Distribution regression made easy

Philippe Van Kerm

Luxembourg Institute of Socio-Economic Research
philippe.vankerm@liser.lu

2016 Swiss Stata Users Group meeting

November 17 2016, University of Bern

The method

A worked example
(eight implementation tips)

Outline

- "Distribution regression methods": Relate some distributional statistics $v(F)$ to multiple 'explanatory' variableS X
  - F is a (univariate) income distribution function
  - $v(F)$ is a generic functional: quantile, inequality measure (quantile share ratios, Gini coefficient, etc.), poverty index
- Two related questions:
  - How does F and/or $v(F)$ vary with X?
    That is, calculate and compare $v(F_x)$ (remember $dim(X) > 1$), 'partial effects')
    - EOp, Educ choices, policy intervention, etc.
  - How much do differences in X account for differences in $v(F)$ over time, country, gender, etc.?

Two main approaches

Two main approaches in recent literature

1. Recentered influence function regression (Firpo et al., 2009, Van Kerm, 2015):

2. Distribution function modelling (e.g., Chernozhukov et al., 2013):
   - model $F(y) = \int F_x(y)h(x)dx$:
     essentially involves modelling the conditional distribution $F_x(y)$
   - plug model predictions for F (or F_x) in $\upsilon(F)$
   - examine counterfactuals ('manipulate' conditional distribution or covariate distribution)

Array of models for conditional distributions F_x

Many models and estimators available, more or less parametrically restricted, e.g.,:

- quantile regression (Koenker and Bassett, 1978)
- parametric income distribution models, 'conditional likelihood' models (Biewen and Jenkins, 2005, Van Kerm et al., 2016)
- duration models (Donald et al., 2000, Royston, 2001, Royston and Lambert, 2011)
- 'distribution regression' (Foresi and Peracchi, 1995)

‘Distribution regression’ is really simple

(Foresi and Peracchi, 1995)

$F_x(y) = \Pr\{y_i \leq y|x\}$ is a binary choice model once y is fixed (dependent variable is $1(y_i < y)$)

Estimate $F_x(y)$ on a (fine) grid of values for y spanning the domain of definition of Y by running repeated standard binary choice models, e.g. a logit model:

$$\begin{aligned} F_x(y) &= \Pr\{y_i \leq y|x\} \\ &= \Lambda(x\beta_y) \\ &= \frac{\exp(x\beta_y)}{1+\exp(x\beta_y)} \end{aligned}$$

And then since $F(y) = E_x(F_x(y))$

$$\hat{F}(y) = \frac{1}{N}\sum_{i=1}^{N} \hat{F}_{x_i}(y) = \frac{1}{N}\sum_{i=1}^{N} \Lambda(x_i \hat{\beta}_y)$$

Why ‘Distribution regression’?

- Flexible: Repeating estimation at different values of y makes little assumptions about the overall shape of conditional distributions
- Evidence that provides better fit to income data than quantile regression (Rothe and Wied, 2013, Van Kerm et al., 2016) although theoretically equivalent (Koenker et al., 2013)
- Faster to run than quantile regression in my experience (though slower than more parameterised models)
- Estimation is straightforward!

Simulation

From F_x to $\upsilon(F_x)$

- Uniform (equally-spaced) sequence of conditional quantile predictions for each observations gives a pseudo-random sample from $\hat{F}_{x_i}$, e.g., $\hat{F}_x^{-1}(.01)$, $\hat{F}_x^{-1}(.02)$, ..., $\hat{F}_x^{-1}(.99)$

  X: $\upsilon(F_x)$ calculated as with direct unit-record data

- predictions after logits give series of $\hat{F}$s (not of $\hat{F}^{-1}$s), so inversion (e.g., by interpolation) required (but easy)

From F_x to $\upsilon(F)$

- Stacking predictions for all observations into one long vector V: pseudo-random sample from the unconditional distribution F
  - GOTO X

Counterfactual distributions

"Generalized Oaxaca-Blinder" decomposition

1. Estimate and predict conditional distribution functions for, say, men $\hat{F}_x^m$ and women $\hat{F}_x^w$
2. Simulate counterfactual distributions $\tilde{F}$ by averaging predictions of one group over covariate distribution of other group, e.g.,

$$\tilde{F}(y) = \frac{1}{N^w} \sum_{i=1}^{N^w} \hat{F}_{x_i}^m$$

3. Decompose differences in the two unconditional CDFs as differences attributed to F_x ('structural' part) and to differences in covariates ('compositional' part):

$$(\hat{F}^w(y) - \hat{F}^m(y)) = (\hat{F}^w(y) - \tilde{F}(y)) + (\tilde{F}(y) - \hat{F}^m(y))$$

(See Chernozhukov et al. (2013) for inferential theory.)

The method

A worked example
(eight implementation tips)

A simple worked example: household incomes in Spain

- Survey data on household disposable income in Spain in 2006 and 2012 (from European Union Statistics on Income and Living Conditions)
- Covariates: gender and age of household head, share of adults at work, number of adults and of children of different ages

  Are female-headed households disadvantaged? How did distribution change before/after Great Recession?

```
svyset [pw=rw] , strata(uniqid) psu(hid)
loc vlist   i.femmain (c.agemain##c.agemain) ///
            c.shatwork ///
            c.nadu2 c.nkid06 c.nkid712 c.nkid1318 c.nkid19plus
```

Tip #1: setting the grid

Tip #1: use quantiles as evaluation grid

```
* 1: evaluation points
loc plist 0.5 2(2)98 99.5
loc imed 25
_pctile inc [aw=rw]  , percentiles(`plist')
loc j 0
foreach p of numlist `plist' {
    loc ++j
    loc v`j' = r(r`j')
    loc p`j'   `p'
    qui gen byte z`j' = (inc<=`v`j'')  if !mi(inc)
    * not frugal on memory but wait...
}
loc P `j'
```

Tip #2: start around the median

Tip #2: start around the median (where F_x is about .50)

```
. * 2: logits
. * Tip: start in middle and move from(...) there!
. svy: logit z`imed'  `vlist'
(running logit on estimation sample)

Survey: Logistic regression

Number of strata   =          1                  Number of obs     =     32,704
Number of PSUs     =     12,365                  Population size   = 45,065,241
                                                 Design df         =     12,364
                                                 F(   9,  12356)   =     115.35
                                                 Prob > F          =     0.0000

------------------------------------------------------------------------------------
                   |             Linearized
               z25 |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------------+----------------------------------------------------------------
         1.femmain |   .0306439   .0596061     0.51   0.607    -.0861933    .1474811
           agemain |  -.0598019   .0141086    -4.24   0.000    -.0874569   -.0321468
                   |
c.agemain#c.agemain |   .0003402   .0001244     2.73   0.006     .0000963     .000584
                   |
          shatwork |    -2.8492   .0948404   -30.04   0.000    -3.035102   -2.663298
             nadu2 |  -.0302988   .0364321    -0.83   0.406    -.1017115    .0411139
            nkid06 |   .3440553   .0613516     5.61   0.000     .2237966    .4643141
           nkid712 |   .5616167   .0621392     9.04   0.000     .4398142    .6834192
          nkid1318 |   .8678787   .0669607    12.96   0.000     .7366252    .9991322
        nkid19plus |   .5545465   .0927833     5.98   0.000     .3726769    .7364162
             _cons |   3.025542   .3916611     7.72   0.000     2.257825    3.793259
------------------------------------------------------------------------------------

.         estimates store z`imed'

.         mat def b`imed' = e(b)

.         predict double F`imed' , pr rules // !rules
```

Tip #3: `predict , rules`

Tips #3: `predict , rules` to predict 0's and 1's when 'completely determined outcomes'

```
* 2: logits
* Tip: start in middle and move from(...) there!
svy: logit z`imed'  `vlist'
    estimates store z`imed'
    mat def b`imed' = e(b)
    predict double F`imed' , pr rules // !rules
```

Tip #4: `from`

Tip #4: Move upwards (and downwards) `from` the middle (to speed up convergence).
(Consider one-step Newton-Raphson only (Cai et al., 2000)?)

```
* upwards...
mat def previousb = b`imed'
forv i=`=`imed'+1'(1)`P' {
    matrix coleq previousb = z`i'
    qui svy : logit z`i' `vlist' , from(previousb) // iterate(1)  als
        estimates store z`i'
        mat def previousb = e(b)
        qui predict double F`i' , pr rules
}
```

Tip #5: combine equations

Tip #5: use `suest` to combine separate estimates into multiple-equations 'object' (`e(b)` and `e(V)`) so you can `test` cross-equation hypotheses

```
. suest z* , svy

Simultaneous survey results for z25, z26, z27, z28, z29, z30, z31, z32, z33, z34, z35, z36, z37, z38, z39, z40,
> z42, z43, z44, z45, z46, z47, z48, z49, z50, z51, z24, z23, z22, z21, z20, z19, z18, z17, z16, z15, z14, z13,
> z11, z10, z9, z8, z7, z6, z5, z4, z3, z2, z1

Number of strata   =          1                 Number of obs     =     32,704
Number of PSUs     =     12,365                 Population size   =  45,065,241
                                                Design df         =     12,364

------------------------------------------------------------------------------------
                    |             Linearized
                    |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
--------------------+---------------------------------------------------------------
z25_z25             |
          1.femmain |   .0306439    .0596061     0.51   0.607    -.0861933    .1474811
            agemain |  -.0598019    .0141086    -4.24   0.000    -.0874569   -.0321468
                    |
c.agemain#c.agemain |   .0003402    .0001244     2.73   0.006     .0000963     .000584
                    |
           shatwork |    -2.8492    .0948404   -30.04   0.000    -3.035102   -2.663298
              nadu2 |  -.0302988    .0364321    -0.83   0.406    -.1017115    .0411139
             nkid06 |   .3440553    .0613516     5.61   0.000     .2237966    .4643141
            nkid712 |   .5616167    .0621392     9.04   0.000     .4398142    .6834192
           nkid1318 |   .8678787    .0669607    12.96   0.000     .7366252    .9991322
          nkid19plus |   .5545465    .0927833     5.98   0.000     .3726769    .7364162
              _cons |   3.025542    .3916611     7.72   0.000     2.257825    3.793259
--------------------+---------------------------------------------------------------
z26_z26             |
          1.femmain |   .0713429    .0595535     1.20   0.231    -.0453912    .1880769
            agemain |  -.0646379    .0141458    -4.57   0.000    -.0923658   -.0369099
                    |
c.agemain#c.agemain |    .000385    .0001252     3.07   0.002     .0001395    .0006305
                    |
```

Tip #5: combine equations

Tip #5: use `suest` to combine separate estimates into multiple-equations 'object' (`e(b)` and `e(V)`) so you can `test` cross-equation hypotheses

```
suest z* , svy
estimates store zfull

test [z6_z6]
test [z1_z1]1.femmain = [z`P'_z`P']1.femmain

test [z1_z1]1.femmain  , notest
forv i=2/`=`P'-1' {
    test [z`i'_z`i']1.femmain , accum notest
}
test [z`P'_z`P']1.femmain , accum

test [z1_z1 = z2_z2 = z6_z6 = z10_z10 = z11_z11] ,
test [z1_z1 = z2_z2 = z6_z6 = z10_z10 = z11_z11] , cons
```

Test examples

e.g., income distribution for female-headed households any different?

```
. test [z`P'_z`P']1.femmain , accum

Adjusted Wald test

 ( 1)  [z1_z1]1.femmain = 0
 ( 2)  [z2_z2]1.femmain = 0
 ( 3)  [z3_z3]1.femmain = 0
 ( 4)  [z4_z4]1.femmain = 0
 ( 5)  [z5_z5]1.femmain = 0
 (48)  [z48_z48]1.femmain = 0
 (49)  [z49_z49]1.femmain = 0
 (50)  [z50_z50]1.femmain = 0
 (51)  [z51_z51]1.femmain = 0

       F( 51, 12314) =    1.25
            Prob > F =    0.1094
```

Tip #6: Inversion and simulation

Example of simple inversion by linear interpolation

First, initialize $F(0)$ and $F(1)$

```
* 4: Calculating distribution stats by 'simulation'
* 4.1: draw realisations from predictions
*      naive, simple approach: draw from uniform and interpolate
    loc v0 = 75
    gen F0 = 0
    loc v`=`P'+1' = 11000
    gen F`=`P'+1' = 1
```

Tip #6: Inversion and simulation

Example of simple inversion by linear interpolation

Then invert

```
* inversion for uniform draws
gen below = .
gen above = .
gen double Fabove = .
gen double Fbelow = .
gen switchabove = .
gen aboveweight = .

loc k 0
gen u = .
quietly {
forv p=1(1)99 {
    replace u = `p'/100
    replace above = .
    forvalues i=0/`=`P'+1' {
        replace switchabove = cond(u <= F`i' & mi(above) & !mi(F`i') , 1, 0)
        replace above = `v`i'' if switchabove==1
        replace below = `v`i'' if     u > F`i'  & !mi(F`i')
        replace Fabove = F`i' if switchabove==1
        replace Fbelow = F`i' if     u > F`i'  & !mi(F`i')
    }
    replace aboveweight = (u - Fbelow)/(Fabove-Fbelow)
    gen draw`++k' = above * aboveweight   +  below * (1 - aboveweight)
}
}
```

Tip #6: Inversion and simulation

Example of simple inversion by linear interpolation

Then stack predicted quantiles and evaluate summary statistics of interest

```
* 4.2 evaluate conditional distributive statistics
    keep pid inc rw draw1-draw99
    reshape long draw , i(pid) j(p)
    egen ctheil = theil(draw) , by(pid)
    su ctheil if p==1
    ineqdeco draw [aw=rw]
```

Tip #7: run one model with full interactions

(if you are tempted to run two parallel models!)

... so testing is easy

```
* 2: Fully interacted model
loc vlist    i.femmain c.agemain c.agemain#c.agemain c.shatwork ///
             c.nadu2 c.nkid06 c.nkid712 c.nkid1318 c.nkid19plus
loc vlist2 `vlist' i.year
    foreach var in `vlist' {
        loc vlist2 `vlist2'  `var'#i.year
    }
gen at2006 = 2006
gen at2012 = 2012
```

Tip #7: run one model with full interactions

(if you are tempted to run two parallel models!)

… so testing is easy

```
* 3: combine estimates to preform tests
suest z* , svy
estimates store zfull
test 2012.year 1.femmain#2012.year ///
      c.agemain#2012.year c.agemain#c.agemain#2012.year ///
      c.shatwork#2012.year c.nadu2#2012.year ///
      c.nkid06#2012.year c.nkid712#2012.year c.nkid1318#2012.year ///
      c.nkid19plus#2012.year
```

Tip #8: margins give you $\hat{F}$ from $\hat{F}_x$

… along with confidence intervals!

```
* Tip: start in middle and move from(...) there!
svy: logit z`imed'  `vlist2'
    estimates store z`imed'
    mat def b`imed' = e(b)
    margins , over(year) at(year=(2006 2012))
    margins , over(year) dydx(year)
    rename (at2006 year) (year temp)
    predict double F`imed'_2006 , pr rules // !rules
    rename (at2012 year) (year at2006)
    predict double F`imed'_2012 , pr rules // !rules
    rename (temp year) (year at2012)
    su F`imed'_2006 F`imed'_2012  [aw=rw] if year==2006
    su F`imed'_2006 F`imed'_2012  [aw=rw] if year==2012
```

Tip #8: `margins` give you $\hat{F}$ from $\hat{F}_x$

```
.         margins , over(year) at(year=(2006 2012))

Predictive margins                              Number of obs     =     66,802
Model VCE    : Linearized

Expression   : Pr(z25), predict()
over         : year

1._at        : 2006.year
                   year            =        2006
               2012.year
                   year            =        2006

2._at        : 2006.year
                   year            =        2012
               2012.year
                   year            =        2012

------------------------------------------------------------------------------
             |            Delta-method
             |     Margin   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
     _at#year |
      1 2006  |   .4527713   .0054656    82.84   0.000     .4420591    .4634836
      1 2012  |   .5010425   .0054765    91.49   0.000     .4903087    .5117762
      2 2006  |   .4621012     .00611    75.63   0.000     .4501259    .4740765
      2 2012  |   .5061872   .0055485    91.23   0.000     .4953124     .517062
------------------------------------------------------------------------------
```

Tip #8: `margins` give you $\hat{F}$ from $\hat{F}_x$

(check for yourself)

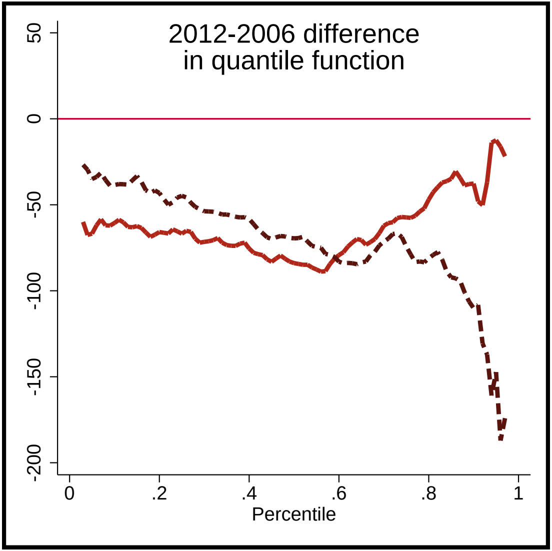
```
.        su F`imed'_2006 F`imed'_2012  [aw=rw] if year==2006

    Variable |     Obs      Weight        Mean   Std. Dev.       Min        Max
-------------+------------------------------------------------------------------
    F25_2006 |  34,098  42883962.3    .4527713    .2299601    .0226829   .9941755
    F25_2012 |  34,098  42883962.3    .4621012    .2048463     .020995   .9896899

.        su F`imed'_2006 F`imed'_2012  [aw=rw] if year==2012

    Variable |     Obs      Weight        Mean   Std. Dev.       Min        Max
-------------+------------------------------------------------------------------
    F25_2006 |  32,704  45065241.1    .5010425    .2396453    .0730445   .9935855
    F25_2012 |  32,704  45065241.1    .5061872    .2181891    .0757136   .9902083
```

2006-2016: Actual and simulated quantiles functions



Conclusion

- DR is
  - easy and intuitive
  - flexible and accurate
  - (some speed vs. accuracy trade off's not discussed here)
- Stata's `suest`, `margins`, `test` are there to make life easier (though one may still want to bootstrap the process)

Biewen, M. and Jenkins, S. P. (2005), 'Accounting for differences in poverty between the USA, Britain and Germany', *Empirical Economics* **30**(2), 331–358.

Cai, Z., Fan, J. and Li, R. (2000), 'Efficient estimation and inferences for varying coefficient models', *Journal of the American Statistical Association* **95**, 888–902.

Chernozhukov, V., Fernández-Val, I. and Melly, B. (2013), 'Inference on counterfactual distributions', *Econometrica* **81**(6), 2205–2268.

Donald, S. G., Green, D. A. and Paarsch, H. J. (2000), 'Differences in wage distributions between Canada and the United States: An application of a flexible estimator of distribution functions in the presence of covariates', *Review of Economic Studies* **67**(4), 609–633.

Firpo, S., Fortin, N. M. and Lemieux, T. (2009), 'Unconditional quantile regressions', *Econometrica* **77**(3), 953–973.

Foresi, S. and Peracchi, F. (1995), ‘The conditional distribution of excess returns: An empirical analysis’, *Journal of the American Statistical Association* **90**(430), 451–466.

Koenker, R. and Bassett, G. (1978), ‘Regression quantiles’, *Econometrica* **46**(1), 33–50.

Koenker, R., Leorato, S. and Peracchi, F. (2013), Distributional vs. quantile regression, Research Paper 11-15-300, CEIS Tor Vergata, University of Rome Tor Vergata.

Rothe, C. and Wied, D. (2013), ‘Misspecification testing in a class of conditional distributional models’, *Journal of the American Statistical Association* **108**(501), 314–324.

Royston, P. (2001), ‘Flexible alternatives to the Cox model, and more’, *Stata Journal* (1), 1–28.

Royston, P. and Lambert, P. C. (2011), *Flexible parametric survival analysis using Stata: Beyond the Cox model*, StataPress, College Station, TX.

Van Kerm, P. (2015), Influence functions at work, United Kingdom Stata Users' Group Meetings 2015 11, Stata Users Group.
URL: *https://ideas.repec.org/p/boc/usug15/11.html*

Van Kerm, P., Choe, C. and Yu, S. (2016), 'Decomposing quantile wage gaps: a conditional likelihood approach', *Journal of the Royal Statistical Society (Series C)* **65**(4), 507–27.
`http://onlinelibrary.wiley.com/doi/10.1111/rssc.12137/pdf`.

This work is part of the project 'Tax-benefit systems, employment structures and cross-country differences in income inequality in Europe: a micro-simulation approach–SIMDECO' supported by the Luxembourg National Research Fund (contract C13/SC/5937475).