conditions. Then, QTEs are defined as:

$$QTE^\tau = Q_{Y1}^\tau - Q_{Y0}^\tau \quad (1)$$

Such quantile differences can be computed from observed data by comparing quantiles of the outcome distribution for individuals with different treatment values. To illustrate, let us look at union wage differentials in a subsample of the National Longitudinal Survey (nlswork.dta) and, for now, naively pretend that union status is randomly assigned (an assumption that will be relaxed soon).

```
use http://www.stata-press.com/data/r14/nlswork, clear
gen wage=exp(ln_wage)
keep if !missing(wage,union,year,age,grade,south,ind_code, occ_code)
```

It is well-known that average wage differentials are computed by comparing average wages among unionized workers with non-unionized workers' average wages. Likewise, median wage differentials are computed by comparing median wages among unionized workers with corresponding median wages among non-unionized workers. That is, we rank unionized workers' wages from the lowest to the highest and find the worker's salary that separates the top 50% from the bottom half. This salary represents the median wage among unionized workers. After repeating this exercise for non-unionized workers, we can compare median wages among the two groups.

In the example below, the median wages among unionized workers (`union=1`) are \$1.460 larger than the median wages among nonunionized workers (`union=0`), while the average wage differentials are roughly the same, at 1.462. However, the quantile differences are smaller at the 25th quantile (1.040) and larger at the 75th quantile (1.849).

```
. tabstat wage, by(union) col(stat) stat(mean p25 p50 p75) format(%9.6f) nototal
Summary for variables: wage
Group variable: union (1 if union)
```

union	Mean	p25	p50	p75
0	6.144633	4.000000	5.346215	7.282847
1	7.606861	5.040323	6.806523	9.131400

Estimating quantile differences is effortless in cases where no control variables are included. With binary independent variables we can simply compare quantile values of treated and untreated directly, as shown above, or we can use the `qreg` command. The output below shows that the median regression coefficients corresponds to the differences calculated based on the `tabstat` command above. (Throughout the paper we suppress the iteration log by including option `nolog` to avoid clutter the output.)

```
. qreg wage union, q(.50) nolog
Median regression
Raw sum of deviations 22415.01 (about 5.6399989)
Min sum of deviations 21891.94
Number of obs = 19,083
Pseudo R2 = 0.0233
```

wage	Coefficient	Std. err.	t	P> t	[95% conf. interval]	
union	1.460308	.0566089	25.80	0.000	1.349349	1.571266
_cons	5.346215	.027407	195.07	0.000	5.292495	5.399935

It is clearly naïve to assume that there is no sorting into unionization based on workplace characteristics nor individual characteristics (Mastekaasa 2013; Card 1996). With selection effects, quantile differences are biased and cannot be interpreted as (quantile) treatment effects, similarly to how confounders bias coefficients in linear regression models. Generally, treatment effects can rarely be estimated without control variables

or using quasi-experimental designs.

Unfortunately, estimating QTE in the presence of control variables is more complicated than in linear regression models. It is widely discussed how including control variables changes the coefficients' interpretation in the conditional quantile regression model (Firpo 2007; Frölich and Melly 2010; Wenz 2019). Thus, we cannot use the `qreg` command directly to estimate QTEs in the presence of covariates. Building on the work of Firpo (2007) and Powell (2020), the RQR model was developed to estimate QTEs in the presence of control variables (Borgen et al. 2021).

3 The residualized quantile regression (RQR) model

The RQR model identifies unconditional QTEs in the presence of control variables using a two-step approach. First, the treatment variable is decomposed into a piece explained by the observed control variables and a residual piece that is orthogonal to the controls, using OLS. Then, the outcome variable is regressed on the residualized treatment variable using the CQR algorithm.

Before introducing the `rqr` and `rqrplot` commands, which make it easy to estimate and plot results from the RQR model, let us show a simple example of how the RQR model works. We continue with the union wage example. To be sure, the union wage effects conditional on the observed control variables in this data cannot be interpreted as causal effects. Nevertheless, we chose this example because the data is accessible for all readers and that it allows for illustrating the use of both observed control variables and high-dimensional fixed effects. In the following we will estimate union wage effects controlling for interview year, age and age squared, highest grade completed, living in the south, and 11 dummies for employer's industry.

To estimate RQR coefficients, we start by regressing the treatment (union status) on the observed control variables using `regress`.

```
. regress union i.year c.age#c.age grade south i.ind_code
```

Source	SS	df	MS	Number of obs	=	19,083
				F(26, 19056)	=	81.92
Model	344.289968	26	13.2419219	Prob > F	=	0.0000
Residual	3080.25177	19,056	.161642095	R-squared	=	0.1005
				Adj R-squared	=	0.0993
Total	3424.54174	19,082	.179464508	Root MSE	=	.40205

union	Coefficient	Std. err.	t	P> t	[95% conf. interval]	
year						
71	.018405	.0195073	0.94	0.345	-.019831	.056641
72	-.0101265	.0183796	-0.55	0.582	-.0461521	.0258992
73	.0011525	.0191724	0.06	0.952	-.0364272	.0387321
77	-.0390011	.0184827	-2.11	0.035	-.0752288	-.0027734
78	.0121601	.0202757	0.60	0.549	-.0275821	.0519023
80	.0550853	.0205532	2.68	0.007	.0147992	.0953715
82	-.0000631	.0209669	-0.00	0.998	-.04116	.0410338
83	-.0286947	.0216576	-1.32	0.185	-.0711455	.0137561

85	-.0003936	.022127	-0.02	0.986	-.0437644	.0429773
87	-.0358374	.0228739	-1.57	0.117	-.0806722	.0089974
88	-.0205444	.0237587	-0.86	0.387	-.0671136	.0260247
age	.0024213	.0057241	0.42	0.672	-.0087984	.013641
c.age#c.age	6.86e-06	.0000881	0.08	0.938	-.0001659	.0001796
grade	.0029041	.0013519	2.15	0.032	.0002542	.005554
south	-.109361	.005994	-18.24	0.000	-.1211098	-.0976122
ind_code						
2	-.0048595	.0741743	-0.07	0.948	-.1502476	.1405286
3	.0431706	.0460504	0.94	0.349	-.0470923	.1334336
4	.2563112	.0358922	7.14	0.000	.1859593	.3266631
5	.491358	.0374438	13.12	0.000	.4179648	.5647511
6	.0581308	.0360734	1.61	0.107	-.0125762	.1288378
7	.0146515	.0366234	0.40	0.689	-.0571337	.0864367
8	.0343703	.0391389	0.88	0.380	-.0423454	.111086
9	.061825	.0376918	1.64	0.101	-.0120543	.1357043
10	.0541964	.0496313	1.09	0.275	-.0430854	.1514782
11	.2062141	.035792	5.76	0.000	.1360587	.2763696
12	.2574015	.0370433	6.95	0.000	.1847935	.3300096
_cons	-.0117578	.093409	-0.13	0.900	-.1948477	.1713321

Then, based on the coefficients from this OLS model, we obtain the residuals, which we here call **r**.

```
. predict double r, residuals
```

Finally, we include the variable **r** (i.e., the residuals) as the only independent variable in the **qreg** model.

```
. qreg wage r, quantile(.50) nolog
```

Median regression		Number of obs =	19,083
Raw sum of deviations	22415.01 (about 5.6399989)		
Min sum of deviations	22263.56	Pseudo R2 =	0.0068

wage	Coefficient	Std. err.	t	P> t	[95% conf. interval]
r	.8049713	.057587	13.98	0.000	.6920957 .917847
_cons	5.66309	.0231363	244.77	0.000	5.617741 5.70844

The treatment variable's coefficient from this quantile regression model is the RQR coefficient. We see that unionized workers' median wages are 0.805 higher than what it would have been in the hypothetical what-if scenario where they were not unionized.

4 The `rqr` command

4.1 Syntax

```
rqr depvar treatment [if] [in] [weight] [, quantile(numlist)
    controls(varlist) absorb(varlist) step1command(string)
    step2command(string) options_step1(string) options_qreg(string)
    options_qrprocess(string) options_predict(string) generate_r(varname)
    smoothing(a,b) print1step]
```

4.2 Options

`quantile(numlist)` specifies the quantile and can be either one quantile or a range of quantiles. The specified quantile(s) should be number(s) between 0 and 1. The default is `quantile(.5)`.

`controls(varlist)` lists the control variables to be included in the first-step regression. High-dimensional fixed effects should be included in the `absorb()` option.

`absorb(varlist)` lists the fixed effects to be included in the first-step regression. The default estimator is `areg` when one fixed effects is listed and the user-written `reghdfe` when more than one fixed effects are included.

`step1command(string)` decides the first-step estimator. The default is `regress` when no fixed effects are included, `areg` when one fixed effects is included, and the user-written `reghdfe` when more than one fixed effects are included.

`step2command(string)` decides the second-step quantile regression model. The default when one quantile is specified in the `quantile(#)` option is `qreg`. When more than one quantile is specified, the user-written `qrprocess` command is the default, and the user cannot choose the `qreg` option. To get the `qreg` estimator with more than one quantile, add option `options_qrprocess(method(qreg))`.

`options_step1(string)` passes options along to the first-step regression model.

`options_qreg(string)` passes options along to the second-step `qreg` command.

`options_qrprocess(string)` passes options along to the second-step `qrprocess` command.

`options_predict(string)` passes options along to the `predict` command that is carried out after the first-step regression. The default is `residuals`.

`generate_r(varname)` saves a variable containing the residuals from the first-step regression.

`smoothing(a,b)` adds uniformly distributed noise over the interval $[a,b]$ to the outcome variable.

`print1step` displays the first-step regression.

4.3 Getting started

To get started, download the `rqr` package from the SSC Archive using `ssc install`, which will download both the `rqr` and the `rqrplot` commands:

```
ssc install rqr
```

To use all the functionalities of the `rqr` command, you should also download the `qrprocess` (Chernozhukov et al. 2020b) and `reghdfe` (Correia 2016) commands. The `rqr` command will issue an error message if the user specifies multiple quantiles or fixed effects and these commands are missing and provide hyperlinks to download the needed command.

```
ssc install qrprocess
ssc install reghdfe
```

Finally, to replicate all of the examples in this paper, you need to download the `estout` package (Jann 2005c, 2007), the `moremata` package (Jann 2005b), the `coefplot` command (Jann 2014), the `kdens` command (Jann 2005a), the `ivqte` command (Frölich and Melly 2010), and the `genqreg` command (written by Matthew J. Baker, David Powell, and Travis Smith). None of these last commands are needed to use the `rqr` command.

```
ssc install estout
ssc install moremata
ssc install kdens
ssc install coefplot
net install st0203, from(http://www.stata-journal.com/software/sj10-3/)
ssc install genqreg
```

4.4 Single quantile

Let us build upon the union wage example discussed above. With effects on only one single quantile to be estimated, the RQR treatment coefficient is estimated by writing the command name `rqr` followed by the outcome variable `wage` and the treatment variable `union`. The quantile is written in the `quantile(#)` option while the control variables are written in the `controls(varlist)` option.

```
. rqr wage union, quantile(.50) controls(i.year c.age#c.age grade south i.ind_code)
```

Residualized Quantile Regression
Quantiles: .50

Number of obs = 19083

	Coefficient	Std. err.	t	P> t	[95% conf. interval]	
wage						
union	.8049713	.0565187	14.24	0.000	.6941898	.9157529
_cons	5.66309	.0227086	249.38	0.000	5.61858	5.707601

Control variables: i.year age c.age#c.age grade south i.ind_code

Algorithm: `qreg`

In addition to the RQR coefficients, the output displays the included control variables below the table along with the command used to estimate the second step. With only one quantile, the Stata algorithm in the official `qreg` command is used as the default in the second step.

In the following examples, we will store the list of control variables in the global macro `x` to simplify the syntax.

```
. global x i.year c.age#c.age grade south i.ind_code
```

4.5 Multiple quantiles

When more than one quantile is requested in the `quantile(#)` option, the `rqr` command draws upon the fast algorithms in the user-written `qrprocess` command. All shorthands described in `numlist` are allowed, including `quantile(.1(.1).9)`.

```
. rqr wage union, quantile(.25 .50 .75) controls($x)
```

Residualized Quantile Regression Number of obs = 19083
Quantiles: .25 .50 .75

	wage	Coefficient	Std. err.	t	P> t	[95% conf. interval]	
Q.25							
	union	.6150905	.0497095	12.37	0.000	.5176554	.7125255
	_cons	4.203465	.0173834	241.81	0.000	4.169392	4.237538
Q.5							
	union	.8049713	.062095	12.96	0.000	.6832596	.926683
	_cons	5.66309	.0235365	240.61	0.000	5.616957	5.709224
Q.75							
	union	1.057971	.0885123	11.95	0.000	.8844787	1.231463
	_cons	7.775508	.0380307	204.45	0.000	7.700964	7.850052

Control variables: i.year age c.age#c.age grade south i.ind_code

Algorithm: Frisch-Newton interior point with preprocessing (from `qrprocess`)

4.6 Fixed effects

Fixed effects can be easily included in the RQR model by adding the `absorb(varname)` option.

```
. rqr wage union, quantile(.5) controls($x) absorb(idcode)
```

Residualized Quantile Regression Number of obs = 19083
Quantiles: .5

	wage	Coefficient	Std. err.	t	P> t	[95% conf. interval]	

union	.4962855	.0919584	5.40	0.000	.316039	.676532
_cons	5.644104	.0241691	233.53	0.000	5.596731	5.691478

Control variables: i.year age c.age#c.age grade south i.ind_code

Fixed effects: idcode (absorbed in first step using areg)

Algorithm: greg

With only one included fixed effects, the `rqr` command uses `areg` in the first step as a default. With more than one fixed effects, the default is to use the user-written `reghdfe` command.

```
. rqr wage union, quantile(.5) controls($x) absorb(idcode occ_code)
```

Residualized Quantile Regression

Number of obs = 18423

Quantiles: .5

wage	Coefficient	Std. err.	t	P> t	[95% conf. interval]	
union	.4850159	.0974893	4.98	0.000	.2939279	.6761039
_cons	5.66976	.0259306	218.65	0.000	5.618934	5.720587

Control variables: i.year age c.age#c.age grade south i.ind_code

Fixed effects: idcode occ_code (absorbed in first step using reghdfe)

Algorithm: greg

Since including fixed effects groups with only one observation is computationally inefficient and overstates statistical significance, the `reghdfe` command excludes such singleton groups by default (Correia 2015). That is why the number of observations is lower in the multiple fixed effects model above. This default behavior can be overridden by adding the option `options_step1(keepsingletons)`.

```
rqr wage union, quantile(.5) controls($x) absorb(idcode occ_code)          ///
    options_step1(keepsingletons)
```

4.7 Table with RQR coefficients

The `rqr` command can be combined with `estimates store` or the user-written `eststo` command to create regression tables. Below we show an example where `eststo` is combined with the `esttab` commands.

```
. eststo clear
. eststo: qui rqr wage union, quantile(.25 .5 .75)
(est1 stored)
. eststo: qui rqr wage union, quantile(.25 .5 .75) controls($x)
(est2 stored)
. eststo: qui rqr wage union, quantile(.25 .5 .75) controls($x) absorb(idcode)
(est3 stored)
. esttab, b(4) se(4) keep(union) nomtitles
```

	(1)	(2)	(3)
--	-----	-----	-----

Q.25

union	1.0403*** (0.0463)	0.6150*** (0.0497)	0.4822*** (0.0641)
Q.5 union	1.4603*** (0.0632)	0.8050*** (0.0621)	0.4963*** (0.0842)
Q.75 union	1.8486*** (0.0859)	1.0580*** (0.0885)	0.5587*** (0.1438)
N	19083	19083	19083

Standard errors in parentheses
 * p<0.05, ** p<0.01, *** p<0.001

4.8 Standard errors

Bootstrapping should be used to get the correct RQR standard errors, where both the first step residualization and the second step QTE estimation should be performed in each bootstrap sample. Bootstrapping the two-step RQR procedure is easily achieved by applying the `bootstrap` prefix.

```
bootstrap, reps(100): rqr wage union, quantile(.25 .50 .75) controls($x)
```

Along with the bootstrapped SEs, the output reports the normal-approximation bootstrapped CIs. However, we can access the percentile CIs, the bias-corrected CIs, and if the `bca` option is evoked, the bias-corrected and accelerated CIs and use these in for example graphs, as we will show later.

4.9 Other options

See Appendix A for examples of how to decide the second-step estimator (section A.1), how to change confidence levels (section A.2), how to show first-step regression (section A.3), how to generate residuals (section A.4), and how to smooth the outcome variable (section A.5).

5 The `rqrplot` command

The `rqrplot` command is a postestimation command that plots RQR coefficients after `rqr`.

5.1 Syntax

```
rqrplot [ , bopts(string) ciopts(string) twopts(string) level(#)
         bootstrap(string) nodraw notabout noci ]
```

5.2 Options

bopts(*string*) allows for customizing the display of the coefficients. The default is solid line graph. See the **twoway** help file for other options.

ciopts(*string*) allows for customizing the confidence intervals. The default is range plot with shading. See the **twoway** help file for other options.

twopts(*string*) allows for customizing the overall graph, including titles and labels. See the **twoway** help file for other options.

level(*#*) decides the confidence level of the confidence intervals. The default is 95% confidence interval. Option cannot be combined with percentile bootstrap and bias-corrected bootstrap CIs, for which the level is decided in the **bootstrap** prefix command.

bootstrap(*string*) requests normal-approximation bootstrap CIs (**bootstrap**(**normal**)), percentile bootstrap CI (**bootstrap**(**percentile**)), or bias-corrected bootstrap CI (**bootstrap**(**bc**)). The default is normal-approximation when **rqr** is estimated with the **bootstrap** prefix.

nodraw suppresses the display of the **twoway** plot.

notabout suppresses the display of the result matrix.

noci plots the coefficients without confidence intervals.

5.3 Basic usage

To use the **rqrplot** command, it suffices to write the command without any options, which will produce the following graph.

```
. quietly rqr wage union, quantile(.05(.05).95) controls($x)
. rqrplot

Plot RQR coefficients
Outcome: wage
Treatment: union
Confidence bands: 95%

      b          se          ll          ul
0.05  .43644813   .0486096   .34116903   .53172725
0.10  .3929728   .04541406   .30395722   .48198837
0.15  .4795796   .04701557   .38742492   .57173425
0.20  .61580986   .04852946   .52068782   .7109319
0.25  .61509049   .04970952   .51765543   .71252555
0.30  .62172371   .05160542   .52057254   .72287488
0.35  .63116586   .05432801   .52467817   .73765355
0.40  .71506602   .05781282   .60174781   .82838422
0.45  .76335907   .06028758   .64519006   .88152808
0.50  .80497134   .062095    .68325967   .92668301
0.55  .87080115   .06422555   .7449134    .9966889
0.60  .96365196   .06742769   .83148772   1.0958161
0.65  1.0137411   .07192158   .87276852   1.1547137
0.70  1.044402    .07904222   .88947225   1.1993318
```

0.75	1.0579706	.08851229	.88447875	1.2314626
0.80	1.0400456	.09453984	.85473919	1.225352
0.85	.92590058	.1046166	.72084278	1.1309583
0.90	.66828531	.12547377	.42234564	.91422498
0.95	.7085585	.19972865	.31707269	1.1000443

The output shows the coefficient, standard error, and upper and lower confidence bands. The table can be suppressed using the `notabout` option and the graph can be suppressed using the `nodraw` option, see Appendix section A.6 for an example.

5.4 Customize graph

It is easy to customize the graph using the `bopts()`, `ciopts()`, and `twopts()`. Consider the following example:

```
rqr wage union, quantile(.03(.02).97) controls($x)
rqrplot, bopts(recast(connected) lcolor(navy) mcolor(navy) msymbol(vsmall)) ///
          ciopts(color(navy%40) lcolor(navy%40)) ///
          twopts(title(Union wage effects))
```

The options in `bopts()` control the outline of the line that reports coefficients, while the `ciopts()` customize how the CIs are reported. Finally, the overall `twopts()` customize the overall look of the graph, including titles. In this particular case, we have added the `recast(connected)` option to recast the `twoway` line graph to `twoway connected` graph, changed the color of the lines, and added a graph title.

If we want to suppress the confidence intervals, we need to include the `noci` option. Moreover, we can include the `recast()` option to change the default area CIs to lines (`recast(rline)`) or caps (`recast(rcap)`).

```
rqrplot, twopts(title(Panel a) name(m1, replace) ylab(,nogrid) xlab(, nogrid))
rqrplot, ciopts(recast(rcap)) twopts(name(m2, replace) title(Panel b)
      ylab(,nogrid) xlab(, nogrid) nodraw) ///
rqrplot, ciopts(recast(rline)) twopts(name(m3, replace) title(Panel c)
      ylab(,nogrid) xlab(, nogrid) nodraw) ///
rqrplot, noci twopts(name(m4, replace) title(Panel d)
      ylab(,nogrid) xlab(, nogrid) nodraw) ///
graph combine m1 m2 m3 m4, title(Union wage effects)
```

When the `rqrplot` postestimation command is used after `rqr` with the `bootstrap` prefix, the default is to display normal-approximation bootstrapped CIs. However,

percentile or bias-corrected CIs can be chosen by using the `bootstrap(percentile)` or `bootstrap(bc)` option in the `rqrplot` command.

```
bootstrap, reps(100): rqr wage union, quantile(.03(.02).97) controls($x)
rqrplot
```

There are two important notes concerning confidence levels. First, the `level()` option in the `rqr` command is not automatically passed on to the `rqrplot` command. The default confidence level in the `rqrplot` command is 95% CIs, unless percentile or bias-corrected bootstrap CIs are requested (see below). To get other confidence levels than the default, we need to include the `level(#)` option in the `rqrplot` command:

```
rqr wage union, quantile(.03(.01).97) controls($x)
rqrplot, level(70)
```

Second, note that the `level(#)` option does not work when requesting percentile or bias-corrected CIs in `rqrplot`. In that case, the level must be written directly in the `bootstrap` prefix command, as shown below.

```
bootstrap, reps(5) level(75): rqr wage union, quantile(.03(.01).97) controls($x)
rqrplot, bootstrap(percentile)
```

5.5 Combine results

The `rqrplot` command allows for plotting only one variable at a time. However, the command stores the result matrix in `r()`, including the RQR coefficients and the confidence intervals. The `r(plotmat)` matrix can be used to combine coefficients from different models. See Appendix section A.7 for an example. Another option is to use the user-written `coefplot` command, as explained in Appendix section A.8, which also includes an example using so-called regression coefficient plot.

6 Discussion

In this paper, we have introduced the `rqr` and `rqrplot` commands that estimate and plot RQR coefficients. By estimating unconditional QTEs, the `rqr` command is similar in spirit to the existing `ivqte` and `genqreg` commands. See Borgen et al. (2021) for a systematic comparison of the three QTE approaches and Appendix section B.2 for a comparison using the union wage example in this paper.

The `rqr` command's main contribution to the Stata toolkit is by allowing for including high-dimensional fixed effects. Fixed effects are often problematic to include in propensity score matching approaches, and the `ivqte` approach is accordingly generally not suited for fixed effects. Although the `genqreg` was not developed to handle a large number of fixed effects, fixed effects can be included as proneness variables in that command. However, this computationally-intensive solution is unfeasible when the number of fixed effects gets large.

To illustrate computational speed differences, consider the union wage example used

in this paper. The sample size consists of 4,135 units and in total 19,138 unit-year observations. When fixed effects are included, estimating treatment effects at one single quantile with the `genqreg` command takes 3,743 seconds (more than one hour) and 99 quantiles take 20 hours. In contrast, the `rqr` command can estimate the treatment effects for 99 quantiles in less than 6 seconds (compare timers 5 and 7 in Appendix section B.1). Moreover, the `genqreg` command will break down if the number of fixed effects increases further.

Even without fixed effects, the `rqr` command is considerably faster than the `genqreg` and `ivqte` commands. The `rqr` command estimates the 99 quantiles in 5 seconds, about ten times faster than the `genqreg` command and five times faster than the `ivqte` command. The computational speed of the `rqr` is partly owing to the fast algorithms developed for Stata by Chernozhukov et al. (2020b), but not exclusively. Even when applying the built-in `qreg` command, the `rqr` model is clearly faster than the `ivqte` and `genqreg` commands.

The main limitation of the `rqr` command compared to the `ivqte` and `genqreg` commands is that it currently does not allow for instrumental variables. However, we will demonstrate that endogenous treatment variables can be handled within the RQR framework and make this approach available to the Stata community in future work. The general idea is that the instrument rather than the treatment is regressed on the control variables in the first step, and the instrument residuals are obtained. Then, in the second step, QTEs are estimated using a method of moments approach (Machado and Silva 2019; Kaplan and Sun 2017; Chernozhukov and Hansen 2005), where the control variables are redundant because of the residualized instrument.

7 Acknowledgments

The contribution of Nicolai T. Borgen was financed by a European Research Council grant (#818425).

8 References

- Borgen, N. T. 2016. Fixed effects in unconditional quantile regression. *The Stata Journal* 16(2): 403–415.
- Borgen, N. T., A. Haupt, and Ø. N. Wiborg. 2020. Quantile Regression Models and the Crucial Difference Between Individual-Level Effects and Population-Level Effects. *SocArXiv*.
- Borgen, N. T., A. Haupt, Ø. N. Wiborg, et al.. 2021. A New Framework for Estimation of Unconditional Quantile Treatment Effects: The Residualized Quantile Regression (RQR) Model. *SocArXiv*.
- Bottai, M., and N. Orsini. 2019a. qmodel: A command for fitting parametric quantile models. *The Stata Journal* 19(2): 261–293.

- . 2019b. qmodel: A command for fitting parametric quantile models. *The Stata Journal* 19(2): 261–293.
- Budig, M. J., and M. J. Hodges. 2010. Differences in disadvantage: Variation in the motherhood penalty across white women’s earnings distribution. *American Sociological Review* 75(5): 705–728.
- Card, D. 1996. The Effect of Unions on the Structure of Wages: A Longitudinal Analysis. *Econometrica* 64(4): 957–979. <http://www.jstor.org/stable/2171852>.
- Chernozhukov, V., I. Fernández-Val, and B. Melly. 2020a. Fast algorithms for the quantile regression process. *Empirical economics* 1–27.
- . 2020b. Quantile and distribution regression in Stata: algorithms, pointwise and functional inference. *Unpublished paper*.
- Chernozhukov, V., and C. Hansen. 2005. An IV model of quantile treatment effects. *Econometrica* 73(1): 245–261.
- Correia, S. 2015. Singletons, cluster-robust standard errors and fixed effects: A bad mix. *Technical Note, Duke University* 7.
- . 2016. Linear Models with High-Dimensional Fixed Effects: An Efficient and Feasible Estimator. Technical report. Working Paper.
- Firpo, S. 2007. Efficient semiparametric estimation of quantile treatment effects. *Econometrica* 75(1): 259–276.
- Firpo, S., N. M. Fortin, and T. Lemieux. 2009. Unconditional quantile regressions. *Econometrica* 77(3): 953–973.
- Fortin, N., T. Lemieux, and S. Firpo. 2011. Decomposition methods in economics. In *Handbook of labor economics*, 1–102. Vol. 4. Elsevier.
- Frölich, M., and B. Melly. 2010. Estimation of quantile treatment effects with Stata. *The Stata Journal* 10(3): 423–457.
- . 2013. Unconditional quantile treatment effects under endogeneity. *Journal of Business & Economic Statistics* 31(3): 346–357.
- Jann, B. 2005a. KDENS: Stata module for univariate kernel density estimation. Statistical Software Components, Boston College Department of Economics. <https://ideas.repec.org/c/boc/bocode/s456410.html>.
- . 2005b. MOREMATA: Stata module (Mata) to provide various functions. Statistical Software Components, Boston College Department of Economics. <https://ideas.repec.org/c/boc/bocode/s455001.html>.
- . 2005c. Making regression tables from stored estimates. *The Stata Journal* 5(3): 288–308.

- . 2007. Making regression tables simplified. *The Stata Journal* 7(2): 227–244.
- . 2014. Plotting regression coefficients and other estimates. *The Stata Journal* 14(4): 708–737.
- Kaplan, D. M. 2020. `sivqr`: Smoothed IV quantile regression. Working Papers 2009, Department of Economics, University of Missouri. <https://ideas.repec.org/p/umc/wpaper/2009.html>.
- Kaplan, D. M., and Y. Sun. 2017. Smoothed estimating equations for instrumental variables quantile regression. *Econometric Theory* 33(1): 105–157.
- Killewald, A., and J. Bearak. 2014. Is the motherhood penalty larger for low-wage women? A comment on quantile regression. *American Sociological Review* 79(2): 350–357.
- Koenker, R. 2005. *Quantile Regression*. Econometric Society Monographs, Cambridge University Press.
- Machado, J. A., and J. S. Silva. 2018a. IVQREG2: Stata module to provide structural quantile function estimation. Statistical Software Components, Boston College Department of Economics. <https://ideas.repec.org/c/boc/bocode/s458571.html>.
- . 2019. Quantiles via moments. *Journal of Econometrics* 213(1): 145–173.
- Machado, J. A. F., and J. S. Silva. 2005. Quantiles for counts. *Journal of the American Statistical Association* 100(472): 1226–1237.
- . 2018b. XTQREG: Stata module to compute quantile regression with fixed effects. Statistical Software Components, Boston College Department of Economics. <https://ideas.repec.org/c/boc/bocode/s458523.html>.
- Mastekaasa, A. 2013. Unionization and Certified Sickness Absence: Norwegian Evidence. *ILR Review* 66(1): 117–141.
- Powell, D. 2020. Quantile treatment effects in the presence of covariates. *Review of Economics and Statistics* 102(5): 994–1005.
- Rios-Avila, F. 2020. Recentered influence functions (RIFs) in Stata: RIF regression and RIF decomposition. *The Stata Journal* 20(1): 51–94.
- Wenz, S. E. 2019. What quantile regression does and doesn't do: A commentary on Petscher and Logan (2014). *Child development* 90(4): 1442–1452.

About the authors

Nicolai T. Borgen is a researcher at the EQOP project at the University of Oslo, where he studies the causes of social inequality in children's language development and school achievements.

Andreas Haupt is lecturer for Sociology (Akademischer Rat) at the Karlsruhe Institute of Technology. His research concerns occupation-specific labor market processes, gender inequality, poverty and income richness.

Øyvind Wiborg is Associate Professor at the Department of Sociology and Human Geography at the University of Oslo.

Appendix A: Additional *rqr* and *rqrplot* options

A.1 Choose second-step *rqr* estimator

The user could choose whether the second step command is *qreg* or *qrprocess* using the *step2command* option:

```
. rqr wage union, quantile(.5) controls($x) step2command(qrprocess)
```

Residualized Quantile Regression Number of obs = 19083
Quantiles: .5

	wage	Coefficient	Std. err.	t	P> t	[95% conf. interval]	
Q.5							
	union	.8049797	.062095	12.96	0.000	.6832681	.9266914
	_cons	5.663082	.0235365	240.61	0.000	5.616949	5.709216

Control variables: i.year age c.age#c.age grade south i.ind_code
Algorithm: Frisch-Newton interior point with preprocessing (from *qrprocess*)

Moreover, within the *qrprocess* command, the algorithmic method can be chosen by passing on commands to the *qrprocess* command.

```
. rqr wage union, quantile(.1) controls($x) options_qrprocess(method(fn)) ///
> step2command(qrprocess)
```

Residualized Quantile Regression Number of obs = 19083
Quantiles: .1

	wage	Coefficient	Std. err.	t	P> t	[95% conf. interval]	
Q.1							
	union	.3959776	.0453423	8.73	0.000	.3071026	.4848525
	_cons	3.409901	.0146008	233.54	0.000	3.381282	3.43852

Control variables: i.year age c.age#c.age grade south i.ind_code
Algorithm: Frisch-Newton interior point (from *qrprocess*)

Note that the *rqr* command does not allow the second-step quantile regression command to be *qreg* when more than one quantile is specified. To get the built-in Stata algorithm with more than one quantile, you need to add the *options_qrprocess(method(qreg))* option, as shown in the example below.

```
. rqr wage union, quantile(.25 .50 .75) controls($x) options_qrprocess(method(qreg))
```

Residualized Quantile Regression Number of obs = 19083
Quantiles: .25 .50 .75

	wage	Coefficient	Std. err.	t	P> t	[95% conf. interval]	
Q.25							
	union	.6150905	.0497095	12.37	0.000	.5176554	.7125255
	_cons	4.203465	.0173834	241.81	0.000	4.169392	4.237538
Q.5							

	union	.8049713	.062095	12.96	0.000	.6832596	.926683
	_cons	5.66309	.0235365	240.61	0.000	5.616957	5.709224
<hr/>							
Q.75	union	1.057971	.0885123	11.95	0.000	.8844787	1.231463
	_cons	7.775508	.0380307	204.45	0.000	7.700964	7.850052

Control variables: i.year age c.age#c.age grade south i.ind_code
 Algorithm: qreg (from qrprocess)

A.2 Confidence level

The default 95% confidence level could be changed using the options passed on to the second step QR estimator. In the following example, we request 75% confidence intervals.

```
. rqr wage union, quantile(.5) controls($x) options_qreg(level(75))
```

Residualized Quantile Regression Number of obs = 19083
 Quantiles: .5

	wage	Coefficient	Std. err.	t	P> t	[75% conf. interval]	
	union	.8049713	.0565187	14.24	0.000	.7399531	.8699895
	_cons	5.66309	.0227086	249.38	0.000	5.636967	5.689214

Control variables: i.year age c.age#c.age grade south i.ind_code
 Algorithm: qreg

A.3 Show first step regression

Sometimes seeing the first-step regression could be useful. This can be achieved by adding the `print1step` option:

```
. rqr wage union, quantile(.5) controls($x) print1step
```

Source	SS	df	MS	Number of obs	=	19,083
Model	344.289968	26	13.2419219	F(26, 19056)	=	81.92
Residual	3080.25177	19,056	.161642095	Prob > F	=	0.0000
				R-squared	=	0.1005
				Adj R-squared	=	0.0993
Total	3424.54174	19,082	.179464508	Root MSE	=	.40205

	union	Coefficient	Std. err.	t	P> t	[95% conf. interval]	
year							
71		.018405	.0195073	0.94	0.345	-.019831	.056641
72		-.0101265	.0183796	-0.55	0.582	-.0461521	.0258992
73		.0011525	.0191724	0.06	0.952	-.0364272	.0387321
77		-.0390011	.0184827	-2.11	0.035	-.0752288	-.0027734
78		.0121601	.0202757	0.60	0.549	-.0275821	.0519023
80		.0550853	.0205532	2.68	0.007	.0147992	.0953715
82		-.0000631	.0209669	-0.00	0.998	-.04116	.0410338
83		-.0286947	.0216576	-1.32	0.185	-.0711455	.0137561
85		-.0003936	.022127	-0.02	0.986	-.0437644	.0429773

87	-.0358374	.0228739	-1.57	0.117	-.0806722	.0089974
88	-.0205444	.0237587	-0.86	0.387	-.0671136	.0260247
age	.0024213	.0057241	0.42	0.672	-.0087984	.013641
c.age#c.age	6.86e-06	.0000881	0.08	0.938	-.0001659	.0001796
grade	.0029041	.0013519	2.15	0.032	.0002542	.005554
south	-.109361	.005994	-18.24	0.000	-.1211098	-.0976122
ind_code						
2	-.0048595	.0741743	-0.07	0.948	-.1502476	.1405286
3	.0431706	.0460504	0.94	0.349	-.0470923	.1334336
4	.2563112	.0358922	7.14	0.000	.1859593	.3266631
5	.491358	.0374438	13.12	0.000	.4179648	.5647511
6	.0581308	.0360734	1.61	0.107	-.0125762	.1288378
7	.0146515	.0366234	0.40	0.689	-.0571337	.0864367
8	.0343703	.0391389	0.88	0.380	-.0423454	.111086
9	.061825	.0376918	1.64	0.101	-.0120543	.1357043
10	.0541964	.0496313	1.09	0.275	-.0430854	.1514782
11	.2062141	.035792	5.76	0.000	.1360587	.2763696
12	.2574015	.0370433	6.95	0.000	.1847935	.3300096
_cons	-.0117578	.093409	-0.13	0.900	-.1948477	.1713321

Residualized Quantile Regression Number of obs = 19083
 Quantiles: .5

wage	Coefficient	Std. err.	t	P> t	[95% conf. interval]	
union	.8049713	.0565187	14.24	0.000	.6941898	.9157529
_cons	5.66309	.0227086	249.38	0.000	5.61858	5.707601

Control variables: i.year age c.age#c.age grade south i.ind_code
 Algorithm: qreg

A.4 Generate residuals

The `rqr` command stores the residuals from the first-step regression in temporary variables, which automatically ceases to exist after the second-step regression. To store the residuals, we can add the option `generate_r(varname)`.

```
. cap drop resid
. rqr wage union, quantile(.5) controls($x) generate_r(resid)
```

Residualized Quantile Regression Number of obs = 19083
 Quantiles: .5

wage	Coefficient	Std. err.	t	P> t	[95% conf. interval]	
union	.8049713	.0565187	14.24	0.000	.6941898	.9157529
_cons	5.66309	.0227086	249.38	0.000	5.61858	5.707601

Control variables: i.year age c.age#c.age grade south i.ind_code
 Algorithm: qreg

A.5 Smoothing the outcome variable

With a discrete outcome variable, we can smooth the outcome variable using the approach suggested by Machado and Silva (2005).

```
. eststo clear
. eststo: qui rqr wage union, quantile(.25 .50 .75) controls($x)
(est1 stored)
. eststo: qui rqr wage union, quantile(.25 .50 .75) controls($x) smoothing(-.5,.5)
(est2 stored)
. esttab, b(4) se(4) keep(union) mtitle("Normal" "Smoothed")
```

	(1) Normal	(2) Smoothed
Q.25 union	0.6151*** (0.0497)	0.5772*** (0.0498)
Q.5 union	0.8050*** (0.0621)	0.8190*** (0.0622)
Q.75 union	1.0580*** (0.0885)	1.0125*** (0.0884)
N	19083	19083

Standard errors in parentheses
* p<0.05, ** p<0.01, *** p<0.001

A.6 Suppress output in rqrplot

The `rqrplot` by default prints the result matrix and generates a graph, both of which can be suppressed by adding the `notabout` and `nodraw` options. These options will only suppress the display of the graph and result matrix in the results window; the `rqrplot` will still create the result matrix and put it in `r()` and create a graph in the background.

```
. rqrplot, nodraw notabout

Plot RQR coefficients
Outcome: wage
Treatment: union
Confidence bands: 95%
```

A.7 Work with result matrix after rqrplot

The `rqrplot` returns the matrix `r(plotmat)`, which the author can access and use. Below is an example where the RQR coefficients are estimated without any controls (m1), with individual-level controls (m2), and with individual-level controls and fixed effects (m3), and then plotted in one graph.

```

rqr wage union, quantile(.03(.02).97)
rqrplot, nodraw notabout
matrix m1=r(plotmat)

rqr wage union, quantile(.03(.02).97) controls($x)
rqrplot, nodraw notabout
matrix m2=r(plotmat)

rqr wage union, quantile(.03(.02).97) controls($x) absorb(idcode)
rqrplot, nodraw notabout
matrix m3=r(plotmat)

frame create b
frame b {
    svmat m1
    svmat m2
    svmat m3
    tw (line m12 m22 m32 m11, lcolor(turquoise sea orangebrown))
        (rarea m14 m15 m11, color(turquoise%40))
        (rarea m24 m25 m11, color(sea%40))
        (rarea m34 m35 m11, color(orangebrown%40))
        , legend(order(1 "No controls" 2 "Individual controls"
            3 "+ fixed effects")) title(Union wage effects)
        xtitle(Quantile)
}
frame drop b

```

A.8 Use coefplot after rqr

We can also use `coefplot` to plot results after `rqr`. The `coefplot` command is handy if we want to show so-called regression coefficient plots or if we want to combine the results from multiple models. Let us show an example of combining results from multiple models first, followed by an example of coefficients plots.

To combine results from multiple models, we start by estimating the regression models and storing the estimates using `eststo`.

```
eststo m1: rqr wage union, quantile(.10(.10).90)
eststo m2: rqr wage union, quantile(.10(.10).90) controls($x)
eststo m3: rqr wage union, quantile(.10(.10).90) controls($x) absorb(idcode)
```

Then we can use `coefplot` to plot the results from the three different models, similar to what we did in section A.7 above.

```
coefplot (m1, mcolor(turquoise) lcolor(turquoise) ciopts(color(turquoise%40))) ///
(m2, mcolor(sea) lcolor(sea) ciopts(color(sea%40))) ///
(m3, mcolor(orangebrown) lcolor(orangebrown) ciopts(color(orangebrown%40))) ///
, nooffsets keep(Q.*:union) vertical ///
legend(order(2 "No controls" 4 "Individual controls" ///
6 "+ fixed effects")) recast(connect) cirecast(rarea) ///
nolabels rename(*union = " ") ///
eqrename(Q* = " ") btitle(Quantile)
```

We can also use `coefplot` to create a coefficient plot, either based on one model or

multiple models. Below is one example of such use.

```
coefplot m1 m2 m3 , nooff keep(Q.*:union)      ///
      legend(order(2 "No controls" 4 "Individual controls" 6 "+ fixed effects")) nolabels rename(*union = " ") ///
      title(Union wage effects)                ///
      eqrename(Q* = " ") l1title(Quantiles) b1title(Coefficients)
```

Appendix B: Comparing `rqr`, `genqreg`, and `ivqte`

B.1 Computational speed

To begin, we look at computational speed in a case without fixed effects. We see that in a dataset containing 19,138 individuals, the RQR model estimated using the `qrprocess` algorithm take only 5.6 seconds (timer 1). The same RQR coefficients take 249% longer with the `qreg` algorithm. The `ivqte` model takes 410% more time (timer 4) and the `genqreg` command takes 855% longer (timer 3).

However, the largest difference are when including high-dimensional fixed effects. While the RQR model can estimate the fixed effects model for 99 quantiles in only 5.9 (with `qrprocess`) and 24.8 (with `qreg`) seconds, the `genqreg` command needs 3743.3 seconds for one single quantile (timer 7), which is a huge 63725% more than `rqr`. If we were to use the `genqreg` command to estimate effects across 19 quantiles (e.g., from 5th to 95th percentile at steps of 5), it may take 71123 seconds (i.e., 1185 minutes or 20 hours).

```

tab1 year, gen(year_)
drop year_1
gen age2=age*age
tab1 ind_code, gen(ind_code_)
drop ind_code_1

global d year_* south ind_code_*
global c age age2 grade

timer clear

* RQR with qrprocess
set seed 19081
timer on 1
qui rqr wage union, quantile(.01(.01).999) controls($x)
timer off 1

* RQR with qreg
timer on 2
forvalues i=1(1)99 {
    local i=`i'/100
    qui rqr wage union, quantile(`i') controls($x)
}
timer off 2

* GQR
timer on 3
forvalues i=1(1)99 {
    local i=`i'/100
    qui genqreg wage union, quantile(`i') proneness($d $c)
}
timer off 3

* PS-QTE
timer on 4
ivqte wage (union), quantile(.01(.01).999) dummy($d) continuous($c) variance
timer off 4

* RQR with qrprocess and fe (99 quantiles)
set seed 19081
timer on 5
rqr wage union, quantile(.01(.01).999) controls($x) absorb(idcode)

```

```

timer off 5
* RQR with qreg
timer on 6
forvalues i=1(1)99 {
    local i=`i'/100
    qui rqr wage union, quantile(`i') controls($x) absorb(idcode)
}
timer off 6
rqr wage union, quantile(.5) controls($x) absorb(idcode)
* GQR with fe (1 quantile)
tab1 idcode, gen(i_)
drop i_1
timer on 7
genqreg wage union, quantile(.5) proneness($d $c i_*)
timer off 7

. timer list
1:      5.58 /      1 =      5.5800
2:     19.46 /      1 =     19.4580
3:     53.26 /      1 =     53.2630
4:     28.45 /      1 =     28.4510
5:      5.87 /      1 =      5.8650
6:     24.84 /      1 =     24.8430
7:    3743.31 /      1 =    3743.3090

```

B.2 Coefficient comparison

Below we compare the RQR coefficients to those of GQR and PS-QTE. The RQR and GQR coefficients are nearly identical, while the propensity score approach provides slightly different coefficients in this particular case. This slight difference disappears without the industry dummies

```

frame create b str20 method str20 q b
forvalues i=.03(.02).97 {
    * RQR
    rqr wage union, quantile(`i') controls($d $c)
    frame post b ("rqr") ("`i'") (_b[union])

    * GQR
    genqreg wage union, quantile(`i') pronen($d $c)
    frame post b ("gqr") ("`i'") (_b[union])

    * PS-QTE
    ivqte wage (union), quantile(`i') d($d) c($c)
    frame post b ("ivqte") ("`i'") (_b[Quantile_1])
}

frame b {
    reshape wide b, i(q) j(method) string
    destring q, replace
    tw connected brqr bgqr bivqte q, xtitle(Quantile)
    legend(order(1 "RQR" 2 "GQR" 3 "PS-QTE" ))
    color(navy orangebrown turquoise)
    msize(vsmall vsmall vsmall ) name(qte, replace)
}

```

```
}  
frame drop b
```