Documentation of STATA code for "Extremal Quantile Regression for Selection Models and the Black-White Wage Gap"

Septempber 2015

Title

myfun_combined

Syntax

myfun_combined varlist [if] [in] , indepnum(real) dnum(real) dfnum(real) cnum(real) gridd(string) yl(real) yr(real) [quant(real 0.5)]

conditions	Description
indepnum(#)	number of independent variables
dnum(#)	total number of discrete independent variables
dfnum(#)	number of free discrete independent variables, free means it is not determined by other components of X
cfnum(#)	total number of free continuous independent variables
gridd(#)	an m*n matrix, grid of independent variables
yl(#)	lower bound of the dependent variable
yr(#)	upper bound of the dependent variable
quant(#)	optional, the quantile of interest of the CLR bound (default=0.5)

Description

myfun_combined is an estimation method based on the paper *Extremal Quantile Regression for Selection Models and the Black-White Wage Gap*. In particular, the code conducts an automatic pretest for homoskedasiticity and yields the point estimates and its standard deviation of β and δ corresponding to the homoskedastic variable(s) along with the CLR bound of the τ-th marginal quantile treatment effect for the heteroskedasitic variable(s) where the quantile index τ is user-specified. In a special note, the grid generated in this function should satisfy the data restriction. For example, in our case of using NLSY79 data, black and hispanic cannot be both 1 while AFQT^2 is fully determined on AFQT. For the detail, please check the program. The main function it calls is the MATA function myfun_combined(). In myfun combined(), the default spacing parameters we use are m = 1.2, l = (0.65,0.85,1.15,1.45). The number of subsamples is set to be 500. The subsample size is (150,300,500,600) for sample size (250, 500, 1000, 2000) and the corresponding linear interpolation for sample size in between. For sample size N larger than 2000, the subsample size is set to be 600 + 0.2(N - 2000). Also, we use the rule of thumb tuning parameter and Gaussian kernel to estimate the nonparametric pieces when computing the CLR bound.

Output

Name	Description
beta_mse	the estimator of beta without imposing partial homoskedasticity
delta_mse	the estimator of delta without imposing partial homoskedasticity
V_mse_star	the asympotic variance covariance matrix for delta_mse(and also beta_mse).
Nb_mse_star	the convergence rate for beta_mse
qstard	the optimal quantile index used to estimate both delta_mse and delta_mse
beta_hom	the point estimates of beta under partial homoskedasticity for those variables who do not pass the interest
delta_hom	the point estimates of delta under partial homoskedasticity for those variables who do not pass the interest
qstarbd	the optimal quantile index used to estimate delta_hom
qstarbb	the optimal quantile index used to estimate beta_hom
theta0	the CLR bound for quant-th marginal quantile treatment effect for those variables who do not pass the pretest

User Guide

Step1: Put myfun_combined.ado into C:/ado/PERSONAL or put myfun_combined.do in current directory.

Step2: Start a new script calling myfun_combined.ado.
2a. Load the data set.
2b. Generate the grid for independent variables in MATA and return it as a matrix to STATA.
2d. Load the data set again and call the command.

Example

test_myfun_combined.do

Notes

grid

There is no general rules for generating the grid of independent variables, since it varies with data sets. The best way is to remember what the grid matrix looks like for your data set and find a way to generate it in STATA.
Grid is an m*n matrix, where m = space * # of val1 * # of val2 *… and n is the number of free independent variables. # of val1 means how many different values the first independent variable can take. For binary variables, it should be two since binary variables can only take 0 and 1. For continuous variables, we use space to measure the number of values. In our example, we take space as 200. Users can specify a different value for space based on their data sets.
Basically, the grid matrix is a matrix containing all combination scenarios given different values of

each independent variables.

Examples:

1. With continuous independent variables

   Usually you should start with generating grid columns for continuous independent variables, since they usually have a large number of different values. The best way I can find is to use egen var = fill(1(1)space 1(1)space). And then do some transformation to this variable.

   For categorical variables and binary variables, use the similar command egen var = fill(a b c a b c).

   Before you generate next columns, always remember to sort the previous column you have generated.

   Ex

```
set obs 2400
egen tempc4 = fill(1(1)`space' 1(1)`space')
gen c4 =    `arange' * tempc4/(`space' + 1) + `p5'
drop tempc4
sort c4
egen c2 = fill(0 1 0 1)
sort c4 c2
egen c3 = fill(26 27 28 26 27 28)
sort c4 c3 c2
egen c1 = fill(0 1 0 1)
```

2. With more than one related binary variables

   We sometimes use more than one binary variables to indicate different categories each observation belongs. In our case, we have black and Hispanic indicating the race of each observations. Then for the grid matrix, remember to drop those scenarios where black and Hispanic are both 1.

CLR_bound

Users should specify the range of values the dependent variables can take.