DEM 7223 - Event History Analysis - Multistate Models

Corey S. Sparks, PhD

10 November, 2022

Multi-state models

- We have discussed these in passing several times e.g. multinomial logistic regression
- Multi-state models are a general form, because they not only allow multiple types of transitions, but these transitions can occur multiple times
- Think of the single>married>divorced>dead model

This example will illustrate how to fit a multistate hazard model using the multinomial logit model.

The outcome for the example is whether a family experiences a transition between poverty states between waves 1 and 5 of the data. The data from the ECLS-K.

```
library(survival, quietly = T)
library(car, quietly = T)
library(survey, quietly = T)
library(tidyverse, quietly = T)
library(VGAM, quietly = T)
```

First we load our data

```
eclskk5<-readRDS("C:/Users/ozd504/OneDrive - University of Texas at San Antonio/classes/de
names(eclskk5)<-tolower(names(eclskk5))
#get out only the variables I'm going to use for this example
myvars<-c( "childid","x_chsex_r", "x_raceth_r", "x1kage_r","x4age",
           "x5age", "x6age", "x7age", "x2povty","x4povty_i", "x6povty_i",
           "x8povty_i","x12par1ed_i", "s2_id","w6c6p_6psu",
           "w6c6p_6str", "w6c6p_20")
```

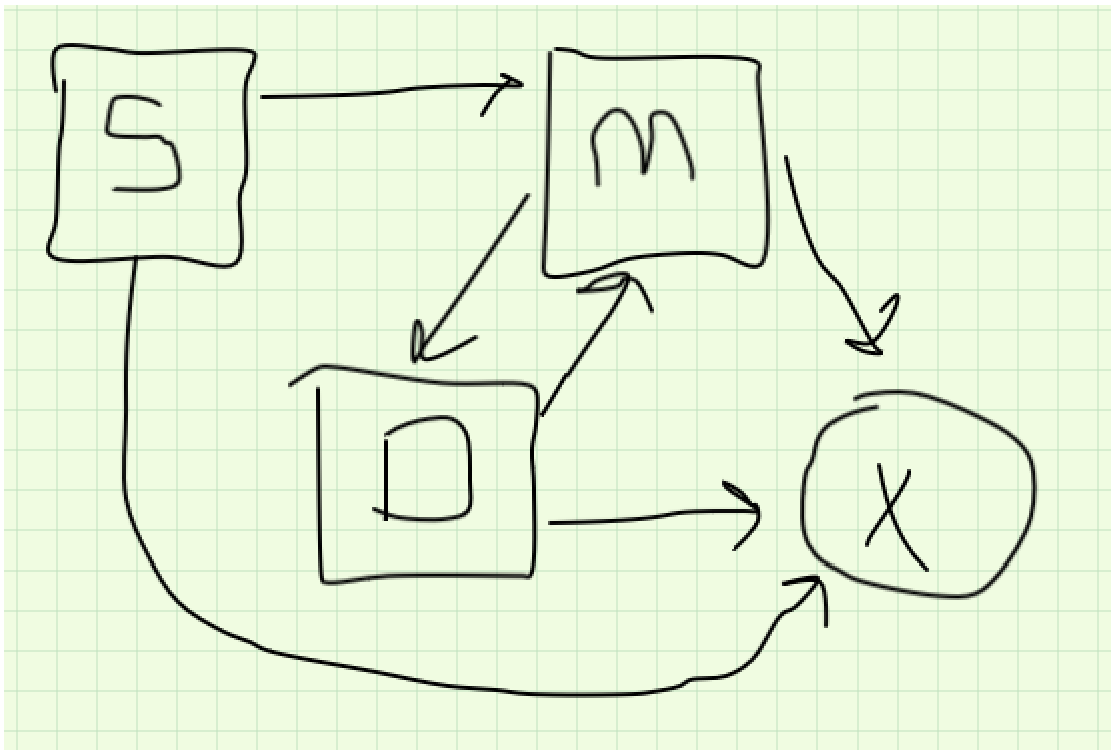


Figure 1: SMD Model

```
eclskk5<-eclskk5[,myvars]


# time varying variables
eclskk5$age_1<-ifelse(eclskk5$x1kage_r==-9, NA, eclskk5$x1kage_r/12)
eclskk5$age_2<-ifelse(eclskk5$x4age==-9, NA, eclskk5$x4age/12)
#for the later waves, the NCES group the ages into ranges of months,
#so 1= <105 months, 2=105 to 108 months.
#So, I fix the age at the midpoint of the interval they give,
#and make it into years by dividing by 12

eclskk5$age_3<-ifelse(eclskk5$x5age==-9, NA, eclskk5$x5age/12)

eclskk5$pov_1<-ifelse(eclskk5$x2povty==1,1,0)
eclskk5$pov_2<-ifelse(eclskk5$x4povty_i==1,1,0)
eclskk5$pov_3<-ifelse(eclskk5$x6povty_i==1,1,0)


#Time constant variables
#Recode race with white, non Hispanic as reference using dummy vars
eclskk5$race_rec<-Recode (eclskk5$x_raceth_r, recodes="1 = 'nhwhite';2='nhblack';3:4='hisp
eclskk5$male<-Recode(eclskk5$x_chsex_r, recodes="1=1; 2=0; -9=NA")
eclskk5$mlths<-Recode(eclskk5$x12par1ed_i, recodes = "1:2=1; 3:9=0; else = NA")
eclskk5$mgths<-Recode(eclskk5$x12par1ed_i, recodes = "1:3=0; 4:9=1; else =NA")
```

Now, I need to form the transition variable, this is my event variable, and in this case it will be 1 if a child enters poverty between the first wave of the data and the third grade wave, and 0 otherwise.

NOTE I need to remove any children who are already in poverty age wave 1, because they are not at risk of experiencing **this particular** transition. Again, this is called forming the *risk set*

```
eclskk5<-eclskk5 %>% filter(is.na(pov_1)==F &
                  is.na(pov_2)==F &
                  is.na(pov_3)==F &
                  is.na(age_1)==F &
                  is.na(age_2)==F &
                  is.na(age_3)==F &
                    pov_1!=1)
```

Now we do the entire data set. To analyze data longitudinally, we need to reshape the data from the current "wide" format (repeated measures in columns) to a "long" format (repeated observations in rows).

The `pivot_longer()` function allows us to do this easily. It allows us to specify our repeated measures, time varying covariates as well as time-constant covariates.

Create states for poverty transitions

we also need to form the transition variable, this is my event variable, and in this case it will be categorical. It is categorical because we have the following state-space in our outcome.

```
e.long1 <- eclskk5 %>%
  #rename(wt = w4c4p_40,strata= w4c4p_4str, psu = w4c4p_4psu)%>%
  select(childid,male, race_rec, mlths, mgths,   #time constant
        age_1, age_2, age_3, #t-varying variables
         pov_1, pov_2, pov_3,
        w6c6p_6psu, w6c6p_6str, w6c6p_20)%>%
   pivot_longer(cols = c(-childid, -male, -race_rec, -mlths, -mgths,
                              -w6c6p_6psu, -w6c6p_6str, -w6c6p_20), #time constant variables go
               names_to  = c(".value", "wave"), #make wave variable and put t-v vars into
               names_sep = "_") %>% #all t-v variables have _ between name and time, like
  group_by(childid)%>%
  mutate(age_enter = round(age),
         age_exit = round(lead(age, 1, order_by=childid)))%>%
  mutate(nexpov = dplyr::lead(pov,n=1, order_by = childid))%>%
  ungroup()%>%
  mutate(povtran = factor(
    case_when(.$pov == 0 & .$nexpov ==0 ~ "stay_notinpov",
                               .$pov == 0 & .$nexpov ==1 ~ "into_pov",
                               .$pov == 1 & .$nexpov ==0 ~ "outof_pov",
                               .$pov == 1 & .$nexpov ==1 ~ "stay_inpov")))%>%
  filter(is.na(age_exit)==F)%>%
  filter(complete.cases(age_enter, age_exit, povtran,
                          mlths, mgths, race_rec,w6c6p_6psu, w6c6p_20,w6c6p_6str))

e.long1$povtran <- relevel(e.long1$povtran, ref = "stay_notinpov")


knitr::kable(head(e.long1[, c("childid", "pov","wave",  "povtran")], n=20))
```

childid	pov	wave	povtran
10000014	0	1	stay_notinpov

childid	pov	wave	povtran
10000014	0	2	stay_notinpov
10000020	0	1	stay_notinpov
10000020	0	2	stay_notinpov
10000022	0	1	stay_notinpov
10000022	0	2	stay_notinpov
10000029	0	1	stay_notinpov
10000029	0	2	stay_notinpov
10000034	0	1	stay_notinpov
10000034	0	2	into_pov
10000040	0	1	stay_notinpov
10000040	0	2	stay_notinpov
10000046	0	1	stay_notinpov
10000046	0	2	stay_notinpov
10000047	0	1	stay_notinpov
10000047	0	2	stay_notinpov
10000053	0	1	stay_notinpov
10000053	0	2	stay_notinpov
10000062	0	1	into_pov
10000062	1	2	stay_inpov

```r
table(e.long1$wave, e.long1$povtran)
```

```
   stay_notinpov into_pov outof_pov stay_inpov
 1          1896      146         0          0
 2          1821       75        51         95
```

Model for categorical outcome

Multinomial Model

We also commonly see a response variable with unordered categories for a response. Such as:

- A person is deciding whether to migrate to a set of possible new labor markets
- A person is deciding which of a set of possible child care arrangement to use
- A person is deciding which of a possible set of means by which to get to work
- The type of contraception a woman chooses

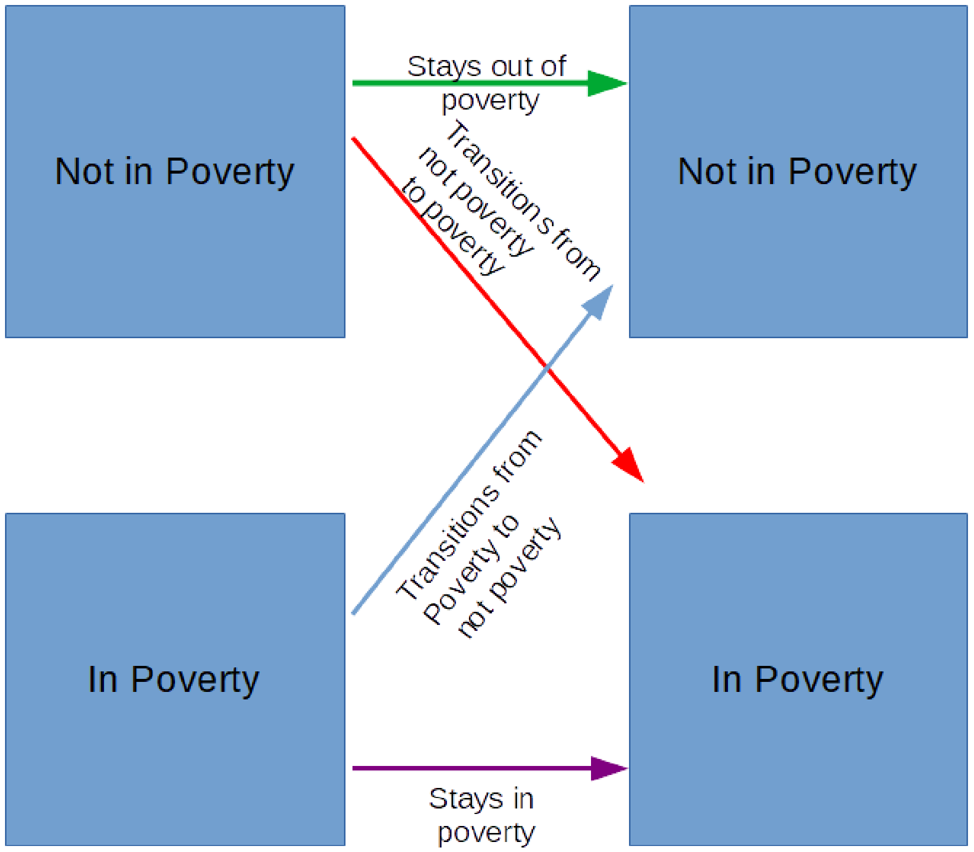


Figure 2: State space

These are example of what generally is called an alternative decision making process.

A distribution commonly used for this type of outcome is the multivariate extension of the binomial distribution, known as the multinomial distribution.

If you have an outcome with J *unordered* classes, then π_1, π_2, ..., π_J are the probabilities of observing the J classes. Remember, $\sum_J \pi_j = 1$.

If we observe y_1 outcomes in the first category and y_2 outcomes in the second and so on, the let:

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_4 \end{bmatrix}, with \sum_{j=1}^{J} y_i = n$$

In this case y follows the multinomial distribution:

$$f(y|n) = \frac{n!}{y_1! y_2! \cdots y_j!} \pi_1^{y1} \pi_2^{y2} \cdots \pi_J^{yJ}$$

When constructing a regression model for such a distribution, we choose one level of the outcome as the **reference level** and compare the probability of choosing other options to it. For example, if we modeled the probability of choosing option j to option 1 (as the reference level), the we would have the logistic regression model:

$$logit(\pi_j) = log\left(\frac{\pi_j}{\pi_1}\right) = x_j' \beta_j$$

This makes *J-1* equations, again with the reference category estimated by the complementary rule. We then have a total of *J-1* regression equations to interpret, where we compare the odds of being in each of the other *j-1* categories to the reference category. If we want to estimate the probability of being in each particular class of the outcome, for a response with a given set of x values, we can solve the equation for π_j as:

$$\pi_j = \frac{exp(x_j' \beta_j)}{1 + \sum_{j=2}^{J} exp(x_j' \beta_j)}$$

This is a much more complicated model than the proportional odds model, and we will have many more effects to interpret, owing to the J-1 separate equations we are estimating.

We will use the multinomial logit model for this outcome, as it is specified for modeling a categorical outcome. The `svyVGAM` package has the `svy_vglm` function to do this, as it accepts a full survey design object. Also, its summary functions do not return test statistics or p-values, so we will need to create those.

```r
options(survey.lonely.psu = "adjust")
des <- survey::svydesign(ids = ~w6c6p_6psu,
                         strata = ~w6c6p_6str,
                         weights = ~w6c6p_20,
                         nest=T,
                         data=e.long1)
library(svyVGAM)
library(gtsummary)
mfit<-svy_vglm(povtran ~ factor(age_enter)+mlths+mgths+race_rec,
           family=multinomial(refLevel = "stay_notinpov"),
           design = des)
mfit%>%
  tbl_regression(tidy_fun = broom.helpers::tidy_parameters)
```

```
x Unable to identify the list of variables.

This is usually due to an error calling `stats::model.frame(x)`or `stats::model.matrix(x)`.
It could be the case if that type of model does not implement these methods.
Rarely, this error may occur if the model object was created within
a functional programming framework (e.g. using `lappy()`, `purrr::map()`, etc.).


Table printed with `knitr::kable()`, not {gt}. Learn why at
https://www.danieldsjoberg.com/gtsummary/articles/rmarkdown.html
To suppress this message, include `message = FALSE` in code chunk header.
```

Characteristic	**Beta**	**95% CI**	**p-value**
(Intercept):1	-1.2	-1.6, -0.83	<0.001
(Intercept):2	-21	-22, -21	<0.001
(Intercept):3	-20	-21, -20	<0.001
factor(age_enter)6:1	-0.04	-0.41, 0.34	0.8
factor(age_enter)6:2	0.05	-0.07, 0.18	0.4
factor(age_enter)6:3	15	14, 17	<0.001
factor(age_enter)7:1	-0.65	-1.0, -0.26	0.001
factor(age_enter)7:2	18	18, 19	<0.001
factor(age_enter)7:3	19	19, 19	<0.001
factor(age_enter)8:1	-0.16	-0.74, 0.43	0.6
factor(age_enter)8:2	18	17, 19	<0.001

Characteristic	**Beta**	**95% CI**	**p-value**
factor(age_enter)8:3	18	18, 19	<0.001
factor(age_enter)9:1	1.7	1.0, 2.5	<0.001
factor(age_enter)9:2	21	20, 22	<0.001
factor(age_enter)9:3	43	42, 44	<0.001
mlths:1	0.48	0.01, 0.95	0.044
mlths:2	0.97	-0.02, 2.0	0.055
mlths:3	0.36	-0.33, 1.0	0.3
mgths:1	-1.3	-1.7, -0.84	<0.001
mgths:2	-0.75	-1.8, 0.35	0.2
mgths:3	-1.3	-1.8, -0.75	<0.001
race_recnhasian:1	-0.75	-1.5, 0.02	0.057
race_recnhasian:2	0.02	-1.4, 1.5	>0.9
race_recnhasian:3	-1.1	-2.0, -0.08	0.034
race_recnhblack:1	-0.01	-0.63, 0.61	>0.9
race_recnhblack:2	0.35	-0.72, 1.4	0.5
race_recnhblack:3	0.00	-0.90, 0.90	>0.9
race_recnhwhite:1	-0.99	-1.6, -0.42	<0.001
race_recnhwhite:2	-0.46	-1.4, 0.54	0.4
race_recnhwhite:3	-1.2	-1.9, -0.41	0.002
race_recother:1	-0.50	-1.0, 0.00	0.052
race_recother:2	0.53	-0.44, 1.5	0.3
race_recother:3	-0.91	-1.7, -0.11	0.026

This can be a lot to interpret, so we will also visualize the model output by estimating the transition probabilities themselves and graphing them.

Now we get estimates of the transition probabilities from the model to visualize them

```r
mfit2<-vglm(povtran ~ factor(age_enter)+mlths+mgths+race_rec,
            family=multinomial(refLevel = "stay_notinpov"),
            data=e.long1,
            weights = w6c6p_20/mean(w6c6p_20))

dat1<-expand.grid(age_enter=5:8,
                  mlths=c(0,1),
                  mgths=c(0,1),
                  race_rec=levels(e.long1$race_rec))

test1<-predict(mfit2, newdata=dat1, type="response")
```

```
dat1$p_staynotpov<-test1[,1]
dat1$p_intopov<-test1[,2]
dat1$p_outofpov<-test1[,3]
dat1$p_stayinpov<-test1[,4]

head(dat1)
```

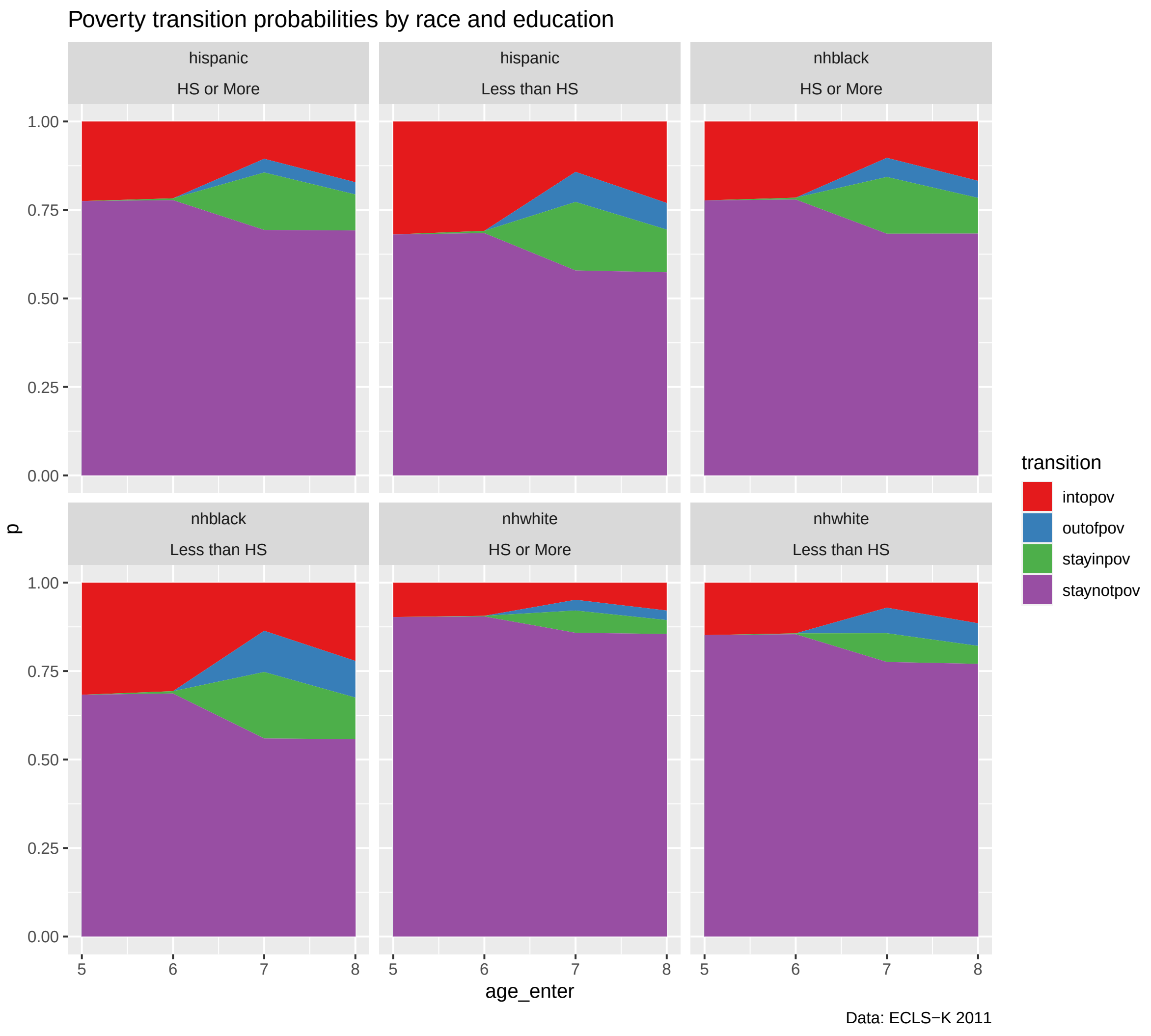
```
  age_enter mlths mgths race_rec p_staynotpov p_intopov   p_outofpov
1         5     0     0 hispanic    0.7750872 0.2249128 4.729574e-10
2         6     0     0 hispanic    0.7776836 0.2172020 5.004988e-10
3         7     0     0 hispanic    0.6934854 0.1054934 3.870104e-02
4         8     0     0 hispanic    0.6920131 0.1717439 3.447326e-02
5         5     1     0 hispanic    0.6806883 0.3193117 1.091738e-09
6         6     1     0 hispanic    0.6845090 0.3090602 1.157919e-09
   p_stayinpov
1 1.078697e-09
2 5.114339e-03
3 1.623202e-01
4 1.017697e-01
5 1.353303e-09
6 6.430779e-03
```

Reshape the data into long form for plotting

```
dat1.long <- dat1 %>%
  pivot_longer(cols = c(-age_enter, -mlths, -mgths, -race_rec),
               names_to  = c(".value", "transition"), #make wave variable and put t-v var
              names_sep = "_")
```

```
library(ggplot2)

dat1.long%>%
  filter(race_rec%in%c("nhwhite", "hispanic", "nhblack"))%>%
  filter( mgths==0)%>%
  mutate(educ = ifelse(mlths ==1, "Less than HS", "HS or More"))%>%
  ggplot(aes(y=p, x=age_enter, fill=transition))+
  geom_area()+
  facet_wrap(~race_rec+educ)+
  labs(title = "Poverty transition probabilities by race and education",
          caption = "Data: ECLS-K 2011")+
  scale_fill_brewer(palette = "Set1")
```


Poverty transition probabilities by race and education
hispanic HS or More
hispanic Less than HS
nhblack HS or More
nhblack Less than HS
nhwhite HS or More
nhwhite Less than HS
p
age_enter
transition
intopov
outofpov
stayinpov
staynotpov
Data: ECLS−K 2011