The Stata Journal

The *Stata Journal* publishes reviewed papers together with shorter notes or comments, regular columns, book reviews, and other material of interest to Stata users. Examples of the types of papers include 1) expository papers that link the use of Stata commands or programs to associated principles, such as those that will serve as tutorials for users first encountering a new field of statistics or a major new technique; 2) papers that go "beyond the Stata manual" in explaining key features or uses of Stata that are of interest to intermediate or advanced users of Stata; 3) papers that discuss new commands or Stata programs of interest either to a wide spectrum of users (e.g., in data management or graphics) or to some large segment of Stata users (e.g., in survey statistics, survival analysis, panel analysis, or limited dependent variable modeling); 4) papers analyzing the statistical properties of new or existing estimators and tests in Stata; 5) papers that could be of interest or usefulness to researchers, especially in fields that are of practical importance but are not often included in texts or other journals, such as the use of Stata in managing datasets, especially large datasets, with advice from hard-won experience; and 6) papers of interest to those who teach, including Stata with topics such as extended examples of techniques and interpretation of results, simulations of statistical concepts, and overviews of subject areas.

The *Stata Journal* is indexed and abstracted by *CompuMath Citation Index*, *Current Contents/Social and Behavioral Sciences*, *RePEc: Research Papers in Economics*, *Science Citation Index Expanded* (also known as *SciSearch*), *Scopus*, and *Social Sciences Citation Index*.

For more information on the *Stata Journal*, including information for authors, see the webpage

http://www.stata-journal.com

Subscriptions are available from StataCorp, 4905 Lakeway Drive, College Station, Texas 77845, telephone 979-696-4600 or 800-782-8272, fax 979-696-4601, or online at

http://www.stata.com/bookstore/sj.html

Subscription rates listed below include both a printed and an electronic copy unless otherwise mentioned.

U.S. and Canada		Elsewhere	
Printed & electronic		**Printed & electronic**	
1-year subscription	$115	1-year subscription	$145
2-year subscription	$210	2-year subscription	$270
3-year subscription	$285	3-year subscription	$375
1-year student subscription	$ 85	1-year student subscription	$115
1-year institutional subscription	$345	1-year institutional subscription	$375
2-year institutional subscription	$625	2-year institutional subscription	$685
3-year institutional subscription	$875	3-year institutional subscription	$965
Electronic only		**Electronic only**	
1-year subscription	$ 85	1-year subscription	$ 85
2-year subscription	$155	2-year subscription	$155
3-year subscription	$215	3-year subscription	$215
1-year student subscription	$ 55	1-year student subscription	$ 55

Back issues of the *Stata Journal* may be ordered online at

http://www.stata.com/bookstore/sjj.html

Individual articles three or more years old may be accessed online without charge. More recent articles may be ordered online.

http://www.stata-journal.com/archives.html

The *Stata Journal* is published quarterly by the Stata Press, College Station, Texas, USA.

Address changes should be sent to the *Stata Journal*, StataCorp, 4905 Lakeway Drive, College Station, TX 77845, USA, or emailed to sj@stata.com.

The Stata Journal (2016)
16, Number 2, pp. 403–415

Fixed effects in unconditional quantile regression

Nicolai T. Borgen
Department of Sociology and Human Geography
University of Oslo
Oslo, Norway
n.t.borgen@sosgeo.uio.no

Abstract. Unconditional quantile regression has quickly become popular after being introduced by Firpo, Fortin, and Lemieux (2009, *Econometrica* 77: 953–973) and is easily implemented using the user-written command `rifreg` by the same authors. However, including high-dimensional fixed effects in `rifreg` is quite burdensome and sometimes even impossible. In this article, I show that when the number of fixed effects is large, the computational speed is massively increased by using `xtreg` rather than `regress` to fit the unconditional quantile regression models. I also introduce the `xtrifreg` command, which should be considered a supplement to `rifreg`. The `xtrifreg` command has many of the same features as `rifreg` but can be used to include a large number of fixed effects, to estimate cluster–robust standard errors, and to estimate cluster–bootstrapped standard errors.

Keywords: st0438, xtrifreg, unconditional quantile regression, fixed effects

1 Introduction

In the last few years, researchers have discussed the usefulness of the conditional quantile regression (CQR) method in social science research. In CQR, the quantiles are defined conditional on the control variables. Thus including control variables not only adjusts for selection bias, but also redefines the quantiles. This redefinition of the quantiles is sometimes advantageous, for instance, when investigating student growth in test scores (Castellano and Ho 2013).

However, in most cases, researchers are not interested in the effects of conditional quantiles. Rather, they want to investigate the effects of a treatment variable on unconditional quantiles (Porter 2015). Thus Firpo, Fortin, and Lemieux (2009) developed the unconditional quantile regression (UQR) model. The advantage of the UQR model is that the quantiles are defined preregression; therefore, the model is not influenced by any right-hand-side variables (Killewald and Bearak 2014). In UQR, one can, for instance, include fixed effects to adjust for selection bias without redefining the quantiles.

However, implementing UQR in Stata with high-dimensional fixed effects (that is, a large number of groups) is either computationally slow or quite burdensome, especially if the researcher wants bootstrapped standard errors. In two recent commentary articles in the *American Sociological Review*, Killewald and Bearak (2014) and Budig and Hodges

(2014) discuss, among other things, how to include a large number of fixed effects in UQR. Killewald and Bearak (2014) use a computationally undemanding approach of demeaning their variables and including the demeaned variables in an ordinary least-squares (OLS) model. This approach introduces some complications regarding, for example, estimation of standard errors (Budig and Hodges 2014; Allison 2009, 18). Even introductory texts to quantile regression do not always acknowledge that when using UQR, the conventional `bootstrap` command will produce incorrect standard errors (for an example, see Porter [2015, 377]).

Budig and Hodges (2014) suggest using the more computationally demanding strategy of including dummy variables, in their case, $N - 1$ person dummy variables. The advantage of this least-squares dummy variables (LSDV) approach is that it is less prone to coding errors. But with high-dimensional fixed effects, the LSDV estimator is very slow (Allison 2009). With large datasets, such as administrative data, the number of fixed effects may also exceed the allowed number of right-hand-side variables (10,998 in Stata/MP and Stata/SE, 798 in Stata/IC); for an example, see Borgen (2015). Demeaning is often preferable to the LSDV estimator because of the incidental-parameter problem (Cameron and Trivedi 2009, 259).[1]

In this article, I demonstrate how to include high-dimensional fixed effects in Stata without any tradeoff between computational speed and ease of implementation. When you have a large number of fixed effects, I suggest a two-step approach based on the intuition in the seminal paper by Firpo, Fortin, and Lemieux (2009). The first step is to obtain the recentered influence function (RIF), which is convenient with the user-written `rifreg` command (Firpo, Fortin, and Lemieux 2009). The second step is to use the `xtreg` command—rather than the `regress` command used in, for instance, `rifreg`—with this RIF as the outcome variable.

The two-step approach I describe may have a few potential pitfalls. I discuss these pitfalls and explain how researchers can avoid them by using the `xtrifreg` command, which I also introduce in this article. Because the `xtrifreg` command is basically a wrapper around `rifreg` and `xtreg`, it makes it more comfortable, reliable, and streamlined to include high-dimensional fixed effects in UQR. This command should be considered a supplement to the `rifreg` command. `xtrifreg` is particularly handy for including bootstrapped standard errors. Unlike the `rifreg` command, it also reports cluster–robust standard errors and cluster–bootstrapped standard errors. In fixed-effects models, using cluster–robust standard errors and bootstrapped standard errors is often advisable (Cameron and Miller 2015).

There are two main advantages to using `xtrifreg` rather than `rifreg`: 1) `xtrifreg` can be used even if the number of fixed effects exceeds 10,997[2], and 2) it is a lot faster than `regress` when the number of fixed effects is large. I will demonstrate

[1] Unlike the demeaning approach (`xtreg`), the LSDV approach (`regress` or `areg`) assumes that the number of fixed effects does not grow with sample size. The estimated variance–covariance matrix, therefore, differs in the LSDV approach compared with the demeaning approach when `vce(cluster` *clustvar*`)` is specified.

[2] Assuming at least one other independent variable, the maximum number of fixed effects in Stata/MP and Stata/SE is $10998 - 1$.

that computational speed is massively increased by using `xtrifreg`. With the current "data revolution" in social sciences, with more big data—such as administrative data (Einav and Levin 2013)—and multiple high-dimensional fixed effects (McCaffrey et al. 2012; Guimarães and Portugal 2010), computational speed is becoming increasingly important.

The remainder of the article is arranged as follows. In section 2, I describe UQR. In section 3, I describe the two-step approach to including fixed effects in UQR. In section 4, I discuss the standard errors in this two-step approach. In section 5, I present the `xtrifreg` command. Finally, in section 6, I conclude.

2 Unconditional quantile regression basics

Researchers can estimate UQR by simply replacing the outcome variable in OLS with the RIF. Firpo, Fortin, and Lemieux (2009) provide a technical introduction, while Porter (2015) and Killewald and Bearak (2014) provide more easily accessible introductions.[3]

RIF is defined as

$$\text{RIF}(Y; q_\tau, F_Y) = q_\tau + \frac{\tau - \mathbb{1}\{Y \leq q_\tau\}}{f_Y(q_\tau)} \tag{1}$$

where q_τ is the value of the outcome variable, Y, at the quantile τ. F_Y is the cumulative distribution function of Y, and $f_Y(q_\tau)$ is the density of Y at q_τ. The indicator function, $\mathbb{1}\{Y \leq q_\tau\}$, identifies whether the value of the outcome variable, Y, for the individual is below q_τ.

Consider the 75th quantile ($\tau = 0.75$). To identify the RIF for this quantile, one needs to 1) estimate the value of the outcome variable, Y, at that quantile, $q_{0.75}$; 2) estimate the density $f_Y(q_{0.75})$ at $q_{0.75}$ using, for instance, kernel methods; and 3) generate a dummy variable, $\mathbb{1}\{Y \leq q_{0.75}\}$, which indicates whether the value of the outcome variable is at or below the value of Y at the 75th quantile, $q_{0.75}$. The resulting RIF is a dummy variable, holding the values $q_{0.75} + \{0.75/f_Y(q_{0.75})\}$ for those above the 75th quantile and the values $q_{0.75} - \{0.25/f_Y(q_{0.75})\}$ for those at or below the 75th quantile. This RIF could serve as the outcome variable in an OLS model (linear probability model), a so-called RIF-OLS (Firpo, Fortin, and Lemieux 2009).[4]

Equation (1) provides two main insights (Porter 2015). First, the transformed outcome variable (the RIF) is defined preregression. Thus, unlike CQR, including any control variables does not change the definition of the quantile. Second, the transformed outcome variable (the RIF) depends heavily on the estimated density, $f_Y(q_\tau)$. It is thus wise to check the sensitivity of the results by using different kernels and bandwidths.

[3] For readers interested in CQR, see Koenker (2005) and Hao and Naiman (2007)

[4] Firpo, Fortin, and Lemieux (2009) also describe two other ways to estimate UQR: RIF-logit and RIF-NP. The differences between the three estimation methods are minor in their application.

3 Including fixed effects in UQR

Fitting UQR models in Stata is made easy by the user-written command `rifreg` (Firpo, Fortin, and Lemieux 2009).[5] This command 1) computes the RIF and 2) includes this RIF as an outcome variable in `regress` along with any right-hand-side variables. In most applications, using `rifreg` is a good choice.[6]

To include fixed effects in `rifreg`, one could simply add the fixed effects as a set of dummy variables. However, in some cases, the number of fixed effects may exceed the number of allowed right-hand-side variables. In Borgen (2015), for instance, the number of sibling fixed effects is 42,860, which makes it impossible to use `rifreg`. More generally, when the number of fixed effects is large, the dummy-variable approach to fixed effects gets burdensome (Allison 2009). Luckily, we can speed up the process by substituting the `regress` command in step 2 with `xtreg`.[7]

I will use a subsample of the National Longitudinal Survey (`nlswork.dta`) to exemplify how to obtain UQR point estimates and standard errors using `xtreg` (see Borgen [2015] for an empirical application). `nlswork.dta` contains information on 4,711 young working women aged 14–26 years in 1968, followed over the years 1968–1988. The total number of observations is 28,453.

To load this dataset, type

```
. use http://www.stata-press.com/data/r14/nlswork
(National Longitudinal Survey.  Young Women 14-26 years of age in 1968)
```

Using `nlswork.dta`, I will estimate the effect of union membership (1 if union member) on log wages at the 50th quantile, with fixed effects on individuals (`idcode`). The model is

$$Y_{it} = \beta_0 + \beta_1 \text{union}_{it} + \alpha_i + \varepsilon_{it}$$

where i indexes individuals and t indexes time, β_0 is the constant term, β_1 is the effect of union membership, α_i are the individual fixed effects, and ε_{it} is the error term. I use this model specification to illustrate how to include fixed effects in UQR, but it is not a correct specification of the causal effect of union membership.

3.1 Obtaining the RIF

In Stata, generating the transformed outcome variable (the RIF) is easy. By using the `retain(`*string*`)` option of `rifreg`, the transformed outcome variable is stored to the dataset.

[5] To install this program, go to http://faculty.arts.ubc.ca/nfortin/datahead.html.

[6] Another useful command is the `ivqte` command by Frölich and Melly (2010), which can implement CQR, the instrumental-variable conditional quantile regression estimator, UQR, and the instrumental-variable unconditional quantile regression estimator.

[7] Using `xtreg` in the second step is similar to the demeaning strategy of Killewald and Bearak (2014) but is easier to implement (Allison 2009).

```
. rifreg ln_wage, quantile(50) retain(q50)
  (output omitted)
```

The `rifreg` command uses the `pctile` command to identify the value of the outcome variable at the 50th quantile ($q_{.50}$) and the `kdensity` command to estimate the density of the outcome variable at the quantile $[f_Y(q_{.50})]$. `rifreg` then includes these values in (1). We could also do this ourselves, without the `rifreg` command, as follows:[8]

```
pctile quantiles=ln_wage, n(100)
kdensity ln_wage, at(quantiles) kernel(gaussian) bwidth(0.0) ///
     generate(quantile density) nodraw
generate indicator=ln_wage<quantile[50]
generate q50=quantile[50]+((.50-indicator)/density[50])
```

When one obtains the transformed outcome variable in a separate first step, there are two potential caveats (neither of which pose a problem when using the `xtrifreg` command, which I will introduce in section 5). First, one should make sure that the sample used to obtain the RIF variable is identical to the sample that will be used in the regression analyses. One could do this by using, for instance, the `marksample` and `markout` commands. If the sample differs, the coefficients and inference will be incorrect.

```
. marksample touse
. markout `touse´ ln_wage union idcode
. rifreg ln_wage if `touse´, quantile(50) retain(q50)
  (output omitted)
```

Second, the `rifreg` command gives all observations that are excluded from the analysis (`e(sample)==0`) the value 0 on the retained variable. In `nlswork.dta`, approximately a third of the observations have missing on the union variable and get the value 0 on the outcome variable.

```
. tabulate q50
        q50 |      Freq.     Percent        Cum.
------------+-----------------------------------
          0 |      9,296       32.58       32.58
   1.155391 |      9,617       33.70       66.28
   2.305247 |      9,621       33.72      100.00
------------+-----------------------------------
      Total |     28,534      100.00
```

It may be wise to replace the 0 values on the retained variable with missing.

```
. replace q50=. if e(sample)!=1
(9,296 real changes made, 9,296 to missing)
```

We are then left with a sample of 19,238 observations on 4,150 women.

[8] Following the `rifreg` command, the indicator function is here defined as 1 if log wages is below the 50th quantile. Note that this is slightly different from how the RIF is defined in (1).

3.2 Regression model

Next, we can include the transformed outcome variable in any linear regression model. For example, we can use `xtreg`, which is considerably more efficient than `regress` if the regression model includes high-dimensional fixed effects. To estimate UQR with fixed effects using `xtreg`, simply type

```
xtreg q50 union, i(idcode) fe
```

The coefficient of union is equivalent to the much more time-demanding approach of including $N-1$ person-dummy variables in `rifreg`.

```
xi: rifreg ln_wage union i.idcode, quantile(50)
```

3.3 Weights

One can also include weights in this two-step approach. However, the weights must be included in both `rifreg` (or `pctile` and `kdensity`) and `xtreg`.

4 Standard errors

The two-step approach outlined in section 3 and the `rifreg` command do not automatically produce identical standard errors. With `rifreg`, we have the option of conventional standard errors (the default in `regress` and `xtreg`),

```
xi: rifreg ln_wage union i.idcode, quantile(50) norobust
```

robust standard errors (the Huber/White/sandwich estimator, which is the default in `rifreg`),

```
xi: rifreg ln_wage union i.idcode, quantile(50)
```

and bootstrapped standard errors.

```
xi: rifreg ln_wage union i.idcode, quantile(50) bootstrap
```

4.1 Conventional standard errors

The conventional standard errors assume that the error term is independent and identically distributed. To replicate the conventional standard errors in `rifreg` using the `xtreg` command, we type the following:

```
xtreg q50 union, fe i(idcode)
```

4.2 Robust standard errors

Reporting the default standard errors in `rifreg` relaxes the assumption that the error term is identically distributed (but not the independence assumption) by using the Huber/White/sandwich estimator. Replicating these standard errors using `xtreg` is complicated, because `xtreg` with the `robust` option reports standard errors that not only relax the assumption that the error term is identically distributed but also relax the independence assumption. With the `vce(robust)` option, `xtreg` requires only that the observations are independent across the panel variable, otherwise known as cluster–robust standard errors in Stata.[9]

However, that `xtreg` reports cluster–robust standard errors is certainly not a drawback. Using cluster–robust standard errors in fixed-effects models is often a better choice than using robust standard errors (Cameron and Miller 2015).

```
xtreg q50 union, fe i(idcode) robust
```

4.3 Bootstrapped standard errors

Most studies using quantile regression report bootstrapped standard errors, which is also what Firpo, Fortin, and Lemieux (2009) report in their main examples. Unfortunately, bootstrapping the standard errors in UQR is slightly more complex than bootstrapping the standard errors in the OLS analysis, because the distribution of Y changes in each bootstrapped sample. Thus the value of the outcome variable, Y, at the 50th quantile, $q_{.50}$, and the estimated density at the 50th quantile, $f_Y(q_{.50})$, differ in each bootstrapped sample [see equation (1)].

Thus the transformed outcome variable must be recalculated in each of the bootstrapped samples. However, the `bootstrap` prefix command and the `vce(bootstrap)` option in `xtreg` bootstrap only the coefficients (Cameron and Trivedi 2009) and leave the transformed outcome variable unchanged.[10] Not all researchers acknowledge this limitation of the `bootstrap` command and the `vce(bootstrap)` option when fitting UQR models. In a highly accessible and useful introductory text to quantile regression, which also shows how to implement CQR and UQR in Stata, Porter (2015, 377) unfortunately makes exactly this error when bootstrapping the UQR standard errors.

The following occur when using the `bootstrap` option of `rifreg`:

1. A dataset of size `_N` is sampled with replacement.
2. In this dataset, the transformed outcome variable is generated.

[9] The user-written command `xtivreg2` provides robust standard errors in fixed-effects models (Schaffer 2005).

[10] With CQR, however, using the `bootstrap` prefix command yields identical standard errors as the `bsqreg` command, because the quantiles are defined (conditional on the covariates) in the regression model, and not before the regression model, as with UQR.

3. Using this transformed outcome variable, one runs a linear regression model and stores the value of the coefficient.
4. The three first steps are then repeated, with the default being 50 replications.
5. The standard deviation of the coefficients from the bootstrapped samples is the bootstrapped standard errors.

To get bootstrapped standard errors in UQR, we need to calculate the transformed outcome variable and run the regression model in each of the bootstrapped samples. We can do this, for instance, by using the following code:

```
. marksample touse
. markout `touse´ ln_wage union idcode
. forvalues i=1/50 {
  2.         preserve
  3.                 bsample if `touse´
  4.                 qui rifreg ln_wage if `touse´, quantile(50) retain(q50b)
  5.                 qui xtreg q50b union if `touse´, i(idcode) fe
  6.                 matrix b=nullmat(b)\e(b)
  7.         restore
  8.         }
. mata VCEmata=st_matrix("b")
. mata st_matrix("vce", variance(VCEmata))
. matrix colnames vce=union _cons
. matrix rownames vce=union _cons
. matrix list vce
symmetric vce[2,2]
            union       _cons
union   .00021213
_cons  -.00004102   .00002818
```

The square root of the diagonal elements in this variance–covariance matrix is the bootstrapped standard errors. With the `bsample` command, clustered observations could be accounted for by using the `cluster(`*varlist*`)` option.

When there are many independent variables, this bootstrap approach is tedious. I suggest using the `xtrifreg` command in these cases.

5 xtrifreg

5.1 Description

`xtrifreg` builds on the user-written `rifreg` command and can be used to fit UQR models with fixed effects. More specifically, `xtrifreg` uses `pctile` to identify the value of the outcome variable, Y, at the quantile τ (q_τ), uses `kdensity` to estimate the density of Y at q_τ [$f_Y(q_\tau)$], and then includes these values in (1). `xtreg` is subsequently used to fit the regression model with the RIF as the outcome variable. When bootstrapping the standard errors, `xtrifreg` uses the `bsample` command.

5.2 Syntax

`xtrifreg` *depvar indepvars* [*if*] [*in*] [*weight*], `fe i(`*varname*`)` [`quantile(`#`)` `kernop(`*string*`)` `width(`#`)` `norobust` `bootstrap` `clusterbootstrap` `reps(`#`)`]

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

By using this `xtrifreg` command, the two-step process is automated (as it is in the original `rifreg` command). Thus the outcome variable should be the original outcome variable (not the transformed outcome variable).

5.3 Options

`fe` specifies that a fixed-effects estimator (that is, `xtreg`) should be used. `fe` is required.

`i(`*varname*`)` specifies the fixed-effects variable. Only one fixed-effects variable can be included in `i(`*varname*`)`. `i()` is required.

`quantile(`#`)` specifies the quantile. The 75th quantile, for instance, can be written as either `quantile(.75)` or `quantile(75)`. The default is `quantile(50)`.

`kernop(`*string*`)` specifies the kernel function, where *string* is `gaussian`, `epanechnikov`, `epan2`, `biweight`, `cosine`, `parzen`, `rectangle`, or `triangle`. The default is `kernop(gaussian)`.

`width(`#`)` specifies the halfwidth of the kernel. The default is `width(0.0)`, which calculates the "optimal value" (see the help file for `rifreg`).

`norobust` specifies to include conventional standard errors. The default is to include cluster–robust standard errors.

`bootstrap` specifies to include bootstrapped standard errors.

`clusterbootstrap` specifies to include cluster–bootstrapped standard errors, with clustering on the fixed-effects variable specified in `i(`*varname*`)`.

`reps(`#`)` specifies the number of bootstrap replications. The default is `reps(50)`.

5.4 Examples

To illustrate the use of `xtrifreg`, I will estimate the effect of union membership on wages at the 50th quantile with fixed effects on individuals. I will show how to include conventional standard errors, cluster–robust standard errors, bootstrapped standard errors, and cluster–bootstrapped standard errors.

For comparison, I begin by fitting the model using the `rifreg` command with conventional standard errors, robust standard errors, and bootstrapped standard errors (ignore the `timer` command for now).

```
. xi: rifreg ln_wage union i.idcode, quantile(50) norobust
  (output omitted)
. estimates store norobust
. xi: rifreg ln_wage union i.idcode, quantile(50)
  (output omitted)
. estimates store robust
. timer on 1
. set seed 339487731
. xi: rifreg ln_wage union i.idcode, quantile(50) bootstrap reps(50)
  (output omitted)
. estimates store bootstrap
. timer off 1
. esttab norobust robust bootstrap, keep(union) se mtitle

------------------------------------------------------------
                      (1)             (2)             (3)   
                 norobust          robust       bootstrap   
------------------------------------------------------------
union               0.126***        0.126***        0.126***
                 (0.0104)        (0.0121)        (0.0145)   
------------------------------------------------------------
N                   19238           19238           19238   
------------------------------------------------------------
Standard errors in parentheses
* p<0.05, ** p<0.01, *** p<0.001
```

The syntax for `xtrifreg` is similar to that of `rifreg`, but instead of including $N-1$ person-dummy variables, we include the `fe` and `i(idcode)` options. The following are examples using `xtrifreg`:

```
. xtrifreg ln_wage union, quantile(50) norobust fe i(idcode)
  (output omitted)
. estimates store norobust
. xtrifreg ln_wage union, quantile(50) fe i(idcode)
  (output omitted)
. estimates store clusterrobust
. timer on 2
. set seed 339487731
. xtrifreg ln_wage union, quantile(50) bootstrap reps(50) fe i(idcode)
  (output omitted)
. estimates store bootstrap
. timer off 2
. set seed 339487731
. xtrifreg ln_wage union, quantile(50) clusterbootstrap reps(50) fe i(idcode)
  (output omitted)
. estimates store clusterbootstrap
. esttab norobust clusterrobust bootstrap clusterbootstrap, keep(union) se mtitle

----------------------------------------------------------------------------
                      (1)             (2)             (3)             (4)   
                 norobust    clusterrob~t       bootstrap    clusterboo~p   
----------------------------------------------------------------------------
union               0.126***        0.126***        0.126***        0.126***
                 (0.0104)        (0.0158)        (0.0145)        (0.0139)   
----------------------------------------------------------------------------
N                   19238           19238           19238           19238   
----------------------------------------------------------------------------
Standard errors in parentheses
* p<0.05, ** p<0.01, *** p<0.001
```

This demonstrates that `rifreg` and `xtrifreg` produce identical point estimates, as well as identical conventional and bootstrapped standard errors. The standard errors in column 2 in the two tables above are not identical, because `xtrifreg` reports cluster–robust standard errors while `rifreg` reports robust standard errors. Also, unlike with `rifreg`, we can include cluster–bootstrapped standard errors using `xtrifreg`. In this particular example, they are somewhat smaller than the cluster–robust and bootstrapped standard errors.

The main advantages of `xtrifreg` are that it can be used when the number of fixed effects exceeds the number of allowed right-hand-side variables and that it is much faster than `regress` when the number of fixed effects is large. In the above code, I used the `timer` command to investigate the number of seconds used to fit the models with bootstrapped standard errors using `rifreg` (timer 1) and `xtrifreg` (timer 2). The `rifreg` command needed more than 8 hours to fit the model. The `xtrifreg` command fit the same model in only about 30 seconds.

```
. timer list
   1:  29784.74 /        1 =   29784.7380
   2:     28.59 /        1 =      28.5950
```

Thus computational speed is massively increased by using `xtrifreg` rather than `rifreg`. Note that the number of bootstrapped replications was only 50, which is far less than the recommended lower limit of at least 200 (Cameron and Trivedi 2009, 433). With 200 bootstrap replications, we should expect the `rifreg` command to take almost one and a half days, and that is for only the effects on one quantile. To estimate the effect throughout the wage distribution, say, at deciles, we would need two weeks if using `rifreg`. However, with `xtrifreg`, we need fewer than 20 minutes.

```
. timer on 3

. forvalues i=10(10)90 {
  2.          set seed 339487731
  3.          qui xtrifreg ln_wage union, q(`i´) fe i(idcode) bootstrap reps(200)
  4.                  estimates store decile`i´
  5. }

. timer off 3

. timer list 3
   3:   1005.48 /        1 =    1005.4820
```

The difference in computational speed between rifreg and xtrifreg depends on many factors, including the Stata version being used, the computer specifications, the sample size, and the number of fixed effects. In this article, I used an analysis sample of 19,238 observations and 4,150 fixed effects. This is indeed a large dataset, but with the "data revolution" (Einav and Levin 2013) and, particularly, the call for expanding access to administrative data (Card et al. 2010), researchers will most likely routinely deal with sample sizes substantially larger than this.

6 Conclusion

In this article, I presented an easy-to-use two-step approach to include high-dimensional fixed effects in UQR. I suggested 1) using either the user-written rifreg command or the official pctile and kdensity commands to obtain the RIF and 2) using this RIF variable as an outcome variable in xtreg. I also introduced a new command, xtrifreg, that automates the process. xtrifreg is especially convenient when bootstrapping the standard errors.

The two-step approach I presented in this article, which is implemented by xtrifreg, is useful when you have a large number of fixed effects or when the number of fixed effects exceeds the number of allowed right-hand-side variables in Stata. Using a sample of 19,238 observations and 4,150 fixed effects, I demonstrated that the computational speed is massively increased by using xtrifreg rather than rifreg.

7 Acknowledgments

I thank the editor, an anonymous reviewer, and Solveig T. Borgen for useful comments and suggestions. This article is part of the project "Long-term effects of school-wide interventions and school environment using longitudinal register data", funded by The Research Council of Norway (grant #238050).

8 References

Allison, P. D. 2009. *Fixed Effects Regression Models*. Thousand Oaks, CA: Sage.

Borgen, N. T. 2015. Changes in the economic returns to attending prestigious institutions in Norway. *European Societies* 17: 219–241.

Budig, M. J., and M. J. Hodges. 2014. Statistical models and empirical evidence for differences in the motherhood penalty across the earnings distribution. *American Sociological Review* 79: 358–364.

Cameron, A. C., and D. L. Miller. 2015. A practitioner's guide to cluster–robust inference. *Journal of Human Resources* 50: 317–372.

Cameron, A. C., and P. K. Trivedi. 2009. *Microeconometrics Using Stata*. College Station, TX: Stata Press.

Card, D., R. Chetty, M. S. Feldstein, and E. Saez. 2010. Expanding access to administrative data for research in the United States. American Economic Association, Ten Years and Beyond: Economists Answer NSF's Call for Long-Term Research Agendas.

Castellano, K. E., and A. D. Ho. 2013. Contrasting OLS and quantile regression approaches to student "growth" percentiles. *Journal of Educational and Behavioral Statistics* 38: 190–215.

Einav, L., and J. D. Levin. 2013. The data revolution and economic analysis. NBER Working Paper No. 19035, The National Bureau of Economic Research. http://www.nber.org/papers/w19035.

Firpo, S., N. M. Fortin, and T. Lemieux. 2009. Unconditional quantile regressions. *Econometrica* 77: 953–973.

Frölich, M., and B. Melly. 2010. Estimation of quantile treatment effects with Stata. *Stata Journal* 10: 423–457.

Guimarães, P., and P. Portugal. 2010. A simple feasible procedure to fit models with high-dimensional fixed effects. *Stata Journal* 10: 628–649.

Hao, L., and D. Q. Naiman. 2007. *Quantile Regression*. Thousand Oaks, CA: Sage.

Killewald, A., and J. Bearak. 2014. Is the motherhood penalty larger for low-wage women? A comment on quantile regression. *American Sociological Review* 79: 350–357.

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

McCaffrey, D. F., J. R. Lockwood, K. Mihaly, and T. R. Sass. 2012. A review of Stata commands for fixed-effects estimation in normal linear models. *Stata Journal* 12: 406–432.

Porter, S. R. 2015. Quantile regression: Analyzing changes in distributions instead of means. In *Higher Education, Vol. 30: Handbook of Theory and Research*, ed. M. B. Paulsen, 335–381. Cham, Switzerland: Springer.

Schaffer, M. E. 2005. xtivreg2: Stata module to perform extended IV/2SLS, GMM and AC/HAC, LIML, and k-class regression for panel-data models. Statistical Software Components S456501, Department of Economics, Boston College. https://ideas.repec.org/c/boc/bocode/s456501.html.

About the author

Nicolai T. Borgen is a postdoctoral researcher in sociology at the University of Oslo.