Discrete-time Event History Analysis

PRACTICAL EXERCISES

Fiona Steele and Elizabeth Washbrook

Centre for Multilevel Modelling

University of Bristol

16-17 July 2013

Discrete-time Event History Analysis

Practical 1: Discrete-Time Models of the Time to a Single Event

Note that the following Stata syntax is contained in the annotated do-file **prac1.do**

You can either type in each command into the command box below at the bottom of the analysis window, or read **prac1.do** into the Do-file Editor and select the relevant syntax for each stage of the analysis.

To open the Do-file Editor, go to the **File** menu and select **Open**. Change the file type to **Do Files (*.,do, *.ado)** and locate **prac1.do**. Highlight the syntax you want to run, then hover over the icons on the tool bar until you find **Execute Selection (do)**. Alternatively, in the analysis window the 7th button from left opens a 'do file editor', from which we can write and run syntax commands.

In the do-file editor the last button on the toolbar executes the commands from the entire syntax file (or just a selection if some portion of the file is highlighted).

1.1 Introduction to the NCDS Dataset

In this exercise, we will analyse a subsample of data from the National Child Development Study (NCDS). This is a cohort study, following all individuals born in Britain in a particular week of March 1958. Partnership histories were collected when the respondents were aged 33. Here, we analyse the time from age 16 to the formation of an individual's first partnership (either a marriage or cohabitation). The Stata data file is called **ncds.dta**.

The file contains the following variables:

Variable	Description	Coding
id	Person identifier	
age1st	Age at first partnership	Equals 33 for censored cases
event	Indicator of event occurrence	1=partnered, 2=single, i.e. censored
ageleft	Age at which respondent left full-time education	
female	Respondent's gender	1=female, 0=male
region	Region of residence at 16	1=Scotland and the North

		2=Wales and the Midlands 3=Southern and Eastern 4=South East, including London
fclass	Father's social class (defined by occupation)	1=class I or II (professional and managerial) 2=class III 3=class IV or V (manual)

Open the data file and use the list command to view the first 20 cases

```
. use ncds, clear
. list in 1/20
```

```
     +-------------------------------------------------------------------+
     | id   age1st      event   ageleft   female     region    fclass |
     |-------------------------------------------------------------------|
  1. |  1       21   Married/        18        0   Wales an       iii |
  2. |  2       31   Married/        21        0   South Ea       iii |
  3. |  3       23   Married/        18        0   Wales an       iii |
  4. |  4       20   Married/        16        1   Scotland   IV or V |
  5. |  5       20   Married/        16        1   Southern   I or II |
     |-------------------------------------------------------------------|
  6. |  6       22   Married/        16        1   Wales an   I or II |
  7. |  7       20   Married/        21        0   Scotland       iii |
  8. |  8       26   Married/        16        0   South Ea   I or II |
  9. |  9       21   Married/        21        1   Southern       iii |
 10. | 10       25   Married/        18        0   South Ea         . |
     |-------------------------------------------------------------------|
 11. | 11       25   Married/        18        0   Wales an   I or II |
 12. | 12       30   Married/        21        0   Scotland       iii |
 13. | 13       25   Married/        18        0   Southern   IV or V |
 14. | 14       18   Married/        16        1   Southern       iii |
 15. | 15       18   Married/        21        1   Southern       iii |
     |-------------------------------------------------------------------|
 16. | 16       19   Married/        16        1   Scotland       iii |
 17. | 17       28   Married/        21        1   Scotland   I or II |
 18. | 18       19   Married/        16        0   South Ea       iii |
 19. | 19       18   Married/        18        1   Wales an   IV or V |
 20. | 20       19   Married/        16        1   Wales an       iii |
     +-------------------------------------------------------------------+
```

All of the above 20 individuals were married by age 33.

To see the number of censored cases:

```
. tab event
```

35 of the 500 individuals were still single by the end of the observation period (age 33).

1.2 Discrete-Time Logit Models

Data preparation: the person-period file

Before fitting a discrete-time logit model, we must restructure the data into person-period format, i.e. with one record per year 'at risk' of partnering.

We carry out the following steps, working with the original data file:

(i) Calculate a duration variable (**dur**) with minimum value 1 rather than 16, i.e. **dur** = **age1st** – 16 + 1.

(ii) Expand the dataset so that each individual contributes **dur** records. For example, a person who married at age 21 will have 21 – 16 + 1 = 6 records.

(iii) For each person, create a variable (**t**) which indicates the time interval for each of their records (coded 1, 2, 3, ….). Transform this to **age** = **t** + 15 (coded 16, 17, ….)

(iv) Create a binary response (**y**) indicating whether an individual has partnered during each time interval. For all individuals, **y** is coded 0 for **age** = 16, . . ., **age1st**. For uncensored cases, **y** is replaced by 1 for **age** = **age1st**.

```
. use ncds, clear
. gen dur=age1st-16+1
. expand dur
. sort id
. by id: gen t=_n
. gen age=t+15
. gen y=0
. replace y=1 if (age==age1st & event==1)
```

Look at the first 20 records of the person-period file:

```
. list in 1/20, nol
```

	id	age1st	event	ageleft	female	region	fclass	dur	t	age	y
1.	1	21	1	18	0	2	2	6	1	16	0
2.	1	21	1	18	0	2	2	6	2	17	0
3.	1	21	1	18	0	2	2	6	3	18	0
4.	1	21	1	18	0	2	2	6	4	19	0
5.	1	21	1	18	0	2	2	6	5	20	0
6.	1	21	1	18	0	2	2	6	6	21	1
7.	2	31	1	21	0	4	2	16	1	16	0
8.	2	31	1	21	0	4	2	16	2	17	0
9.	2	31	1	21	0	4	2	16	3	18	0
10.	2	31	1	21	0	4	2	16	4	19	0
11.	2	31	1	21	0	4	2	16	5	20	0
12.	2	31	1	21	0	4	2	16	6	21	0
13.	2	31	1	21	0	4	2	16	7	22	0
14.	2	31	1	21	0	4	2	16	8	23	0
15.	2	31	1	21	0	4	2	16	9	24	0
16.	2	31	1	21	0	4	2	16	10	25	0
17.	2	31	1	21	0	4	2	16	11	26	0
18.	2	31	1	21	0	4	2	16	12	27	0
19.	2	31	1	21	0	4	2	16	13	28	0
20.	2	31	1	21	0	4	2	16	14	29	0

The first individual has 6 records, one for each age from 16 to 21. Notice that their time-invariant characteristics, **female** and **fclass**, take the same value for each record.

Next we calculate the time-varying covariate **fulltime** by comparing **ageleft** with **age** for each record.

```
. gen fulltime=1
. replace fulltime=0 if age>ageleft
```

Fitting age as a step function

The first model we fit will treat age as a categorical variable. We first need to calculate dummy variables for **t** (or from **age** – the results will be the same whichever we use).

```
. tab t, gen(t)
```

18 dummies variables, called **t1**-**t18**, will be added to the dataset. These are dummies for ages 16 to 33.

We will include **t2**-**t18** in our model, so that we are taking age 16 as the reference category. The model also includes **female** and **fulltime**.

```
. logit y t2-t18 female fulltime
```

We can use Stata's post-estimation commands to calculate predicted probabilities for y, i.e. the discrete-time hazard. We will plot the hazard for the sub-sample of men who have left full-time education (as a way of fixing the values of the covariates female and fulltime).

```
. predict haz, pr
. sort t
. scatter haz t if (female==0 & fulltime==0)
```

You should see that the hazard increases then decreases.

Fitting a quadratic in age

Next we fit a quadratic for age by including t and t^2 in the model as covariates.

```
. gen tsq=t*t
. logit y t tsq female fulltime
```

and calculate and plot the predicted hazard

```
. predict hazquad, pr
. sort t
. scatter hazquad t if (female==0 & fulltime==0)
```

Allowing for non-proportional effects of gender

We allow the effect of gender to depend on age by extending the model to include interactions between **female** and **t** and between **female** and $\mathbf{t^2}$.

```
. gen t_fem=t*female
. gen tsq_fem=tsq*female
. logit y t tsq female t_fem tsq_fem fulltime
```

We can test for non-proportionality by testing whether the coefficients of **t_fem** and **tsq_fem** are both equal to zero, using a Wald test.

```
. test t_fem tsq_fem
```

The p-value for the test is 0.01, so we reject the null that both interaction effects are zero and conclude that the effect of gender is non-proportional.

Finally, we plot the hazard for men and women on the same plot (for the sub-sample with **fulltime**==0).

```
. predict hazint, pr
. sort t
. scatter hazint t if female==1 & fulltime==0, legend(label(1 "F")) || ///
scatter hazint t if female==0 & fulltime==0, legend(label(2 "M"))
```

(Note the use of the continuation symbols /// which allows us to break a single Stata command over several lines of text.)

1.3 Further exercises

Modify the do-file **prac1.do** to address the following questions:

- Does the hazard of time to first marriage for males differ across region of residence at 16?
- Are regional differences for this group proportional at all ages?

(Hints: Drop observations belonging to females at the start. Use a quadratic in age to capture the baseline hazard.)

Practical 2: Discrete-Time Logit Models for Recurrent Events

Note that the following Stata syntax is contained in the annotated do-file **prac2.do**

You can either type in each command, or read prac2.do into the Do-file Editor and select the relevant syntax for each stage of the analysis.

To open the Do-file Editor, go to the **File** menu and select **Open**. Change the file type to **Do Files (*.,do, *.ado)** and locate **prac2.do**. Highlight the syntax you want to run, then hover over the icons on the tool bar until you find **Execute Selection (do)**.

See Practical 1 for more detailed instructions.

2.1 Introduction to BHPS Dataset

In these exercises, we will be applying recurrent events models in analyses of women's employment transitions. We use data from the British Household Panel Survey (BHPS), which began in 1991. Adult household members have been reinterviewed each year, and members of new households formed from the original sample households were also followed.

We will be using compete work, marital, cohabitation and fertility histories that have been constructed from a combination of retrospective data collected at Wave 2 (in 1992) and panel data collected for subsequent years. We focus on employment histories from age 16 to the age of interview in 2005, with histories censored at retirement age 60. In this exercise, we focus on transitions from *non-employment* (including unemployment and out of the labour market) to *employment* (full-time or part-time work and self-employment). A non-employment spell is defined as a continuous period out of employment. Spell durations were rounded to the nearest year[1] and the data were then expanded to person-episode-year format.

We will consider a range of time-varying covariates that were constructed from the various event histories, including the number of years in the current state (the duration variable *t*), age, characteristics of the previous job (if any), marital status, and indicators of pregnancy and the number and age of children.

[1] Employment status is actually available for each month, and it would be preferable to analyse durations in months. Note, however, that grouping durations into years does not lead to the omission of any transitions. Every episode is taken into account, even those lasting less than a year, but we do not distinguish between those that last 1 month and those that last a year.

For the purposes of illustration, a random sample of 2000 women has been selected, which reduces to 1994 after dropping cases with incomplete covariate information.

The Stata data file **bhps.dta** contains the following variables:

Variable	Description	Coding
pid	Person identifier	
spell	Employment/non-employment episode identifier	Reset to 1 when **pid** changes
t	Year of episode (reset to 1 at start of each episode)	
tgp	Year of episode with t ≥ 10 grouped together	
employ	Employment status	0 = non-employed 1 = employed
event	Employment transition indicator	0 = no change in status 1 = change in employment status
event2	Transition to fulltime/part-time job (relevant only if employ=0; coded 0 if employ=1)	0 = no change (still non-employed) 1 = fulltime job 2 = part-time job
jobclass	Occupation class (coded 0 if employ=0)†	1 = professional, managerial, technical 2 = skilled non-manual, manual 3 = partly skilled, unskilled
ptime	Part-time employment (coded 0 if employ=0) †	0 = fulltime 1 = part-time
everjob	Ever worked	0 = Never worked 1 = Currently or previously employed
ljobclass2 **ljobclass3**	Dummies for occupation class of last job (coded 0 if everjob=0)*	**ljobclass2** = 1 if skilled non-man., man. **ljobclass3** = 1 if partly skilled, unskilled
lptime	Last job was part-time (coded 0 if everjob=0)*	0 = fulltime 1 = part-time
ageg8	Age in years, grouped (time-varying)	5-year categories from 16-19 to 45-49, then 50-59
marstat	Marital status (time-varying)	1 = single 2 = married 3 = cohabiting
birth	Due to give birth within next year (time-varying)	0 = no 1 = yes
nchildy	Number of children aged ≤ 5 years	0 = none 1 = one 2 = two or more
nchildo	Number of children aged > 5 years	0 = none 1 = one 2 = two or more

† **jobclass** and **ptime** will only be considered in analysis of transition out of employment (employ=1).

*The occupation class and part-time status of the last job are only relevant for women who have previously worked (i.e. with **everjob**=1). By including **everjob** in the model, and coding **ljobclass2**, **ljobclass3** and **lptime** as zero for

everjob=0, the coefficients of **ljobclass2**, **ljobclass3** and **lptime** can be interpreted as effects of previous class and part-time status *among women who have worked before*.

2.2 Exploring the Data Structure

Before fitting any models, we explore the structure of the data. We read in the data file; sort by person ID, spell and time interval; and then view selected variables for the first 30 records.

```
. use bhps, clear
. sort pid spell t
. list pid spell t employ event everjob lptime marstat in 1/30
```

	pid	spell	t	employ	event	everjob	lptime	marstat
1.	10014578	1	1	Employed	0	1	0	Married
2.	10014578	1	2	Employed	0	1	0	Married
3.	10014578	1	3	Employed	0	1	0	Married
4.	10014578	1	4	Employed	0	1	0	Married
5.	10014578	1	5	Employed	0	1	0	Married
6.	10014578	1	6	Employed	0	1	0	Married
7.	10014578	1	7	Employed	0	1	0	Married
8.	10014578	1	8	Employed	0	1	0	Married
9.	10014578	1	9	Employed	0	1	0	Married
10.	10014578	1	10	Employed	0	1	0	Married
11.	10014578	1	11	Employed	1	1	0	Married
12.	10014578	2	1	Not employed	0	1	0	Married
13.	10014578	2	2	Not employed	0	1	0	Married
14.	10014578	2	3	Not employed	0	1	0	Married
15.	10014578	2	4	Not employed	0	1	0	Married
16.	10014578	2	5	Not employed	0	1	0	Married
17.	10014578	2	6	Not employed	0	1	0	Married
18.	10017933	1	1	Employed	0	1	0	Single
19.	10017933	1	2	Employed	0	1	0	Single
20.	10017933	1	3	Employed	0	1	0	Single
21.	10017933	1	4	Employed	0	1	0	Single
22.	10017933	1	5	Employed	0	1	0	Single
23.	10040331	1	1	Not employed	0	0	0	Married
24.	10040331	1	2	Not employed	0	0	0	Married
25.	10040331	1	3	Not employed	0	0	0	Married
26.	10040331	1	4	Not employed	0	0	0	Married
27.	10040331	1	5	Not employed	0	0	0	Married
28.	10040331	1	6	Not employed	0	0	0	Married
29.	10040331	1	7	Not employed	0	0	0	Single
30.	10040331	1	8	Not employed	0	0	0	Single

Stata has some useful commands for manipulating longitudinal data, in particular allowing us to calculate summary statistics for each individual (e.g. the total number of spells).

Total number of women

First, we obtain a count of the total number of women in the data file. The simplest way to do this is to use the 'codebook' command for the individual ID (pid).

```
. codebook pid
```

Stata will return the number of unique values which is 1994, along with other summary statistics.

Alternatively we can create an indicator for the first record for each woman. The following syntax creates an indicator **firstwom** which equals 1 for the first record (and is missing for all other records). **_n** is an internal Stata variable which, when used with `by pid`, is the observation number for each record within an individual. We then request a summary of **pid** for the woman-based file (by selecting the first record for each woman). (We could have summarised any variable; the important thing is that we have selected 1 record per woman.)

```
. by pid: gen firstwom=1 if _n==1
. sum pid if firstwom==1
```

Selecting non-employment spells

In this exercise we focus on transitions into employment. Hence we want to exclude spells in which the woman is employed (**employ**=1). After dropping these observations from the datafile we can check the number of women who experienced at least one non-employment spell. We need to recreate the `firstwom` indicator for the new restricted sample because some women's first record may have been for an employment spell and that record will have been dropped.

```
. drop if employ==1

. drop firstwom
. by pid: gen firstwom=1 if _n==1
. sum pid if firstwom==1
```

You should find that 1399 women experienced at least one non-employment spell.

Total number of non-employment spells

Next we obtain a count of the total number of (non-employment) spells in the data file. We do this by creating an indicator **lastsp** which identifies the last record for each spell (within a woman), rather like **firstwom**. **_N** is an internal Stata variable which, when used with `by pid spell`, is equal to the total number of records for each spell. The last record for a spell will therefore have **_n** = **_N**. We then request the summary of one of the variables (e.g. **pid**) for the spell-based file (by selecting the last record for each spell).

```
. by pid spell: gen lastsp=1 if _n==_N
. sum pid if lastsp==1
```

Distribution of the total number of spells per woman

To obtain a count of the number of spells per woman (**nspell**), we count the number of records with a non-missing value for **lastsp**. We then tabulate **nspell** for the woman-based file (by selecting the first record for each woman).

```
. by pid: egen nspell=count(lastsp)
. drop lastsp
. tab nspell if firstwom==1
```

2.3 Modelling Recurrent Events in Stata

We will begin by fitting a discrete-time model with only duration effects, including dummy variables for **tgp** (which has durations of 10 or more years grouped into one category). The dummies are named **tgp1**-**tgp10** and we take the first category as the reference. In order to fit a random effects logit model, using the `xtlogit` command, we first use the `xtset` command to specify the individual identifier.

```
. tab tgp, gen(tgp)
```

```
. xtset pid
. xtlogit event tgp2-tgp10, re
```

```
Random-effects logistic regression              Number of obs      =     15297
Group variable: pid                             Number of groups   =      1399

Random effects u_i ~ Gaussian                   Obs per group: min =         1
                                                               avg =      10.9
                                                               max =        44

                                                Wald chi2(9)       =    180.59
Log likelihood  = -3684.6506                    Prob > chi2        =    0.0000

------------------------------------------------------------------------------
       event |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        tgp2 |  -.6371191   .1021843    -6.23   0.000    -.8373968   -.4368415
        tgp3 |  -1.053616   .1314592    -8.01   0.000    -1.311271   -.7959602
        tgp4 |  -1.410675   .1638596    -8.61   0.000    -1.731834   -1.089516
        tgp5 |   -1.34301   .1779576    -7.55   0.000    -1.691801   -.9942196
        tgp6 |  -1.197793   .1897274    -6.31   0.000    -1.569652    -.825934
        tgp7 |  -1.246341   .2115337    -5.89   0.000    -1.660939   -.8317423
        tgp8 |  -1.449068   .2468906    -5.87   0.000    -1.932965   -.9651716
        tgp9 |  -1.531057   .2737371    -5.59   0.000    -2.067572   -.9945422
       tgp10 |  -2.098082   .1841749   -11.39   0.000    -2.459058   -1.737106
       _cons |  -1.350212   .0972977   -13.88   0.000    -1.540912   -1.159512
-------------+----------------------------------------------------------------
    /lnsig2u |   1.067351   .1425229                      .7880109     1.34669
-------------+----------------------------------------------------------------
     sigma_u |   1.705188   .1215142                     1.482909    1.960786
         rho |    .469165   .0354952                      .400631    .5388821
------------------------------------------------------------------------------
Likelihood-ratio test of rho=0: chibar2(01) =   234.45 Prob >= chibar2 = 0.000
```

We find that the probability of entering employment decreases with the duration spent out of work. There is significant unobserved heterogeneity between women (see likelihood test of rho=0), and the standard deviation of the woman-level random effect is estimated as 1.705.

Now let's add dummy variables for **agegp** (taking the first category as the reference) and **everjob** and interpret the results.

```
. tab ageg8, gen(agegp)
. xtset pid
. xtlogit event tgp2-tgp10 agegp2-agegp8 everjob, re
```

```
------------------------------------------------------------------------------
       event |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
```

```
        tgp2 |  -.6381259   .0997176    -6.40   0.000    -.8335688    -.442683
        tgp3 |  -.9603769   .1299996    -7.39   0.000    -1.215172   -.7055823
        tgp4 |  -1.274958   .1635169    -7.80   0.000    -1.595445   -.9544709
        tgp5 |  -1.160597   .1779746    -6.52   0.000     -1.50942    -.811773
        tgp6 |  -.9743299   .1894804    -5.14   0.000    -1.345705   -.6029552
        tgp7 |  -.9764327   .2102316    -4.64   0.000    -1.388479   -.5643863
        tgp8 |  -1.149324   .2444203    -4.70   0.000    -1.628379    -.670269
        tgp9 |  -1.207394   .2704627    -4.46   0.000    -1.737491   -.6772967
       tgp10 |  -1.650968   .1729189    -9.55   0.000    -1.989883   -1.312054
      agegp2 |   .0880516   .1629492     0.54   0.589    -.2313229    .4074261
      agegp3 |   .1798995   .1612324     1.12   0.265    -.1361103    .4959092
      agegp4 |   .1525526   .1645784     0.93   0.354    -.1700151    .4751203
      agegp5 |   .1784371   .1695818     1.05   0.293    -.1539371    .5108112
      agegp6 |   .1354204   .1794666     0.75   0.451    -.2163278    .4871685
      agegp7 |  -.0646183    .187695    -0.34   0.731    -.4324937    .3032572
      agegp8 |  -1.181726   .1965003    -6.01   0.000    -1.566859   -.7965925
     everjob |   2.379486   .1018423    23.36   0.000     2.179879    2.579094
       _cons |  -2.468315   .1619436   -15.24   0.000    -2.785719   -2.150912
-------------+----------------------------------------------------------------
    /lnsig2u |  -.9991706   .3013405                     -1.589787   -.4085541
-------------+----------------------------------------------------------------
     sigma_u |   .6067822    .091424                      .4516293    .8152365
         rho |   .1006505   .0272773                      .0583797    .1680653
------------------------------------------------------------------------------
Likelihood-ratio test of rho=0: chibar2(01) =    18.45 Prob >= chibar2 = 0.000
```

We find that the probability of entering employment decreases with the duration spent out of work. There is little effect of age apart from a lower probability in the 50-59 category compared to all younger ages. Women who have worked before are more likely than those who have not to enter employment.

There is significant unobserved heterogeneity between women (see likelihood test of rho=0), and the standard deviation of the woman-level random effect is estimated as 0.607.

2.4 Prediction of individual discrete-time hazard probabilities

The coefficients estimated in the random effects logit model, when exponentiated, give us the effect on the odds that an individual transitions into employment, holding constant the values of their other covariates and their (unobserved) random effect. For example, the odds of a transition into employment are reduced by exp(-0.64) = 0.53 times when a year of non-employment has elapsed relative to less than a year passed in that state. This number is the same for all women regardless of their age, of whether they have ever had a job and of the strength of their unobserved tendency to enter employment. Variation in these factors, however, mean that a fall in the odds of 0.53 times can translate into very different effects on the probability scale for different groups of individuals, and it is often the effect of the average probabilities across all groups in which we are ultimately interested.

To illustrate, consider a woman aged 16-19 who has never had a job and has a low propensity to enter employment (and random effect of -1, one standard deviation below the mean). In the first year of her non-employment spell her predicted probability of entering employment is 0.030 and in the second year it falls to 0.016, a fall of 1.4 percentage points or nearly 50 percent. In contrast, a women aged 35-39 who has worked previously and who has a high tendency to employment (and random effect of 1) has a transition probability of 0.748 in the first year of the spell and 0.613 in the second year, a change of 13.5 percentage points or around 18 percent. In order to understand the implications of a given set of coefficients we need to simulate how probabilities change for a population with the characteristics observed in our sample.

This is straightforward for a model without random effects. We can 'switch on' and 'switch off' values of a particular covariate, keeping all the other covariates fixed at their observed values for each individual. This generates two hypothetical probabilities for each individual and the difference between the two gives the individual-specific effect of a unit change in the covariate of interest. These effects can then be averaged over particular sub-groups or the sample as a whole. In a random effects world, however, these hypothetical probabilities will depend on the (unobserved) value of the individual's random effect. Different choices of u_j give rise to differences in the gap between the 'on' and 'off' probabilities.

Here we present two options for choosing values of the random effects. The first sets every individual's random effect to the mean value for the sample – zero. These probabilities have a *cluster-specific* (or conditional) interpretation because we are conditioning on a particular value of the random effect which is fixed across individuals; they refer to a hypothetical individual with the mean random effect value. The second method recognizes that the effects of high and low random effect values on the predicted probabilities are generally not symmetric. Where the underlying probability that an event occurs is low, for example, the increase in probability associated with a random effect one standard deviation above the mean is larger than the decrease associated with a random effect one standard deviation below the mean. Even though the effects are normally (and symmetrically) distributed among the population they will not cancel each other out when translated onto the probability scale. The second method, therefore, uses simulation to assign each individual an effect randomly which then enters the calculation of their predicted probabilities. Predicted probabilities from this method have a *population-averaged* (or marginal) interpretation because they are averaged across different values of the random effect, according to its distribution in the population.

Let's see how this works in practice on the model estimated at the end of the previous section. We will calculate predicted transition probabilities for each individual at each of the ten elapsed time points of a

non-employment spell. Individuals will retain their own age covariates but we will contrast their probabilities in the situations in which they have, and have not, ever had a job. Note that the probabilities we will calculate are the discrete-time hazard functions, i.e. the conditional probabilities of a transition in interval t given that no transition has occurred before t. In many cases the survival function, which is derived from the conditional probabilities, is more useful for interpretation; we will return to this later.

Method 1: Predictions with u fixed at zero (cluster-specific probabilities)

First we re-estimate the underlying model and store the results with the name **m1**:

```
. xtlogit event tgp2-tgp10 age2-age8 everjob, re
. estimates store m1
```

To begin we apply the first method, assuming a universal random effect value of zero, for all individuals. We begin with predictions in which all individuals are assumed never to have had a job (**everjob**=0). We set the variables **tgp2**-**tgp10** to zero and calculate the linear prediction for women in the first year of a non-employment spell (the reference case), saving it as **xbt1e0**. We then transform the linear predictor to the probability scale using the inverse logit function and save the resulting probability **pt1e0**.

```
. replace everjob=0

. foreach i of num 2/10    {
     replace tgp`i'=0
     }

. estimates for m1: predict xbt1e0, xb
. gen pt1e0=invlogit(xbt1e0)
. drop xbt1e0
```

We then switch on each duration dummy one at a time, recalculate the probabilities for that particular time interval then switch it off again, giving nine more predictions **pt2e0**,...,**pt10e0**.

```
. foreach i of num 2/10    {
          replace tgp`i'=1
```

```
            estimates for m1: predict xbt`i'e0, xb
            gen pt`i'e0=invlogit(xbt`i'e0)
            drop xbt`i'e0
            replace tgp`i'=0
            }
```

The process is then repeated with the variable **everjob** set to 1 for all individuals, generating probabilities indexed by **e1** rather than **e0**. (Note that the two steps can be combined by incorporating the loop for **everjob**=0, 1 into the loop over the duration dummies. They are shown separately here to avoid the use of multiple subscript variables.)

```
. replace everjob=1
. foreach i of num 2/10    {
      replace tgp`i'=0
      }
. estimates for m1: predict xbt1e1, xb
. gen pt1e1=invlogit(xbt1e1)
. drop xbt1e1
 . foreach i of num 2/10   {
            replace tgp`i'=1
            estimates for m1: predict xbt`i'e1, xb
            gen pt`i'e1=invlogit(xbt`i'e1)
            drop xbt`i'e1
            replace tgp`i'=0
            }
```

We now have 20 probability variables that are specific to each individual, covering the 10 durations in both states of **everjob**. The average of the individual predictions can be viewed using the summarize command:

```
. sum pt1e0-pt10e1, sep(0)
```

Variable	Obs	Mean	Std. Dev.	Min	Max
pt1e0	15297	.0731928	.0262172	.0253317	.0920869
pt2e0	15297	.0402366	.0146392	.0135441	.050857
pt3e0	15297	.0295229	.0107938	.0098497	.0373703
pt4e0	15297	.0217511	.0079801	.0072104	.0275618
pt5e0	15297	.0243137	.0089101	.0080769	.0307983

```
      pt6e0 |     15297    .0291273    .0106511    .0097145    .0368716
      pt7e0 |     15297    .0290682    .0106297    .0096943     .036797
      pt8e0 |     15297    .0245816    .0090072    .0081677    .0311366
      pt9e0 |     15297    .0232314    .0085176    .0077105    .0294317
     pt10e0 |     15297    .0150506    .0055383    .0049618    .0190887
      pt1e1 |     15297    .4422568    .1207993    .2191623    .5227519
      pt2e1 |     15297    .3019904    .0939702    .1291291    .3665478
      pt3e1 |     15297    .2408633    .0783517     .097007    .2953986
      pt4e1 |     15297    .1895212    .0637182    .0727285    .2343539
      pt5e1 |     15297    .2071936    .0689006    .0808279    .2554932
      pt6e1 |     15297    .2384059    .0776813    .0957916    .2925028
      pt7e1 |     15297     .238037    .0775804    .0956096    .2920678
      pt8e1 |     15297    .2089974    .0694212    .0816693    .2576434
      pt9e1 |     15297    .1998241    .0667575    .0774186    .2466939
     pt10e1 |     15297    .1393654    .0482369    .0510998    .1736613
```

So for example, the mean probability that a woman who has never worked enters employment between five and six years into a non-employment spell (**pt5e0**) is 0.024. The values of this probability among the sample range from 0.008 to 0.031 depending on the age of the individual in question. (Each probability can take one of eight possible values, one for each of the age groups in the sample. You can see this by using, e.g. `codebook` **`pt5e0`**.)

Method 2: Predictions with simulated values of u (population-averaged probabilities)

The second prediction method, in which individual random effect values are simulated, requires a little modification to the code above. First we need to create an indicator for the first observation of each individual. This will be used when deriving an individual random effect that is constant across time for each woman.

```
. sort pid
. by pid: gen firstob=_n==1
```

Next, as we will be using a random number generator to draw the individual random effects, it is useful to set the random seed to a fixed value so that the results are the same whenever we run the do file:

```
. set seed 121
```

We begin, as before, with the value of **everjob** set universally to zero. The first probability to be calculated is that of entrance to employment in year 1, the base case (**pst1e0**). The difference to this step compared with the first method is that we generate an individual-specific time-invariant random effect **u** that is added on to the linear prediction before we use the `invlogit` function to derive the probability. The function `rnormal(`*m*`,`*s*`)` returns random numbers drawn from a normal distribution with mean *m* and standard deviation *s*. Here we set *s* to the estimated random effect standard deviation which is stored in the results as **e(sigma_u)**.

```
. replace everjob=0

. foreach i of num 2/10    {
      replace tgp`i'=0
      }

. estimates for m1: predict xbt1e0, xb
. estimates for m1: gen u=rnormal(0,e(sigma_u)) if firstob==1
. by pid: replace u=u[_n-1] if u==.
. gen pst1e0=invlogit(xbt1e0+u)
. drop xbt1e0 u
```

The same modification is then made to the sections of code that calculate **pst2e0**,..., **pst10e0**, **pst1e1** and **pst2e1**,..., **pst10e1**.

```
. foreach i of num 2/10    {
           replace tgp`i'=1
           estimates for m1: predict xbt`i'e0, xb
           estimates for m1: gen u=rnormal(0,e(sigma_u)) if firstob==1
           by pid: replace u=u[_n-1] if u==.
           gen pst`i'e0=invlogit(xbt`i'e0+u)
           drop xbt`i'e0 u
           replace tgp`i'=0
           }

. replace everjob=1
. foreach i of num 2/10    {
      replace tgp`i'=0
      }
. estimates for m1: predict xbt1e1, xb
. estimates for m1: gen u=rnormal(0,e(sigma_u)) if firstob==1
. by pid: replace u=u[_n-1] if u==.
. gen pst1e1=invlogit(xbt1e1+u)
. drop xbt1e1 u
. foreach i of num 2/10    {
           replace tgp`i'=1
```

```
            estimates for m1: predict xbt`i'e1, xb
            estimates for m1: gen u=rnormal(0,e(sigma_u)) if firstob==1
            by pid: replace u=u[_n-1] if u==.
            gen pst`i'e1=invlogit(xbt`i'e1+u)
            drop xbt`i'e1 u
            replace tgp`i'=0
            }
```

Again, we can view the average of the individual predicted probabilities using

```
. sum pst1e0-pst10e1, sep(0)
```

Note that instead of the eight possible values taken by **pt5e0** using the first method above, typing `codebook` **`pst5e0`** shows that it now takes one of 3,894 values. The greater variation is, of course, induced by the variation in random effects across individuals.

2.5 Creating a dataset of average predictions

Currently we have a dataset of hypothetical probabilities stored at the individual level. Typically we are interested in the averages for different sub-groups rather than predictions for any particular individual. Converting the data to a dataset of averages (rather than viewing the means using `summarize`, for example) has the advantage that we can manipulate and graph the average probabilities.

First we collapse the data so we have a single row vector containing the mean values of each of our 40 probabilities (10 time points × 2 values of **everjob** × 2 methods). We are going to transform the data into two variables, each of which contains all the probabilities from a single method. We need 20 rows of data for each variable, indexed by all the possible combinations of **tgp** and **everjob**.

```
. collapse (mean) pt1e0-pt10e1 pst1e0-pst10e1

. expand 20
. gen everjob=0 in 1/10
. replace everjob=1 in 11/20
. sort everjob
. by everjob: gen tgp=_n
```

Having set up the dataset structure, we now create the two column variables and fill them in with the corresponding probability values.

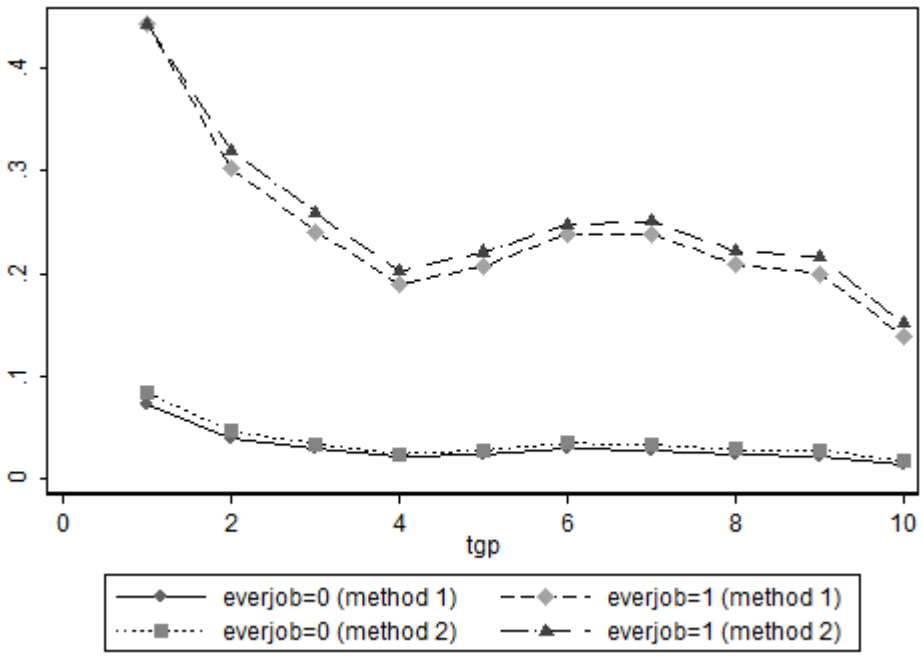
```
. gen pmethod1=.
. gen pmethod2=.

. foreach i of num 1/10    {
     foreach j of num 0/1  {
          replace pmethod1=pt`i'e`j' if tgp==`i' & everjob==`j'
          replace pmethod2=pst`i'e`j' if tgp==`i' & everjob==`j'
          drop pt`i'e`j' pst`i'e`j'
          }
     }
```

Now we can view and plot the hazard functions:

```
. list

. twoway (connected pmethod1 tgp if everjob==0) ///
(connected pmethod1 tgp if everjob==1) ///
(connected pmethod2 tgp if everjob==0) ///
(connected pmethod2 tgp if everjob==1), ///
legend(order(1 "everjob=0 (method 1)" 2 "everjob=1 (method 1)" ///
3 "everjob=0 (method 2)" 4 "everjob=1 (method 2)")) scheme(s1mono)
```



The pattern in the transition probabilities is the same using both methods but assuming a zero random effect value for everyone (method 1) always results in lower probabilities than the simulation method 2. Why? In this example the average probabilities are always below 0.5, so a positive random effect raises the predicted probability by more than a negative random effect of the same absolute size lowers it. Since positive and negative values are equally likely among the population, the average is pulled upwards. We also find that the probability of entering employment generally decreases with duration non-employed (with a few bumps, which is consistent with the coefficients for **tgp**). At each duration, the probability of entering employment is much higher for women who have worked before. However, remember that we have fitted a proportional odds model which forces the difference in the log-odds of a transition for **everjob**=0 and 1 to be fixed across values of **tgp**.

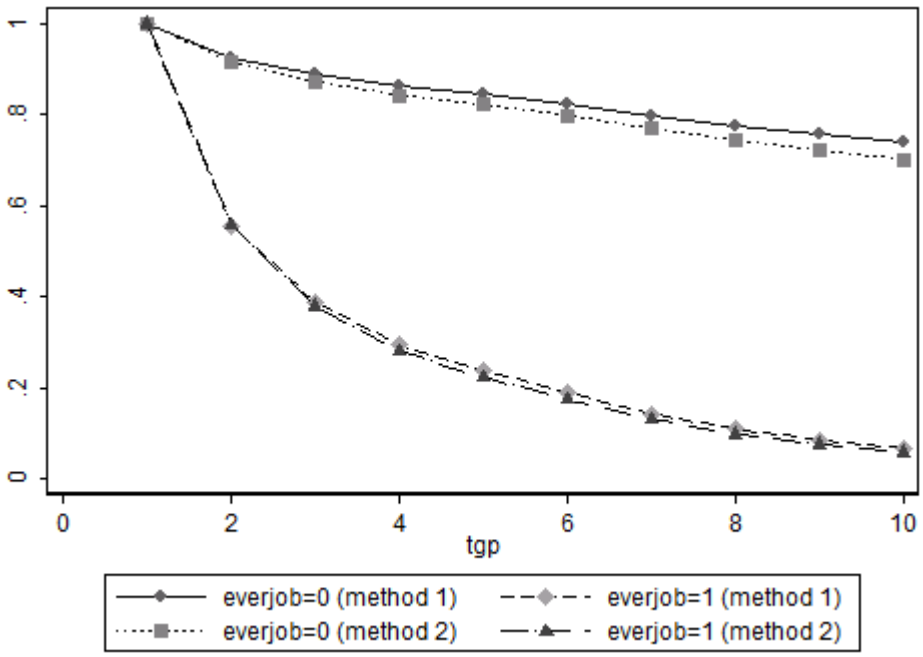
Generating survival functions

The hazard is one way of summarizing how the probability of exit varies with time spent in a state. At a particular point it tells us, for example, the probability a women enters employment before the sixth year given that she has had a non-employment spell of five years. However, often we are interested in more aggregated probabilities such as the question: what is the probability that a women entering non-employment will remain out of employment for at least five years? This question is answered by the survival function S_t. (The reverse question of the probability she will be enter employment within six years is answered by the cumulative distribution function, $1 - S_t$.)

The formula for deriving the survival probability at time t is $S_t = S_{t-1}(1 - p_{t-1})$ where p_t is the conditional hazard probability we have already calculated. We can implement it in Stata for the two sets of probability estimates as follows:

```
. sort everjob tgp
. gen smethod1=1
. gen smethod2=1
. by everjob: replace smethod1=smethod1[_n-1]*(1-pmethod1[_n-1]) if tgp>1
. by everjob: replace smethod2=smethod2[_n-1]*(1-pmethod2[_n-1]) if tgp>1
```

The command `list` allows us to view the survival probabilities and we can also graph them.



Practical 3: Models for Multiple States

In this practical, we model British women's entry into employment jointly with their exits from employment using a two-state duration model. This involves specifying two equations: one for transitions *into* employment, and a second for transitions *out of* employment. Each equation includes a woman-level random effect, and the equations are linked by allowing for a correlation between these random effects.

At the end of the practical (if time permits), we analyse transitions between employment and non-employment using an autoregressive model which includes lagged employment status as a covariate, in stead of duration.

The analysis is based on 1994 women. There are a total of 2284 non-employment and 2700 employment episodes. Combining non-employment and employment episodes gives a total of 33,083 person-year observations.

3.1 Specifying a two-state duration model in Sabre

Two-state duration models are essentially random coefficient models. They can be fitted in Stata v10 (and later versions) using `xtmelogit`. We will be using Sabre because Sabre is much faster and can be run from within Stata. In Sabre (and other software packages), a two-state model is fitted as a *bivariate* model. For each state, we have a binary response indicating whether a transition has occurred; together these form a bivariate response. In the data file **bhps.dta**, this bivariate response is the binary transition indicator **event**. To determine the type of transition, we need to consider **event** together with the origin state (**employ**). For example, a transition out of non-employment is indicated by **employ**=0 & **event**=1.

Details of all Sabre commands are available online.[2] When using Sabre in Stata, all Sabre commands are preceded by `sabre,`. Note that there are no facilities for calculating predicted probabilities in Sabre, nor is it possible to store the parameter estimates in Stata for post-estimation calculations. Section 3.3 shows a way of reading in the parameter estimates that overcomes this problem.

The file **prac3.do** contains Stata and Sabre commands for preparing the data for a two-state analysis, reading the data into Sabre and fitting a random effects two-state model.

[2] You can download the complete Sabre user guide from http://sabre.lancs.ac.uk/sabreStata_coursebook.pdf

The models take a few minutes to estimate so run the do-file and, while you are waiting, read the following descriptions of what it does.

We begin with some data manipulation in Stata. This involves creating dummy variables for covariates **tgp** and **ageg8** (taking the first category as the reference in each case), dummy variables for each state (**r1** for non-employment and **r2** for employment), and interactions between **r1** and **r2** with duration (**tgp**), age (**ageg8**) and, for non-employment only, **everjob**. Variables with prefix **r1_** will be covariates in the transitions into employment equation while those with prefix **r2_** will be covariates in the transitions into non-employment equation.

```
use bhps, clear
sort pid spell t

* Create dummy variables for all categorical variables (taking 1st category as reference
in each case)

local i = 2
while `i' <=10 {
  gen tgp`i' = tgp==`i'
  local i = `i' + 1
}
local i = 2
while `i' <=8 {
  gen age`i' = ageg8==`i'
  local i = `i' + 1
}

*Create dummies for employment and non-employment states
*Create response index (1=non-employment, 2=employment)
gen r1 = employ==0
gen r2 = employ==1
gen r=employ+1
gen r1_t2=r1*tgp2
gen r1_t3=r1*tgp3
gen r1_t4=r1*tgp4
gen r1_t5=r1*tgp5
gen r1_t6=r1*tgp6
gen r1_t7=r1*tgp7
gen r1_t8=r1*tgp8
gen r1_t9=r1*tgp9
gen r1_t10=r1*tgp10
gen r1_age2=r1*age2
gen r1_age3=r1*age3
gen r1_age4=r1*age4
gen r1_age5=r1*age5
gen r1_age6=r1*age6
gen r1_age7=r1*age7
gen r1_age8=r1*age8
gen r1_ejob=r1*everjob

gen r2_t2=r2*tgp2
gen r2_t3=r2*tgp3
gen r2_t4=r2*tgp4
gen r2_t5=r2*tgp5
gen r2_t6=r2*tgp6
```

```
gen r2_t7=r2*tgp7
gen r2_t8=r2*tgp8
gen r2_t9=r2*tgp9
gen r2_t10=r2*tgp10
gen r2_age2=r2*age2
gen r2_age3=r2*age3
gen r2_age4=r2*age4
gen r2_age5=r2*age5
gen r2_age6=r2*age6
gen r2_age7=r2*age7
gen r2_age8=r2*age8

compress
```

The next step is to declare the list of variables that will be used in the analysis, and then to read the data into Sabre. (We use the continuation symbols /// so that we can have Stata commands over several lines.)

```
sabre, data pid r r1 r2 event ///
   r1_t2 r1_t3 r1_t4 r1_t5 r1_t6 r1_t7 r1_t8 r1_t9 r1_t10 ///
   r1_age2 r1_age3 r1_age4 r1_age5 r1_age6 r1_age7 r1_age8 r1_ejob ///
   r2_t2 r2_t3 r2_t4 r2_t5 r2_t6 r2_t7 r2_t8 r2_t9 r2_t10 ///
   r2_age2 r2_age3 r2_age4 r2_age5 r2_age6 r2_age7 r2_age8
sabre pid r r1 r2 event ///
   r1_t2 r1_t3 r1_t4 r1_t5 r1_t6 r1_t7 r1_t8 r1_t9 r1_t10 ///
   r1_age2 r1_age3 r1_age4 r1_age5 r1_age6 r1_age7 r1_age8 r1_ejob ///
   r2_t2 r2_t3 r2_t4 r2_t5 r2_t6 r2_t7 r2_t8 r2_t9 r2_t10 ///
   r2_age2 r2_age3 r2_age4 r2_age5 r2_age6 r2_age7 r2_age8, read
```

To set up the model, we need to specify the following:

- dependent variable (**event**)
- type of model (bivariate, b)
- individual identifier (**pid**)
- distribution of each response (binomial, b)
- link function for each response (logit, l)
- variable indexing the response (**r**)
- variables whose coefficients will be the intercept terms in each equation (**r1**, **r2**)
- number of variables in the first equation (18 in **r1** equation)

```
sabre, yvar event
sabre, model b
sabre, case pid
sabre, family first=b second=b
sabre, link first=l second=l
sabre, rvar r
sabre, constant first=r1 second=r2
sabre, nvar 18
```

We can now fit the model. The last two Sabre commands ask for the model specification (m) and parameter estimates (e) to be displayed.

```
sabre, fit r1 r1_t2 r1_t3 r1_t4 r1_t5 r1_t6 r1_t7 r1_t8 r1_t9 r1_t10 ///
   r1_age2 r1_age3 r1_age4 r1_age5 r1_age6 r1_age7 r1_age8 r1_ejob ///
   r2 r2_t2 r2_t3 r2_t4 r2_t5 r2_t6 r2_t7 r2_t8 r2_t9 r2_t10 ///
   r2_age2 r2_age3 r2_age4 r2_age5 r2_age6 r2_age7 r2_age8
sabre, dis m
sabre, dis e
```

3.2 Interpretation of a simple model

The parameter estimates are given below.

Parameter	Estimate	Std. Err.
r1	-2.4658	0.16232
r1_t2	-0.65553	0.99460E-01
r1_t3	-0.99362	0.13019
r1_t4	-1.3169	0.16397
r1_t5	-1.2021	0.17838
r1_t6	-1.0182	0.19003
r1_t7	-1.0232	0.21112
r1_t8	-1.1891	0.24494
r1_t9	-1.2477	0.27089
r1_t10	-1.6876	0.17308
r1_age2	0.94456E-01	0.16162
r1_age3	0.17794	0.15943
r1_age4	0.14976	0.16261
r1_age5	0.17955	0.16735
r1_age6	0.13883	0.17733
r1_age7	-0.51148E-01	0.18541
r1_age8	-1.1370	0.19426
r1_ejob	2.1568	0.11085
r2	-1.4638	0.13720
r2_t2	-0.48025	0.90781E-01
r2_t3	-0.81950	0.11200
r2_t4	-0.63996	0.11954
r2_t5	-0.77101	0.13853
r2_t6	-1.0303	0.16497
r2_t7	-1.4016	0.20436
r2_t8	-0.72566	0.17274
r2_t9	-1.1784	0.22156
r2_t10	-0.86606	0.14067
r2_age2	-0.45316	0.14731
r2_age3	-0.37672	0.14796
r2_age4	-0.52813	0.15255
r2_age5	-0.83986	0.15942
r2_age6	-1.1249	0.17007
r2_age7	-0.92228	0.17698
r2_age8	-0.95878	0.18455
scale1	0.58410	0.96386E-01
scale2	1.0847	0.70528E-01
corr	0.56736	0.13860

The estimates for variables **r1_** are effects on the log-odds of a transition *into* employment (i.e. out of non-employment) and the woman-level random effect standard deviation is **scale1**. The estimates for **r2_** are

effects on the log-odds of a transition *out of* employment and **scale2** the woman-level standard deviation. The correlation between the random effects is **corr**.

The effects of duration, age and **everjob** on transitions into employment are all in the same directions as in the single-state model of Practical 2. Turning to the second equation, we find that the probability of a transition out of employment decreases with the duration employed. We also find strong age effects with older women being less likely to exit employment. This age effect is likely to be at least partly explained by younger women taking time out of paid work to raise children. Finally, we find a positive residual correlation between the probability of entering and exiting employment (see lecture notes for interpretation).

3.3 Predicted probabilities

In Practical 2 (section 2.4) we saw how to calculate predicted discrete-time hazard probabilities in order to assess the magnitude of the effects of covariates on the probability scale. For models fitted using Stata functions such as `xtlogit` (or `runmlwin`), this task is made easier by Stata's post-estimation `predict` command. For models fitted using Sabre, however, the parameter estimates are not stored in Stata so we have to output the results to a text file, edit the file so that only the results table remains, and import the results back into Stata in matrix form.

The Stata do-file **prac3_predprob.do**. The do-file has been annotated, but we give an overview of the steps here. Much of the syntax has been copied directly from **prac2.do** and **prac3.do**.

- The Sabre results were written to a text log file, which was edited to strip away all output except the results table shown in Section 3.2. This edited output was then saved as **prac3_parests.txt**. The file contains the 3 columns of the results table – parameter name (a string variable), estimate and standard error – which are read into Stata as 3 variables using the `infile` command.

- The estimated coefficients are read into two matrices: br1 for the 'r1' equation and br2 for the 'r2' equation. The estimated random effect standard deviations and correlation are stored as scalars (constants).

- Next we read in the BHPS data and derive the explanatory variables for the two equations (as in **prac3.do**).

- For illustration, we calculate probabilities of making a transition from non-employment into employment using estimates for the 'r1' equation. These predictions are made only for women who were observed in non-employment over the observation period. As in Practical 2, predictions are made for each duration (**tgp**) and category of the binary variable **everjob**.

- The syntax for calculating the predictions closely follows that in **prac2.do**. There are only two major differences:

  (i) We have to calculate the linear predictor $\beta_0 \mathrm{r1} + \beta_1 \mathrm{r1\_t2} \ldots + \beta_{18} \mathrm{r1\_ejob}$ 'manually' as the predict command is not available to us. This involves multiplying each element of the coefficient matrix br1 by the relevant covariate, and summing the results.

  (ii) In this two-state model there are now two random effects which follow a bivariate normal distribution. Therefore, in the simulation method, we must generate two random effects (using the `drawnorm` command) even though we will only use the random effect for the 'r1' equation in the predictions.

- Having calculated predicted probabilities for each individual, we take averages and plot them for each value of **tgp** and **everjob**. The syntax for doing this is exactly the same as in **prac2.do**.

3.4 Further exercises

Modify the Stata do-file **prac3.do** to include the following additional covariates in the two equations.

Transitions into employment (**r1**): **ljobclass2**, **ljobclass3**, **lptime**, **marstat**, **birth**, **nchildy** and **nchildo**

Transitions out of employment (**r2**): **jobclass2**, **jobclass3**, **ptime**, **marstat**, **birth**, **nchildy** and **nchildo**

Interpret the full model.

3.5 Other software

The model fitted in Section 3.1 can also be estimated within Stata using the `xtmelogit` command and using MLwiN via the `runmlwin` command. Code to do this is provided in **prac3_xtmelogit.do** (takes around 2 hours to run) and in **prac3_mlwin.do** (takes around 20 minutes).

3.6 Autoregressive models

An alternative way of modelling transitions between states is to include the lagged response as a predictor, instead of the duration in the current state. The Stata do-file **prac3_ar1.do** gives annotated syntax for fitting first-order autoregressive models for employment transitions. An overview of the data preparation and model specification is given below.

We begin by fitting a model ignoring the initial condition, which involves specifying a model for employment status at $t > 1$ (**employ**) conditional on employment status at $t - 1$ (**emplag**). We then extend the model by including an equation for employment status at $t = 1$.

Calculate lagged covariates

In a first-order regressive model, the outcome variable is y_t with y_{t-1} included as a covariate. We are therefore modelling transitions between $t - 1$ and t, which we might expect to be influenced by characteristics measured at $t - 1$ before any transition occurred. We therefore calculate lags of the outcome (**employ**) and other covariates. We consider the following covariates (in addition to lagged employment status): age, marital status, and employed part-time (=0 if not employed), marital status. We do not use lagged age as it increases by at most one category between $t - 1$ and t, but we could have done.

Model without the initial condition

We fit a random effects model for employment transitions using `xtlogit`. As the lagged outcome, **emplag**, is missing for the first occasion, the first record for each woman is dropped from the analysis sample.

```
xtset pid
xtlogit employ emplag age2-age8 marstlag2 marstlag3 ptlag, re
```

The parameter estimates are given below.

```
Random-effects logistic regression              Number of obs      =     31089
Group variable: pid                             Number of groups   =      1988

Random effects u_i ~ Gaussian                   Obs per group: min =         1
                                                               avg =      15.6
                                                               max =        43
```

```
                                                Wald chi2(11)      =   2011.53
Log likelihood  = -8845.8597                    Prob > chi2        =    0.0000

------------------------------------------------------------------------------
      employ |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      emplag |    2.42078   .0688065    35.18   0.000     2.285922    2.555638
        age2 |   .5745761    .143522     4.00   0.000     .2932782    .8558741
        age3 |    .780549   .1533502     5.09   0.000     .4799881     1.08111
        age4 |   .9893677   .1616479     6.12   0.000     .6725435    1.306192
        age5 |   1.344383   .1680994     8.00   0.000     1.014914    1.673852
        age6 |   1.663325   .1748987     9.51   0.000      1.32053    2.006121
        age7 |    1.51931   .1820902     8.34   0.000      1.16242    1.876201
        age8 |   .9152313   .1817886     5.03   0.000     .5589321     1.27153
   marstlag2 |  -.6074299   .0863344    -7.04   0.000    -.7766422   -.4382176
   marstlag3 |   .0545504   .1465648     0.37   0.710    -.2327113    .3418121
       ptlag |  -.2538185   .0849909    -2.99   0.003    -.4203976   -.0872394
       _cons |  -1.479799   .1595179    -9.28   0.000    -1.792448   -1.167149
-------------+----------------------------------------------------------------
    /lnsig2u |   2.126883   .0723744                     1.985032    2.268734
-------------+----------------------------------------------------------------
     sigma_u |   2.896322   .1048097                     2.698014    3.109205
         rho |   .7182982   .0146447                     .6887291    .7460938
------------------------------------------------------------------------------
Likelihood-ratio test of rho=0: chibar2(01) =  1816.07 Prob >= chibar2 = 0.000
```

The large, significant coefficient of **emplag** suggests strong state-dependence: a woman's probability of being employed at t depends strongly on whether she was employed at $t-1$. For a woman who was employed part-time at $t-1$, the effect of previous employment is reduced by -0.25. There is also a large amount of unobserved heterogeneity, as indicated by the large random effect standard deviation of 2.90.

Modelling the initial condition: Common random effect for the first and subsequent occasions

Next, we extend the model by specifying an equation for employment status at the first observation. To specify separate equations for employment at $t=1$ and >1 , we calculate two indicator variables: **r1** for $t=1$ and **r2** for $t>1$. These are then multiplied by the covariates to be included in each equation, rather like the two-state duration model where **r1** and **r2** were indicators for the employment and non-employment states.

We begin by including the same individual-specific random effect in both equations. This model can be fitted using `xtlogit`, excluding the constant because r1 and r2 act as constants for the two equations.

```
xtlogit employ r1 r1_age2-r1_age8 r1_mst2 r1_mst3 ///
       r2 r2_emplag r2_age2-r2_age8 r2_mst2 r2_mst3 r2_ptlag, ///
       nocons re
```

We then fit the same model using Sabre. Note that we specify adaptive quadrature (using `sabre, quadrature a`) with 12 mass points, as this is closest to the default estimation procedure of `xtlogit`. The results are not exactly the same, but close enough.

Parameter	Estimate	Std. Err.
r1	0.66972	0.17453
r1_age2	0.19656	0.22361
r1_age3	0.49891	0.25310
r1_age4	0.89823	0.29074
r1_age5	1.4106	0.32147
r1_age6	1.3976	0.37357
r1_age7	0.97827	0.39946
r1_age8	-0.15197	0.50385
r1_mst2	-1.2206	0.18881
r1_mst3	0.20181	0.47434
r2	-1.3968	0.16353
r2_emplag	2.1222	0.63121E-01
r2_age2	0.68745	0.14247
r2_age3	0.92621	0.15223
r2_age4	1.1440	0.16083
r2_age5	1.5239	0.16773
r2_age6	1.8766	0.17508
r2_age7	1.7317	0.18305
r2_age8	1.0903	0.18363
r2_mst2lag	-0.71666	0.85393E-01
r2_mst3lag	-0.53002E-01	0.14355
r2_ptlag	-0.10046	0.82100E-01
scale	3.3634	0.11466

Compared to the model without the initial condition, the estimated of lagged employment status has reduced from 2.42 to 2.12. The standard deviation of the random effect has increased from 2.90 to 3.36. So we find weaker evidence for state-dependence and stronger evidence for unobserved heterogeneity.

Modelling the initial condition: Allowing different random effect variances for the first and subsequent occasions

The previous model includes the same random effect for $t = 1$ and for $t > 1$, which implies that the between-individual residual variance is the same for the first occasion and for subsequent occasions. We can allow the random effect variance to differ for the first occasion and for subsequent occasions with the following options:

```
sabre, rvar r       [r is the index distinguishing the 1st and subsequent occasion]
sabre, nvar 10      [specifies 2 equations with 10 predictors in the first]
sabre, depend y     [fits a different scale parameter (st. dev.) for each equation]
```

Note that this is equivalent to fitting a common random effect, but with a random effect loading λ for $t = 1$.

Parameter	Estimate	Std. Err.
r1	0.75485	0.21395
r1_age2	0.34668	0.26404
r1_age3	0.66150	0.29879
r1_age4	1.1090	0.34667
r1_age5	1.7853	0.39036
r1_age6	1.7178	0.44947
r1_age7	1.3762	0.48951
r1_age8	0.10329	0.61350
r1_mst2	-1.5449	0.23702
r1_mst3	0.25643	0.55072
r2	-1.4295	0.16029
r2_emplag	2.1524	0.62900E-01
r2_age2	0.70602	0.14095
r2_age3	0.93750	0.15004
r2_age4	1.1433	0.15826
r2_age5	1.5083	0.16500
r2_age6	1.8479	0.17217
r2_age7	1.6998	0.17986
r2_age8	1.0699	0.18012
r2_mst2lag	-0.68193	0.83013E-01
r2_mst3lag	-0.53982E-01	0.13988
r2_ptlag	-0.10390	0.80570E-01
scale1	4.6528	0.34956
scale2	3.2120	0.11324

There is more residual variation in employment status at $t = 1$ (random effect SD = 4.65) than at $t > 1$ (SD=3.21). However, allowing for the differential variation has little impact on the coefficient of lagged employment status (2.12 versus 2.15).

Practical 4: Models for Competing Risks

We return to the analysis of transitions out of non-employment, but distinguish between full-time and part-time employment using a competing risks model. We will fit a bivariate model with two binary responses: one indicating entry into full-time employment (treating part-time as censored) and a second for entry into part-time employment (treating full-time as censored). These responses are stacked into a single response **y** with a response index **r**.

The analysis is based on the 1399 women who have had a non-employment episode. These women contribute a total of 2284 episodes, of which 775 end in full-time work and 709 in part-time work. There are 15,297 person-year records.

4.1 Specifying a bivariate competing risks model in Sabre

The file **competing_risks.do** contains Stata and Sabre commands for preparing the data for a bivariate competing risks analysis, reading the data into Sabre and fitting a random effects competing risks model.

As for the two-state analysis of Practical 3, we begin with some data manipulation in Stata. This involves the following steps:

- select non-employment spells
- create dummy variables for covariates **tgp** and **ageg8** (taking the first category as the reference)
- set up the bivariate structure, with 2 records per year, the bivariate response **y** and response index **r** (coded 1 for full-time and 2 for part-time)
- create dummy variables for response type (**r1** for full-time and **r2** for part-time)
- interactions between each of **r1** and **r2** and duration (**tgp**) and age (**agegp**)

Variables with prefix **r1_** will be covariates in the transitions into full-time employment equation while those with prefix **r2_** will be covariates in the transitions into part-time employment equation.

As for the two-state analysis, the models take a few minutes to estimate so run the do-file and, while you are waiting, read the following descriptions of what it does.

```
use bhps, clear
sort pid spell t

* Select non-employment spells
keep if employ==0

* Create dummy variables for all categorical variables (taking 1st category as reference
in each case)

local i = 2
while `i' <=10 {
  gen tgp`i' = tgp==`i'
  local i = `i' + 1
}
local i = 2
while `i' <=8 {
  gen age`i' = ageg8==`i'
  local i = `i' + 1
}

* Create bivariate data structure for Sabre
expand 2
sort pid spell t
egen r=seq(), from(1) to(2) block(1)

gen y=0
replace y=1 if r==1 & event2==1
replace y=1 if r==2 & event2==2

*Create dummies for response types
gen r1 = r==1
gen r2 = r==2

gen r1_t2=r1*tgp2
gen r1_t3=r1*tgp3
gen r1_t4=r1*tgp4
gen r1_t5=r1*tgp5
gen r1_t6=r1*tgp6
gen r1_t7=r1*tgp7
gen r1_t8=r1*tgp8
gen r1_t9=r1*tgp9
gen r1_t10=r1*tgp10
gen r1_age2=r1*age2
gen r1_age3=r1*age3
gen r1_age4=r1*age4
gen r1_age5=r1*age5
gen r1_age6=r1*age6
gen r1_age7=r1*age7
gen r1_age8=r1*age8

gen r2_t2=r2*tgp2
gen r2_t3=r2*tgp3
gen r2_t4=r2*tgp4
gen r2_t5=r2*tgp5
gen r2_t6=r2*tgp6
gen r2_t7=r2*tgp7
gen r2_t8=r2*tgp8
gen r2_t9=r2*tgp9
gen r2_t10=r2*tgp10
gen r2_age2=r2*age2
gen r2_age3=r2*age3
gen r2_age4=r2*age4
gen r2_age5=r2*age5
```

```
gen r2_age6=r2*age6
gen r2_age7=r2*age7
gen r2_age8=r2*age8

compress
```

To see the bivariate structure :

```
. list pid spell t r event2 y in 201/212

     +---------------------------------------+
     |      pid   spell   t   r   event2   y |
     |---------------------------------------|
201. | 10188533       2   1   1        0   0 |
202. | 10188533       2   1   2        0   0 |
203. | 10188533       4   1   1        1   1 |
204. | 10188533       4   1   2        1   0 |
205. | 10192972       2   1   1        0   0 |
     |---------------------------------------|
206. | 10192972       2   1   2        0   0 |
207. | 10192972       2   2   1        2   0 |
208. | 10192972       2   2   2        2   1 |
209. | 10192972       4   1   1        0   0 |
210. | 10192972       4   1   2        0   0 |
     |---------------------------------------|
211. | 10192972       4   2   1        2   0 |
212. | 10192972       4   2   2        2   1 |
     +---------------------------------------+
```

The first individual (PID=10188533) has two non-employment spells. The first is censored at **t**=1 while the second ends in full-time employment at **t**=1 (as indicated by **event2**=1 and **y**=1 for **r**=1). The second individual (PID=10192972) also has two spells, both ending in part-time employment at **t**=2.

We are now ready to read the data into Sabre, and to specify and fit the model. Notice that the model specification is the same as for the two-state model.

```
sabre, data pid r r1 r2 y ///
   r1_t2 r1_t3 r1_t4 r1_t5 r1_t6 r1_t7 r1_t8 r1_t9 r1_t10 ///
   r1_age2 r1_age3 r1_age4 r1_age5 r1_age6 r1_age7 r1_age8 ///
   r2_t2 r2_t3 r2_t4 r2_t5 r2_t6 r2_t7 r2_t8 r2_t9 r2_t10 ///
   r2_age2 r2_age3 r2_age4 r2_age5 r2_age6 r2_age7 r2_age8
sabre pid r r1 r2 y ///
   r1_t2 r1_t3 r1_t4 r1_t5 r1_t6 r1_t7 r1_t8 r1_t9 r1_t10 ///
   r1_age2 r1_age3 r1_age4 r1_age5 r1_age6 r1_age7 r1_age8 ///
   r2_t2 r2_t3 r2_t4 r2_t5 r2_t6 r2_t7 r2_t8 r2_t9 r2_t10 ///
   r2_age2 r2_age3 r2_age4 r2_age5 r2_age6 r2_age7 r2_age8, read
#delimit cr

sabre, yvar y
sabre, model b
sabre, case pid
sabre, family first=b second=b
sabre, link first=l second=l
sabre, rvar r
sabre, constant first=r1 second=r2

sabre, nvar 17
```

```
*Fit random effects model

sabre, fit r1 r1_t2 r1_t3 r1_t4 r1_t5 r1_t6 r1_t7 r1_t8 r1_t9 r1_t10 ///
   r1_age2 r1_age3 r1_age4 r1_age5 r1_age6 r1_age7 r1_age8 ///
   r2 r2_t2 r2_t3 r2_t4 r2_t5 r2_t6 r2_t7 r2_t8 r2_t9 r2_t10 ///
   r2_age2 r2_age3 r2_age4 r2_age5 r2_age6 r2_age7 r2_age8
sabre, dis m
sabre, dis e
```

4.2 Interpretation of a simple model

The parameter estimates are given below.

Parameter	Estimate	Std. Err.
r1	-2.1675	0.20494
r1_t2	-1.0089	0.13233
r1_t3	-1.3486	0.17078
r1_t4	-1.8404	0.22475
r1_t5	-1.9230	0.25362
r1_t6	-1.7314	0.25818
r1_t7	-1.9246	0.30123
r1_t8	-2.6515	0.43410
r1_t9	-2.7186	0.47303
r1_t10	-3.2384	0.24483
r1_age2	0.21961	0.19483
r1_age3	0.21626	0.20485
r1_age4	0.93152E-01	0.21637
r1_age5	0.44412	0.22184
r1_age6	0.54894	0.23953
r1_age7	0.40449	0.24991
r1_age8	-0.73728	0.26942
r2	-3.6024	0.27078
r2_t2	-0.31182	0.12474
r2_t3	-0.91857	0.16729
r2_t4	-1.2864	0.20724
r2_t5	-1.2671	0.21841
r2_t6	-1.3271	0.23784
r2_t7	-1.3957	0.25835
r2_t8	-1.4434	0.27902
r2_t9	-1.6151	0.31287
r2_t10	-2.3884	0.20484
r2_age2	0.50005	0.26266
r2_age3	1.2525	0.25557
r2_age4	1.7061	0.25897
r2_age5	1.7365	0.26648
r2_age6	1.6373	0.28018
r2_age7	1.2801	0.29286
r2_age8	0.41669	0.29916
scale1	1.4820	0.12967
scale2	1.1911	0.10977
corr	-0.12359	0.98723E-01

We find that the probability of entering either type of employment decreases with the duration non-employed, but especially for full-time employment (from the estimates for **r1_t2** to **r1_t10**). Turning to the effects of age, we find that the probability of entering full-time employment increases with age up to 44 years (**ageg**=6) with a dip in the early 30s (**ageg**=4), but then decreases with the lowest probability for 50-59

years (**ageg**=8). For part-time employment, there is a sharper increase up to age 39 (**ageg**=5) and then a decrease.

4.3 Further exercises

Modify the Stata do-file **prac4.do** to include the following additional covariates in *both* equations:

marstat, **birth**, **nchildy** and **nchildo**

Interpret the full model.

[Note: If you also include **everjob, ljobclass2**, **ljobclass3** and **lptime** you will find that the random effect correlation is estimated as -1. These covariates are all indicators of a woman's employment history and their addition leads to the model being unidentified. We do not have sufficient information to be able to estimate this model which is likely to be due to the small number of women who have more than 1 transition of each type. The situation may improve if we analysed the full BHPS sample (over 8000 women).]

4.4 Other software

The model fitted in Section 4.1 can also be estimated within Stata using the `xtmelogit` command and using MLwiN via the `runmlwin` command. Code to do this is provided in **prac4_xtmelogit.do** (takes around 2 hours to run) and in **prac4_mlwin.do** (takes around 20 minutes).

Practical 5: Multiprocess Modelling

5.1. Modelling Two Correlated Processes in Sabre

We will begin by analysing transitions from non-employment (**employ**=0) jointly with fertility. This involves specifying two equations: one for the log-odds of moving out of non-employment (indicated by **event** for **employ**=0) and another for the log-odds of a birth (indicated by **birth** for both non-employment and employment episodes). Each equation includes a woman-level random effect and, as in the two-state and competing risks models, the equations are linked by allowing for a correlation between these random effects.

The analysis is based on 1994 women, 1399 of whom have at least one non-employment episode. Models for two correlated processes can be fitted in Sabre as bivariate response models (like two-state and competing risks models). The first step of the analysis is to create the bivariate data structure which involves stacking **event** (for **employ**=0) and **birth** into a single response which we will call **y**. The response type is then indicated by **r** (which we will code 1 for an employment response and 2 for a birth response). There are 15,297 person-year records for non-employment episodes and 33,083 person-year records for the birth histories (i.e. the full dataset of non-employment and employment episodes). Therefore the stacked dataset will have 48,380 records.

The file **prac5.do** contains Stata and Sabre commands for preparing the data for a multiprocess analysis, reading the data into Sabre and fitting a multiprocess model.

Run the whole do-file in one step (rather than copying-and-pasting sections to the Stata Command line). The models take a few minutes to estimate so run the do-file and, while you are waiting, read the following descriptions of what it does.

The following variables will be included in the employment and birth equations.

- *Transitions from non-employment*: **tgp**, **ageg8**, **everjob**, **ljobclass2**, **ljobclass3**, **lptime**, **marstat**, **birth**, **nchildy**, **nchildo**

- *Births*: **ageg8**, **employ**, **ptime**, **jobclass**, **marstat**, **nchildy**, **nchildo**

We begin with some data manipulation in Stata:

(i) Create dummies for the categorical predictors **tgp**, **ageg8**, **jobclass**, **marstat**, **nchildy** and **nchildo**

```
use bhps, clear
sort pid spell t

* Create dummy variables for all categorical variables (taking 1st category as
reference in each case)

local i = 2
while `i' <=10 {
  gen tgp`i' = tgp==`i'
  local i = `i' + 1
}
local i = 2
while `i' <=8 {
  gen age`i' = ageg8==`i'
  local i = `i' + 1
}
local i = 2
while `i' <=3 {
  gen jobclass`i' = jobclass==`i'
  replace jobclass`i'=0 if jobclass`i'==.
  gen marstat`i' = marstat==`i'
  local i = `i' + 1
}
local i = 1
while `i' <=2 {
  gen nchildy`i' = nchildy==`i'
  gen nchildo`i' = nchildo==`i'
  local i = `i' + 1
}
save temp, replace
```

This file is saved as **temp.dta** and will be used for the fertility analysis (as it contains records for both non-employment and employment).

(ii) Select non-employment episodes and create the bivariate data structure

We next select non-employment episodes and create a response index (**r**) which equals 1 for these records. We also define the bivariate response **y** which equals **event** for the employment process. This dataset is stacked on top of the dataset for the fertility analysis (**temp.dta**) using `append`, and **r** and **y** are filled in for the birth response. The final step, before reading the data into Sabre, is to create dummies for **r** and interact them with the explanatory variables to be included in the employment and birth equations (with prefix **r1** and **r2** respectively). Note that age categories 6, 7 and 8 are combined in the birth equation so that the final category is 40+ years.

```
*create bivariate structure for exits from non-employment and births

*first select non-employment episodes and code r=1 for employment transitions
keep if employ==0
gen r=1
gen y=event

*now append records for non-employment and employment episodes as we want to
model births during both
append using temp

replace r=2 if r==.
replace y=birth if r==2

sort pid t r

*Create dummies for employment and birth responses
*Create response index (1=employment, 2=birth)
gen r1 = r==1
gen r2 = r==2

gen r1_t2=r1*tgp2
gen r1_t3=r1*tgp3
gen r1_t4=r1*tgp4


etc. ……
```

Having read the data into Sabre, we specify the multiprocess model as follows:

```
sabre, yvar y
sabre, model b
sabre, case pid
sabre, family first=b second=b
sabre, link first=l second=l
sabre, rvar r
sabre, constant first=r1 second=r2
sabre, nvar 28
```

The setup is exactly the same as for the two-state and competing risks models, i.e. a bivariate logit model for the response **y** indexed by **r**. The maximum number of variables in either equation is 28 (the employment equation).

Run **prac5.do** to set up the data and fit the model. The output is given below:

```
Parameter              Estimate         Std. Err.
_____________________________________________________
r1                      -2.2427          0.16596
r1_t2                  -0.63474          0.10472
r1_t3                  -0.90959          0.13514
r1_t4                   -1.1985          0.16929
```

```
r1_t5                -1.0567          0.18490
r1_t6               -0.89786          0.19629
r1_t7               -0.97147          0.21634
r1_t8                -1.1949          0.25004
r1_t9                -1.2899          0.27521
r1_t10               -1.7450          0.17693
r1_age2              0.18618          0.16901
r1_age3              0.32321          0.17657
r1_age4              0.19732          0.18628
r1_age5           0.77399E-01         0.19539
r1_age6             -0.14595          0.20674
r1_age7             -0.40602          0.21615
r1_age8              -1.5449          0.22379
r1_ejob               2.8937          0.15165
r1_lpt              -0.43433          0.10113
r1_ljclass2         -0.37151          0.12931
r1_ljclass3         -0.64051          0.13894
r1_mstat2           -0.11410          0.93774E-01
r1_mstat3            0.36128          0.16081
r1_bir               -1.0118          0.13951
r1_nkidy1           -0.34529          0.10842
r1_nkidy2           -0.60340          0.17063
r1_nkido1            0.17890          0.12124
r1_nkido2            0.27299          0.13280
r2                   -2.2126          0.13888
r2_age2              0.40295          0.14406
r2_age3              0.50240          0.15373
r2_age4              0.24069          0.16716
r2_age5             -0.50539          0.18434
r2_age678            -3.4133          0.23746
r2_emp               -2.2446          0.13554
r2_pt                0.43439          0.14626
r2_jc2               0.23195          0.15249
r2_jc3               0.59980          0.18514
r2_mst2               1.6898          0.95765E-01
r2_mst3              0.85662          0.15997
r2_nkidy1           -0.52233          0.85387E-01
r2_nkidy2            -1.8974          0.13548
r2_nkido1            -1.0274          0.10193
r2_nkido2            -2.2089          0.13656
scale1               0.69797          0.90544E-01
scale2                1.1129          0.64518E-01
corr                 0.33724          0.11468
```

To test the significance of the residual correlation, we compare this model to a model that assumes a zero correlation. To fit this model we add the command **`sabre, corr n`** before **`sabre, fit`**. This model is also fitted in **prac5.do**.

The log-likelihood values for the two models are -7882.6 for the multiprocess model and -7886.6 for the `single process' model that assumes a zero correlation. The likelihood ratio test statistic is twice the differences between these values = 8 which we compare with a chi-squared distribution on 1 d.f. to get a p-value of 0.005.[3] We therefore conclude that there is strong evidence of a residual correlation.

[3] To calculate the p-value for a test statistic x on n d.f. in Stata, type `di chi2tail(n,x)`

The estimated correlation is 0.337. Women whose unobserved characteristics place them at a high chance of moving out of non-employment also tend to have a high probability of a birth. Or, put another way, women with short non-employment spells tend also to have short birth intervals.

We now compare the estimated effects of an imminent birth and the presence of children on transitions out of non-employment for the two models:

Single-process model (ρ = 0)

```
r1_bir              -0.84170          0.12512
r1_nkidy1           -0.21184          0.97149E-01
r1_nkidy2           -0.34573          0.14306
r1_nkido1            0.25110          0.11785
r1_nkido2            0.44617          0.11653
```

Multiprocess model (ρ ≠ 0)

```
r1_bir               -1.0118          0.13951
r1_nkidy1           -0.34529          0.10842
r1_nkidy2           -0.60340          0.17063
r1_nkido1            0.17890          0.12124
r1_nkido2            0.27299          0.13280
```

We find that negative effects have become stronger while positive effects have become weaker. Women with a high probability of leaving non-employment are selected into **birth**=1 and categories 1 and 2 of **nchildy** and **nchildo**, inflating the probability of a transition in these categories.

5.2 Exercise: Multiprocess Analysis of Transitions from Employment and Births

Now fit a multiprocess model for transitions from *employment* (**employ**=1) jointly with fertility. This will involve modifications to **prac5.do** .

(Hint: You will need to replace the explanatory variables relating to previous work experience with variables that relate to the current job.)

- Carry out a likelihood ratio test of the null hypothesis that the residual correlation between transitions from employment and fertility is zero, i.e. H_0: ρ = 0
- Interpret the estimated correlation

- Compare estimates of the effects of the fertility indicators (**birth**, **nchildy**, **nchildo**) on transitions out of employment for the multiprocess model (with $\rho \neq 0$) and the single-process model (with $\rho = 0$). Do the estimates change in the direction you would have expected?

5.3 Modelling Three Correlated Processes

It is possible to estimate models for up to three correlated processes in Sabre. For three processes, we would stack the three binary responses into a single trivariate response **y**, define a response index **r** with three categories, and interact dummies for **r** with the explanatory variables to appear in each equation. The specification of a trivariate logit model is given below. (Note that the number of variables in the first and second equations needs to be specified; Sabre automatically deduces the number in the third equation.)

```
sabre, yvar y
sabre, model t
sabre, case pid
sabre, family first=b second=b third=b
sabre, link first=l second=l third=l
sabre, rvar r
sabre, constant first=r1 second=r2 third=r3
sabre, nvar first=n1 second=n2
```

However, there is an upper limit to the number of variables that can be included in a model using Sabre-in-Stata (although not in standalone Sabre). For this reason, it is not possible to combine full models for the three processes explored above – employment transitions, non-employment transitions and births – in a single model in Sabre. An example of this model, estimated using `xtmelogit`, is provided in **prac5_3process_xtmelogit.do**. Note this model took around 72 hours to fit!