

Stata code used in "The role of tobacco taxes in starting and quitting smoking: duration analysis of British data"

Martin Forster & Andrew M. Jones

Contact details:

Professor Andrew Jones
Department of Economics and Related Studies
University of York
York
YO10 5DD
United Kingdom
Fax: +44-1904-433759
E-mail: amj1@york.ac.uk
Web: <http://www.york.ac.uk/res/herc>

Most of the estimates reported in our paper are computed using standard commands from Stata v.6.0. However estimation of the split population model of starting and the gamma model of quitting with heaping effects requires custom programs:-

1. SPLIT POPULATION DURATION MODEL WITH TIME VARYING COVARIATES (TVCs)

Estimation of the split population model is done with Stata programs, written for the maximum likelihood "d0" and "lf" routines. To deal with TVCs the dataset is "expanded" by the age of starting. The data are stset using id(.) and, in the case of method d0, the subroutine mlsum is used to allow for the repeated observations on each respondent.

```
/* STATA PROGRAM USED IN M. FORSTER AND A.M.JONES , "THE ROLE OF TOBACCO TAXES IN STARTING AND QUITTING SMOKING */
```

```
/* 1. Split-population model with TVCs (written for method d0, method lf code is similar): */
```

```
program define splitd0;
```

```
/*program for maximum likelihood estimation of split population duration model with probit splitting mechanism and log-logistic durations and allowing for time varying covariates.
```

```
NOTES:
```

```
i. Duration data should have been stset before calling this program e.g.,
```

```
stset T, failure(start) id(serno)
```

```
ii. The program is called using a command of the form e.g.,
```

```
ml model d0 splitd0 (duration: T = list of covariates) (probit: start=list of covariates) /shape
```

```
*/
```

```

version 6.0;
args todo b lnf;
tempvar theta1 theta2 theta3; /*theta1-covariates for log-
logistic equation theta2- covariates for probit equation
theta3- shape parameter for log-logistic */

mlevel `theta1' = `b', eq(1);
mlevel `theta2' = `b', eq(2);
mlevel `theta3' = `b', eq(3);

local id : char _dta[st_id];
local t = "$ML_y1";
local c = "$ML_y2";

tempvar h p s d l r pr prj;

quietly {gen double `l'=exp(-`theta1');
gen double `d'=normprobit(`theta2');
gen double `r'=ln((1+(`l'*(_t0))^(1/`theta3'))
/(1+(`l'*`t')^(1/`theta3')));

/* NOTE: "serno" is the individual identifier in our
application, in general st_id could be used. */
sort serno;
by serno: gen double `pr'= sum(`r');
gen double `prj'= exp(`pr');
gen double `s'=1 - `d' +`d'*`prj';
gen double `h'= - ln(`theta3')+((1/`theta3')-
1)*ln(`t')+(1/`theta3')*ln(`l')- ln(1+(`l'*`t')^(1/`theta3')));

mlsum `lnf'=`c'*( ln(`d')+ `h' + ln(`prj'))
+(1 -`c')*ln(`s') if T==agestrt ;
/* NOTE: the statement "T==agestrt" is specific to our
application and selects the final period for each individual
observation */
};

end;

```

2. GAMMA DURATION MODEL WITH A HEAPING EFFECT

Torelli and Trivellato (1993) propose a solution to the "heaping effect" based on an explicit measurement model. This model is superimposed on the underlying duration model leading to a reformulation of the log-likelihood function.

Torelli and Trivellato compare four methods of dealing with heaping:-

- i. Re-formulating the likelihood to allow for the measurement model. This requires specifying a parametric model of the measurement errors.
- ii. The ad hoc approach of adding dummy variables for the heaped observations.
- iii. Smoothing the data prior to estimation by using random draws from a uniform distribution to spread the actual heaped observations. This means that the results are contingent on the random numbers that are generated.
- iv. Ignoring the heaping and estimating the underlying duration model.

Method i: We have programmed ml estimation of the gamma model, using the "lf" routine in Stata. We assume that the heaped observations are those where EXFAGAN is a multiple of 5 or 10. Because heaping is due to EXFAGAN the problem only relates to complete spells i.e., those who have quit smoking. For the observations, the usual contribution to the likelihood, $f(t_i)$, is replaced by,

$$F({}_u t_i) - F({}_l t_i)$$

where ${}_l t_i$ is the lower limit and ${}_u t_i$ the upper limit of an interval of length 5 around t_i .

Method iii: The problem of heaping relates to EXFAGAN, rather than the other components of the dependent variable, so we apply the smoothing method to this variable. For each of the potentially heaped values (5,10,...) the actual observation is smoothed using pseudo random integers (the stata command generates EXSMOOTH = EXFAGAN - 3 + int(5*uniform())) with the seed set at 123456789). No adjustment is required for the censored observations whose durations do not depend on EXFAGAN.

Torelli, N. and Trivellato, U. (1993) Modelling inaccuracies in job-search duration data. Journal of Econometrics, 59:187-211.

```

/* 2.Gamma duration model with heaping effect (adapted from
gamma_lf.ado).

NOTES: :
i. Duration data should have been stset before calling this
program e.g.,

stset T, failure(exfag) id(serno)

ii. The program is called using a command of the form e.g.,

gammalf T $qvar if male==1, cluster(serno) dead(exfag) robust t0(_t0)
*/

/* The first program "gammalf" is a minor modification of the Stata
ado file for the gamma model. It calls the program gammalf4 which
contains the modified likelihood function for the model with heaping.
This program is not included in this document*/

/* The following program, "gammalf4", returns the log-likelihood
function for the gamma model with heaping. */

program define gammalf4
version 6.0
    local lnf "`1'"
    local I "`2'"
    local s "`3'" /* ln_sigma */
    local k "`4'" /* kappa */
    quietly {
        local id : char _dta[st_id]
        local t = "$ML_y1"

```

```

local t0 = "$EREGt0"
local d = "$EREGd"
tempvar z z0 let let0 et et0 l sig zu zl
gen double `l'=(`k')^(-2)
gen double `sig'=exp(`s') /* sigma */
gen double `z'=(ln(`t')-`I')/`sig'
gen double `z0'=(ln(`t0')-`I')/`sig' if `t0'>0
gen double `zu'=`z'
gen double `zl'=`z'
replace `zu'=(ln(`t'+2.5)-`I')/`sig' if exheap==1
replace `zl'=(ln(`t'-2.5)-`I')/`sig' if exheap==1 & `t'>3

if `k'<0 {
    replace `z' = -`z'
    replace `z0' = -`z0'
}
gen double `et'=gammmap(`l', `l'*exp(`z'/sqrt(`l')))
gen double `et0'=gammmap(`l', `l'*exp(`z0'/sqrt(`l')))
if `k'>0 {
    replace `et'=1-`et'
    replace `et0'=1-`et0'
}
gen double `let'= ln(`et')
gen double `let0'= ln(`et0') if `t0'>0
if abs(`k')<0.01 {
    replace `lnf' = `d' * /*
        */ (-0.5*ln(2*_pi)-0.5*(`z'^2)- ln(`sig')-
`let')+`let'
}
else {
replace `lnf' = ((`l'-0.5)*ln(`l')) /*
    */ + (`z'*sqrt(`l'))-`l'*exp(`z'/sqrt(`l')) /*
    */ -lngamma(`l') - ln(`sig') if `d'==1 & exheap==0
replace `lnf' = gammmap(`l',`l'*exp(`zu'/sqrt(`l'))) - /*
    */ gammmap(`l',`l'*exp(`zl'/sqrt(`l'))) if `d'==1 &
exheap==1

replace `lnf' = `let' if `d'==0
}
replace `lnf' = `lnf'-`let0' if `t0'>0
}
end

```