

Documentation for Generalized Quantile Regression code

David Powell*
RAND

September 17, 2013

This documentation refers to Stata code for the GQR and IV-GQR estimators introduced in Powell [2013]. This code is preliminary and will be updated over time. The do-file is titled GQRshare.do. After downloading, please include the command “do GQRshare.do” in your do-file to load the program. The code currently limits you to one or two treatment variables (+ a constant). This is primarily because the code grid-searches over a range of possible values and including several treatment variables would be computationally difficult. In theory, the estimator can handle several treatment variables and I have had success using other optimization methods (see Chernozhukov and Hong [2003]), but this code is not ready to be shared. The available code does allow you to use as many control variables and instruments as you’d like.

The primary motivation for using GQR is that you are interested in estimating the relationship between the treatment variables and the distribution of the outcome variable. It is typical in applied work to simply use a conditional quantile framework and include additional covariates in the same manner as the treatment variables. Unfortunately, this likely provides different parameters than you are interested in. I am not the first to make this observation as both Firpo [2007] and Frölich and Melly [2012] introduce estimators for the case where there is one binary treatment variable. The primary advantage of quantile regression is that it allows the parameters of interest to vary based on a nonseparable disturbance term. Adding covariates into a conditional quantile model separates this disturbance term, partially undermining the original motivation for employing quantile regression. In the paper, I show that even with a randomized treatment variable, using quantile regression while controlling for covariates leads to inconsistent estimates. The GQR estimator circumvents this issue. I’ll refer you to the paper for more details.

*dpowell@rand.org

Here is the main information that you need for the code:

1. Please drop any observations with missing data. The code will not do this for you and missing data might cause problems.
2. The syntax for the command is as follows:

```
gqr y, end1(d1) end2(d2) instr(z) prone(x) min1(-5) max1(5) intvl1(.1)  
min2(-5) max2(5) intvl2(.1) tau(50) weight(w)
```

3. y is your outcome variable.
4. d1 is one treatment variable (one variable only). d2 is the other (one variable only) if you have two treatment variables.
5. z represents the instruments - you can have as many instruments as you'd like. If d1 is exogenous, then d1 is also your instrument. Same with d2. (Note: You should have at least as many instruments as treatment variables, but the code itself does not require this.)
6. x represents your control variables.
7. min1() - enter the minimum value that you'd like to search over for the coefficient on d1
8. max1() - enter the maximum value that you'd like to search over for the coefficient on d1
9. intvl1() - the code searches between min1() and max1() at an interval of intvl1()
10. min2() - enter the minimum value that you'd like to search over for the coefficient on d2
11. max2() - enter the maximum value that you'd like to search over for the coefficient on d2
12. intvl2() - the code searches between min2() and max2() at an interval of intvl2()
13. tau() - the quantile of interest. Please include an integer between 1 and 100.
14. w is your variable with your weights. You do not need to include weights.

If you only have one treatment variable, then just ignore the end2(), min2(), max2(), intvl2() options.

The code will provide the coefficients on d1 and d2 for the quantile of interest. It will save this output in "e(bt1)" and (if you have a second treatment variable) "e(bt2)", respectively. The constant will be saved in "e(ct)". It is advisable to bootstrap to obtain standard errors. Because of the bootstrap, it is recommended that you search over a wide range of values so that you do not obtain arbitrarily small standard errors. To bootstrap, simply use:

```

local res ‘ “bt1=r(bt1) bt2=r(bt2) ct=r(ct)” ’
bootstrap ‘res’: gqr y, end1(d1) end2(d2) instr(z) prone(x) min1(-5)
max1(5) intvl1(.1) min2(-5) max2(5) intvl2(.1) tau(50) weight(w)

```

I should note again that this code is preliminary and it is possible that the code will fail. For example, each iteration of the grid search requires a probit regression. If a probit regression does not converge properly, then Stata will exit from the do-file completely. Including lots of control variables - such as fixed effects for small T - will raise the chances of this happening.

Please email me (dpowell@rand.org) with any concerns or questions.

References

- Victor Chernozhukov and Han Hong. An MCMC approach to classical estimation. *Journal of Econometrics*, 115(2):293–346, August 2003.
- Sergio Firpo. Efficient semiparametric estimation of quantile treatment effects. *Econometrica*, 75(1):259–276, 01 2007.
- Markus Frölich and Blaise Melly. Unconditional quantile treatment effects under endogeneity. Technical report, Brown University, January 2012.
- David Powell. A new framework for estimation of quantile treatment effects: Nonseparable disturbance in the presence of covariates. Working papers, RAND Corporation Publications Department, 2013.