

**SC706: Longitudinal Data Analysis**  
**Instructor: Natasha Sarkisian**

**Additional Notes on Multi-record Data Management for Event History Analysis**

How can we use data that is only available for a couple of years, say, only 1982 and 1986?

We can either use these data as a time-invariant variable – for instance, take the 1982 variable and use it as time-invariant (in the reshape command, do not specify the stem for that variable – then it will be applied to all lines of data).

We could also use these data as time-variant – we could, for example, apply 1982 data to years 1983-85, and then apply 1986 data to years 1987 and later. In the wide format, we can do it by generating variables for each year and making the variables for 1983-85 equal to 1982 variable and then make all the variables starting with 1987 equal to 1986 variable. In the long format, we would fill in the missing values (for those years where we do not have data) with data from previous lines (so that the data from 1982 “trickle down” until the next time point where data are available, which would be 1986). Let’s get to long format:

```
. set memory 100m
(102400k)

. use http://www.sarkisian.net/sc706/marriage.dta

. gen id = _n

. reshape long mar fexp educ interv newage emp enrol, i(id) j(year)
(note: j = 79 81 82 83 84 85 86 87 88 89 90 91 92 93 94)
(note: mar79 not found)
(note: fexp79 not found)
(note: educ79 not found)
(note: newage79 not found)
(note: emp79 not found)
(note: enrol79 not found)
(note: mar81 not found)
(note: fexp81 not found)
(note: educ81 not found)
(note: newage81 not found)
(note: emp81 not found)
(note: enrol81 not found)
```

| Data                           | wide | -> | long   |
|--------------------------------|------|----|--------|
| Number of obs.                 | 1204 | -> | 18060  |
| Number of variables            | 98   | -> | 13     |
| j variable (15 values)         |      | -> | year   |
| xij variables:                 |      |    |        |
| mar79 mar81 ... mar94          |      | -> | mar    |
| fexp79 fexp81 ... fexp94       |      | -> | fexp   |
| educ79 educ81 ... educ94       |      | -> | educ   |
| interv79 interv81 ... interv94 |      | -> | interv |
| newage79 newage81 ... newage94 |      | -> | newage |
| emp79 emp81 ... emp94          |      | -> | emp    |
| enrol79 enrol81 ... enrol94    |      | -> | enrol  |

```
. gen enroln=enrol
(2408 missing values generated)

. bysort id: replace enroln=enroln[_n-1] if enroln==.
(0 real changes made)
```

We would run this command multiple times until we get the message “0 real changes made” – that would mean all the replacements are made.

We can also use the same approach to filling in date of birth (in your multirecord data notes, we do that using egen command):

```
. gen dob=interv-newage
(4304 missing values generated)

. gen dob2=dob
(4304 missing values generated)

. bysort id: replace dob2=dob2[_n-1] if dob2==.
(1812 real changes made)

. bysort id: replace dob2=dob2[_n-1] if dob2==.
(0 real changes made)

. bysort id: replace dob2=dob2[_n+1] if dob2==.
(1204 real changes made)

. bysort id: replace dob2=dob2[_n+1] if dob2==.
(1196 real changes made)

. bysort id: replace dob2=dob2[_n+1] if dob2==.
(33 real changes made)

. bysort id: replace dob2=dob2[_n+1] if dob2==.
(14 real changes made)

. bysort id: replace dob2=dob2[_n+1] if dob2==.
(10 real changes made)

. bysort id: replace dob2=dob2[_n+1] if dob2==.
(10 real changes made)

. bysort id: replace dob2=dob2[_n+1] if dob2==.
(10 real changes made)

. bysort id: replace dob2=dob2[_n+1] if dob2==.
(7 real changes made)

. bysort id: replace dob2=dob2[_n+1] if dob2==.
(4 real changes made)

. bysort id: replace dob2=dob2[_n+1] if dob2==.
(3 real changes made)

. bysort id: replace dob2=dob2[_n+1] if dob2==.
(1 real change made)

. bysort id: replace dob2=dob2[_n+1] if dob2==.
(0 real changes made)
```

Now let's take the same data and set it up for discrete time analysis.

```
. use http://www.sarkisian.net/sc706/marriage.dta, clear
```

First let's generate time variable, starting with 1982 wave as time point 1.

```
. gen timeyear=.
(1204 missing values generated)

. replace timeyear=82 if mar82==1
(330 real changes made)

. replace timeyear=83 if mar83==1 & timeyear==.
(102 real changes made)

. replace timeyear=84 if mar84==1 & timeyear==.
(102 real changes made)
```

We could keep going this way, or we can do the same thing using a loop:

```
. drop timeyear

. gen timeyear=.
(1204 missing values generated)

. for num 82/94: replace timeyear=X if marX==1 & timeyear==.

-> replace timeyear=82 if mar82==1 & timeyear==.
(330 real changes made)

-> replace timeyear=83 if mar83==1 & timeyear==.
(102 real changes made)

-> replace timeyear=84 if mar84==1 & timeyear==.
(102 real changes made)

-> replace timeyear=85 if mar85==1 & timeyear==.
(77 real changes made)

-> replace timeyear=86 if mar86==1 & timeyear==.
(110 real changes made)

-> replace timeyear=87 if mar87==1 & timeyear==.
(78 real changes made)

-> replace timeyear=88 if mar88==1 & timeyear==.
(58 real changes made)

-> replace timeyear=89 if mar89==1 & timeyear==.
(27 real changes made)

-> replace timeyear=90 if mar90==1 & timeyear==.
(44 real changes made)

-> replace timeyear=91 if mar91==1 & timeyear==.
(39 real changes made)

-> replace timeyear=92 if mar92==1 & timeyear==.
(20 real changes made)

-> replace timeyear=93 if mar93==1 & timeyear==.
(11 real changes made)
```

```
-> replace timeyear=94 if mar94==1 & timeyear==.
(17 real changes made)
```

So we filled in values of timeyear for everyone who has experienced the event (got married) during the observation period. But we also need to assign some value to those who did not get married. We could assign the last wave (94) to all of them, but that assumes that nobody dropped out from the study, and that is usually an unrealistic assumption. We have to decide how to handle missing data. For now, we'll decide to ignore it if someone is missing a value of marital status somewhere in the middle, but if someone drops out of the study, we would want to take care of that and make sure to assign the last year when that woman was observed rather than 94. For that purpose, we will create marmiss:

```
. gen marmiss=.
(1204 missing values generated)

. replace marmiss=94 if mar94==.
(0 real changes made)

. replace marmiss=93 if mar93==. & marmiss~=.
(0 real changes made)
```

Or once again we'll do a loop:

```
. drop marmiss

. gen marmiss=.
(1204 missing values generated)

. for num 94/82: replace marmiss=X if marX==. & marmiss~=.

-> replace marmiss=94 if mar94==. & marmiss~=.
(0 real changes made)

-> replace marmiss=93 if mar93==. & marmiss~=.
(0 real changes made)

-> replace marmiss=92 if mar92==. & marmiss~=.
(0 real changes made)

-> replace marmiss=91 if mar91==. & marmiss~=.
(0 real changes made)

-> replace marmiss=90 if mar90==. & marmiss~=.
(0 real changes made)

-> replace marmiss=89 if mar89==. & marmiss~=.
(0 real changes made)

-> replace marmiss=88 if mar88==. & marmiss~=.
(0 real changes made)

-> replace marmiss=87 if mar87==. & marmiss~=.
(0 real changes made)

-> replace marmiss=86 if mar86==. & marmiss~=.
(0 real changes made)

-> replace marmiss=85 if mar85==. & marmiss~=.
```

```

(0 real changes made)

-> replace marmiss=84 if mar84==. & marmiss~=.
(0 real changes made)

-> replace marmiss=83 if mar83==. & marmiss~=.
(0 real changes made)

-> replace marmiss=82 if mar82==. & marmiss~=.
(0 real changes made)

```

Now let's take care of timeyear:

```

. replace timeyear=94 if timeyear==.
(189 real changes made)

. replace timeyear=marmiss-1 if marmiss~=.
(0 real changes made)

. gen time=timeyear-81

```

Time variable is ready; need an outcome variable:

```

. gen outcome=0

. for num 82/94: replace outcome=1 if marX==1

-> replace outcome=1 if mar82==1
(330 real changes made)

-> replace outcome=1 if mar83==1
(102 real changes made)

-> replace outcome=1 if mar84==1
(102 real changes made)

-> replace outcome=1 if mar85==1
(77 real changes made)

-> replace outcome=1 if mar86==1
(110 real changes made)

-> replace outcome=1 if mar87==1
(78 real changes made)

-> replace outcome=1 if mar88==1
(58 real changes made)

-> replace outcome=1 if mar89==1
(27 real changes made)

-> replace outcome=1 if mar90==1
(44 real changes made)

-> replace outcome=1 if mar91==1
(39 real changes made)

-> replace outcome=1 if mar92==1
(20 real changes made)

-> replace outcome=1 if mar93==1
(11 real changes made)

```

```
-> replace outcome=1 if mar94==1
(17 real changes made)
```

```
. tab outcome
```

| outcome | Freq. | Percent | Cum.   |
|---------|-------|---------|--------|
| 0       | 189   | 15.70   | 15.70  |
| 1       | 1,015 | 84.30   | 100.00 |
| Total   | 1,204 | 100.00  |        |

We are ready to reshape these data:

```
. gen id=_n
```

```
. reshape long mar fexp educ interv newage emp enrol , i(id) j(year)
(note: j = 79 81 82 83 84 85 86 87 88 89 90 91 92 93 94)
(note: mar79 not found)
(note: fexp79 not found)
(note: educ79 not found)
(note: newage79 not found)
(note: emp79 not found)
(note: enrol79 not found)
(note: mar81 not found)
(note: fexp81 not found)
(note: educ81 not found)
(note: newage81 not found)
(note: emp81 not found)
(note: enrol81 not found)
```

```
Data                                wide  ->  long
-----
Number of obs.                       1204  ->  18060
Number of variables                   102   ->   17
j variable (15 values)                ->  year
xij variables:
      mar79 mar81 ... mar94  ->  mar
      fexp79 fexp81 ... fexp94 ->  fexp
      educ79 educ81 ... educ94 ->  educ
      interv79 interv81 ... interv94 ->  interv
      newage79 newage81 ... newage94 ->  newage
      emp79 emp81 ... emp94 ->  emp
      enrol79 enrol81 ... enrol94 ->  enrol
```

Let's drop years 79 and 81 because we are not using them so those extra lines would mess up our count:

```
. drop if year<82
(2408 observations deleted)
```

```
. bysort id: drop if _n>time
(9048 observations deleted)
```

We will also drop year 82 because we are only starting to observe after that, so we want to focus on those not yet married in 1982.

```
. drop if year==82
(1204 observations deleted)
```

Now we are ready to run discrete time event history analysis models; for example:

```
. xi: logit mar black hispanic emp educ, i.year cluster(id)
i.year      _Iyear_83-94      (naturally coded; _Iyear_83 omitted)
option _Iyear_ not allowed
r(198);
```

```
. xi: logit mar black hispanic emp educ i.year, cluster(id)
i.year      _Iyear_83-94      (naturally coded; _Iyear_83 omitted)
```

```
Iteration 0: log pseudolikelihood = -2053.9337
Iteration 1: log pseudolikelihood = -1972.7348
Iteration 2: log pseudolikelihood = -1968.3636
Iteration 3: log pseudolikelihood = -1968.348
Iteration 4: log pseudolikelihood = -1968.348
```

```
Logistic regression                                Number of obs =      5400
                                                    Wald chi2(15) =     143.97
                                                    Prob > chi2      =      0.0000
Log pseudolikelihood = -1968.348                Pseudo R2       =      0.0417
```

(Std. Err. adjusted for 874 clusters in id)

| mar       | Coef.     | Robust Std. Err. | z     | P> z  | [95% Conf. Interval] |           |
|-----------|-----------|------------------|-------|-------|----------------------|-----------|
| black     | -1.00351  | .1081466         | -9.28 | 0.000 | -1.215474            | -.7915467 |
| hispanic  | -.2362471 | .1118099         | -2.11 | 0.035 | -.4553905            | -.0171037 |
| emp       | -.4706133 | .1131818         | -4.16 | 0.000 | -.6924455            | -.248781  |
| educ      | .0308611  | .0207387         | 1.49  | 0.137 | -.0097861            | .0715082  |
| _Iyear_84 | .1452322  | .1517073         | 0.96  | 0.338 | -.1521086            | .4425731  |
| _Iyear_85 | -.011192  | .1638714         | -0.07 | 0.946 | -.332374             | .30999    |
| _Iyear_86 | .5686604  | .1506331         | 3.78  | 0.000 | .2734248             | .8638959  |
| _Iyear_87 | .4457557  | .1652025         | 2.70  | 0.007 | .1219648             | .7695467  |
| _Iyear_88 | .3040639  | .1789773         | 1.70  | 0.089 | -.0467252            | .654853   |
| _Iyear_89 | -.3633764 | .2302113         | -1.58 | 0.114 | -.8145822            | .0878293  |
| _Iyear_90 | .2888447  | .1968221         | 1.47  | 0.142 | -.0969195            | .6746089  |
| _Iyear_91 | .2749053  | .193626          | 1.42  | 0.156 | -.1045947            | .6544052  |
| _Iyear_92 | -.1913355 | .259459          | -0.74 | 0.461 | -.6998658            | .3171949  |
| _Iyear_93 | -.7268348 | .331836          | -2.19 | 0.028 | -1.377221            | -.0764481 |
| _Iyear_94 | -.1874204 | .2780665         | -0.67 | 0.500 | -.7324207            | .3575798  |
| _cons     | -1.76345  | .2803201         | -6.29 | 0.000 | -2.312868            | -1.214033 |