DEM 7223 - Event History Analysis - Discrete Time Hazard Model Part 1

true

October 12, 2020

Contents

Notes

Person-Level vs. Person-Period

- In the continuous time hazards world, the person-level data is satisfactory, because it provides:

1. the duration time for each individual in the data, which tells how long the person was at risk
2. the censoring indicator which tells if the person actually experienced the event of interest
3. the covariates that describe the effects of the person's characteristics on their "risk score"

- This data format was sufficient when we considered the continuous time treatment of the hazard model given by the K-M estimator, parametric models and the Cox model.

Person - period data

- In contrast, the person-period data represents a discrete-time representation of the duration data.
- In the person-period world, each observation contributes one "period" of risk for each time point they are at risk of experiencing the event
  - e.g. if we are talking about marital duration, and one couple experiences a divorce 2 years after marriage, if we measure "periods" as years, then this couple would contribute 2 "observations" to the data, because they were at risk the first year and experienced the event their second year.
- So each duration for each person, measured as time=Ti will have tij episodes in the new period-based data
- Each individual will exist in the data until they experience the event, or they are censored, and will contribute no more to the data after this point.
- This means that at each tij there is a certain number of persons at risk of experiencing the event and a certain number that do; which is the definition of a hazard rate
- The biggest issue is getting the data into the appropriate format
- We have already done this with the piecewise-constant hazard model
- There we specified arbitrary breaks, now we specify more structured time periods

Constructing Hazards for each period

- In the person-period world, we can construct the hazard of an event occurring at a specific time just as in the continuous time world.
- This is still the conditional failure probability at each time point, and is still calculated by dividing:

$$h(t) = \frac{\text{y failures}}{\text{n at risk}}$$

- This is very similar to how life-tables are constructed, except in this perspective we allow for individual covariate effects on the hazard.

Expressing the hazards in alternative forms:

- There are several ways to present the discrete time hazard rate
  - Probability scale (bound between 0 – 1)
  - These are tough bounds to work within
- Odds scale (bound 0 – inf)
  - odds= $\frac{h(t)}{1-h(t)}$, expresses the probability of an event occurring to the probability that it did not occur
  - Bad thing is that it includes 0 as a possible value
- log-odds (or logit) $ln\left(\frac{h(t)}{1-h(t)}\right)$, which is unbounded
  - log scale allows for easier comparison of hazards over time, makes big differences smaller, and small differences bigger and homogenizes variability

Modeling the hazard for each period

- This is typically done via logistic regression!
  - e.g.

$$logit(h(t)) = [\alpha_1 D_1 + \alpha_2 D_2 + ... + \alpha_t D_T] + x'\beta$$

* Where the D's represent the distinct risk periods, the α's represent the shape of the log-odds for the hazard at each time, and the β's represent the effects of covariates on the hazard.

- This formulation is called the *general form model* because it allows the shape of the hazard function to vary at each time point, as defined by the α's!
- This allows great flexibility in modeling the changing nature of risk
- It is constrained (for the time-being) on the assumption that the covariates are not time-dependent, and their effect is the same in each risk period
- Each of the α's represents the value of the logit hazard (log-odds) of an event occurring at that particular time point for individuals in the "baseline group" $\rightarrow$ no covariate effect
- The β's, or slope parameters, assess the effect of a 1 unit difference in a covariate on the risk of the event occurring.
- Similar to what we saw in the proportional hazards model, the discrete time model has the following interpretation
- When the covariates are not time-dependent
- if x_1 is our only covariate, the log-odds can be written:

$$\text{if x=1} : logit(h(t)) = [\alpha_1 D_1 + \alpha_2 D_2 + ... + \alpha_t D_T] + [\beta_1 x_1]$$

$$\text{if x=1} : logit(h(t)) = [\alpha_1 D_1 + \alpha_2 D_2 + ... + \alpha_t D_T]$$

- So when x=1 the entire hazard is increased (or decreased) by the constant, β.
- This effect is assumed to be constant at all time points and can be compared to the proportional hazards assumption
- When x=0 this is called the *baseline logit hazard function*
- The parameters $\alpha_1 D_1 + \alpha D_2 + ... + \alpha_t D_t$ act as t distinct intercept parameters, describing the baseline risk (on a log odds scale) at each risk period of time, T.

Expressions for the hazard

- Sometimes we want to express the hazard in a different scale (instead of the logit scale, we want the probability scale)
- Singer &Willett p 376 give transformations (or inversions) for going between the scales
  - These are the same as the transformations of the probability into the log-odds as in logistic regression
  - For instance if we want to express the entire regression in the probability scale we do:

$$h(t_{ij}) = \frac{1}{1 + exp - (\alpha_1 D_1 + \alpha_2 D_2 + ... + \alpha_t D_T + x'\beta)}$$

Interpreting the coefficients

- The α's represent the shape of the hazard at each time period. If they are approximately equal, then the hazard is flat over time, if they decline over time, so does the hazard and vice versa.
- We can also see nonlinear changes in the hazard an important consideration in interpreting the α's is the identification of the baseline group, this is the group for which all x's are 0, so you may need to standardize any continuous covariates.
- The β's represent the effect of the covariates.
- For a dichotomous predictor (or indeed a continuous predictor that is z scored), these indicate the shift in the log-odds for an individual with x=1 relative to x=0
- if $\beta > 0$, the odds ratio is >1, This represents an increase in the odds of experiencing the event, $\beta < 0$, and the odds ratio <1, represents a decrease in the odds of experiencing the event
- If we use dummy variable coding for a factor variable, remember the interpretation in always done in reference to the category you leave out

Data Examples

This example will illustrate how to fit the discrete time hazard model to longitudinal and continuous duration data (i.e. person-level data).

The first example will use as its outcome variable, the event of a child dying before age 5. The data for this example come from the model.data Demographic and Health Survey for 2012 children's recode file. This file contains information for all births in the last 5 years prior to the survey.

The longitudinal data example uses data from the ECLS-K. Specifically, we will examine the transition into poverty between kindergarten and 8th grade.

```
#Load required libraries
library(foreign)
library(survival)
library(car)
library(survey)
```

Using Longitudinal Data

As in the other examples, I illustrate fitting these models to data that are longitudinal, instead of person-duration. In this example, we will examine how to fit the Cox model to a longitudinally collected data set.

```
eclskk5<-readRDS("C:/Users/ozd504/OneDrive - University of Texas at San Antonio/classes/dem7223/dem7223
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")
eclskk5<-eclskk5[,myvars]
```

```
eclskk5$age1<-ifelse(eclskk5$x1kage_r==-9, NA, eclskk5$x1kage_r/12)
eclskk5$age2<-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 mo
eclskk5$age3<-ifelse(eclskk5$x5age==-9, NA, eclskk5$x5age/12)

eclskk5$pov1<-ifelse(eclskk5$x2povty==1,1,0)
eclskk5$pov2<-ifelse(eclskk5$x4povty_i==1,1,0)
eclskk5$pov3<-ifelse(eclskk5$x6povty_i==1,1,0)

#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='hispanic';5='nhasian'; 6:8='other';-9
                          as.factor = T)
eclskk5$race_rec<-relevel(eclskk5$race_rec, ref = "nhwhite")
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<-subset(eclskk5, is.na(pov1)==F&is.na(pov2)==F&is.na(pov3)==F&
                  is.na(age1)==F&is.na(age2)==F&
                  is.na(age3)==F&pov1!=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 `reshape()` function allows us to do this easily. It allows us to specify our repeated measures, time varying covariates as well as time-constant covariates.

```
e.long<-reshape(data.frame(eclskk5), idvar="childid",
                varying=list(c("age1","age2"),
                                                    c("age2", "age3")),
                v.names=c("age_enter", "age_exit"),
                times=1:2, direction="long" )
e.long<-e.long[order(e.long$childid, e.long$time),]

e.long$povtran<-NA

e.long$povtran[e.long$pov1==0&e.long$pov2==1&e.long$time==1]<-1
e.long$povtran[e.long$pov2==0&e.long$pov3==1&e.long$time==2]<-1

e.long$povtran[e.long$pov1==0&e.long$pov2==0&e.long$time==1]<-0
e.long$povtran[e.long$pov2==0&e.long$pov3==0&e.long$time==2]<-0

#find which kids failed in earlier time periods and remove them from the second & third period risk set
failed1<-which(is.na(e.long$povtran)==T)
e.long<-e.long[-failed1,]
```

```
e.long$age1r<-round(e.long$age_enter, 0)
e.long$age2r<-round(e.long$age_exit, 0)
e.long$time_start<-e.long$time-1
head(e.long[, c("childid","time_start" , "time",
                "age_enter", "age_exit", "pov1", "pov2",
                "pov3","povtran", "mlths")], n=10)
```

```
##             childid time_start time age_enter age_exit pov1 pov2 pov3 povtran
## 10000014.1 10000014          0    1  5.651667 7.161667    0    0    0       0
## 10000014.2 10000014          1    2  7.161667 7.644167    0    0    0       0
## 10000020.1 10000020          0    1  5.698333 7.380833    0    0    0       0
## 10000020.2 10000020          1    2  7.380833 7.780833    0    0    0       0
## 10000022.1 10000022          0    1  5.717500 7.306667    0    0    0       0
## 10000022.2 10000022          1    2  7.306667 7.748333    0    0    0       0
## 10000029.1 10000029          0    1  5.783333 7.238333    0    0    0       0
## 10000029.2 10000029          1    2  7.238333 7.723333    0    0    0       0
## 10000034.1 10000034          0    1  6.353333 7.775000    0    0    1       0
## 10000034.2 10000034          1    2  7.775000 8.295833    0    0    1       1
##            mlths
## 10000014.1     0
## 10000014.2     0
## 10000020.1     0
## 10000020.2     0
## 10000022.1     0
## 10000022.2     0
## 10000029.1     1
## 10000029.2     1
## 10000034.1     0
## 10000034.2     0
```

Descriptive analysis of rates

Since we have a simple binary outcome, we can take the mean of it and arrive at our basic rates of occurrence of our event.

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following object is masked from 'package:car':
##
##     recode

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
e.long%>%
  group_by(time)%>%
  summarise(prop_event= mean(povtran, na.rm=T))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 2 x 2
##    time prop_event
##   <int>      <dbl>
## 1     1     0.0719
## 2     2     0.0395
```

We can likewise do simple descriptive analysis of the outcome by characteristics of the respondents:

```
e.long%>%
  filter(complete.cases(mlths))%>%
  group_by(time, mlths)%>%
  summarise(prop_event= mean(povtran, na.rm=T))
```

```
## `summarise()` regrouping output by 'time' (override with `.groups` argument)
```

```
## # A tibble: 4 x 3
## # Groups:   time [2]
##    time mlths prop_event
##   <int> <dbl>      <dbl>
## 1     1     0     0.0586
## 2     1     1     0.276
## 3     2     0     0.0306
## 4     2     1     0.213
```

Now we fit the discrete time model using full survey design. In the ECLS-K, I use the longitudinal weight for waves 1-7, as well as the associated psu and strata id's for the longitudinal data from these waves from the parents of the child, since no data from the child themselves are used in the outcome.

```
options(survey.lonely.psu = "adjust")
e.long<-e.long%>%
  filter(complete.cases(w6c6p_6psu, race_rec, mlths))

des2<-svydesign(ids = ~w6c6p_6psu,
                strata = ~w6c6p_6str,
                weights=~w6c6p_20,
                data=e.long,
                nest=T)
```

Basic Discrete time model

Following the notation in the notes, we specify a generalized linear model for a binary outcome at each time period of risk. In this case, the binomial event is 0 if a child doesn't transition into poverty between waves, and 1 if they do. This can be specified as a logistic regression model with a choice of link functions. We can use a logit specification for the hazard:

$\text{logit}(h(t)) = [\alpha_1 D_1 + \alpha_2 D_2 + \cdots \alpha_t D_t] + \sum_k \beta_k x_k$

or, as I do below, use a complementary log-log link:

$log(-log(1 - h(t))) = [\alpha_1 D_1 + \alpha_2 D_2 + \cdots \alpha_t D_t] + \sum_k \beta_k x_k$

while the choice of link function generally has no effect on predictions, the interpretation of the log-log link is the same as the proportional hazards model, instead of the odds ratio interpretation.

```
#Fit the model
fitl1<-svyglm(povtran~as.factor(time_start)+mlths+race_rec-1,
              design=des2, family=binomial(link="cloglog"))
```

```
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
```

```
summary(fitl1)
```

```
##
## Call:
## svyglm(formula = povtran ~ as.factor(time_start) + mlths + race_rec -
##     1, design = des2, family = binomial(link = "cloglog"))
##
## Survey design:
## svydesign(ids = ~w6c6p_6psu, strata = ~w6c6p_6str, weights = ~w6c6p_20,
##     data = e.long, nest = T)
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## as.factor(time_start)0  -3.2163     0.2245 -14.325  < 2e-16 ***
## as.factor(time_start)1  -3.8138     0.2077 -18.363  < 2e-16 ***
## mlths                    1.0732     0.2260   4.749 1.05e-05 ***
## race_rechispanic         1.2158     0.2839   4.283 5.77e-05 ***
## race_recnhasian          0.2467     0.4218   0.585 0.560437
## race_recnhblack          1.0310     0.2818   3.658 0.000489 ***
## race_recother            0.5223     0.2772   1.885 0.063625 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 0.9539066)
##
## Number of Fisher Scoring iterations: 6
```

```
#make a Hazard ratio
sums<-data.frame(round(summary(fitl1)$coef, 3))
sums$HR<-round(exp(sums[,1]), 3)
sums
```

```
##                          Estimate Std..Error t.value Pr...t..    HR
## as.factor(time_start)0   -3.216      0.225 -14.325    0.000 0.040
## as.factor(time_start)1   -3.814      0.208 -18.363    0.000 0.022
## mlths                     1.073      0.226   4.749    0.000 2.924
## race_rechispanic          1.216      0.284   4.283    0.000 3.374
## race_recnhasian           0.247      0.422   0.585    0.560 1.280
## race_recnhblack           1.031      0.282   3.658    0.000 2.804
## race_recother             0.522      0.277   1.885    0.064 1.685
```

We can use the model to generate predicted probabilities for different types of people in our analysis:

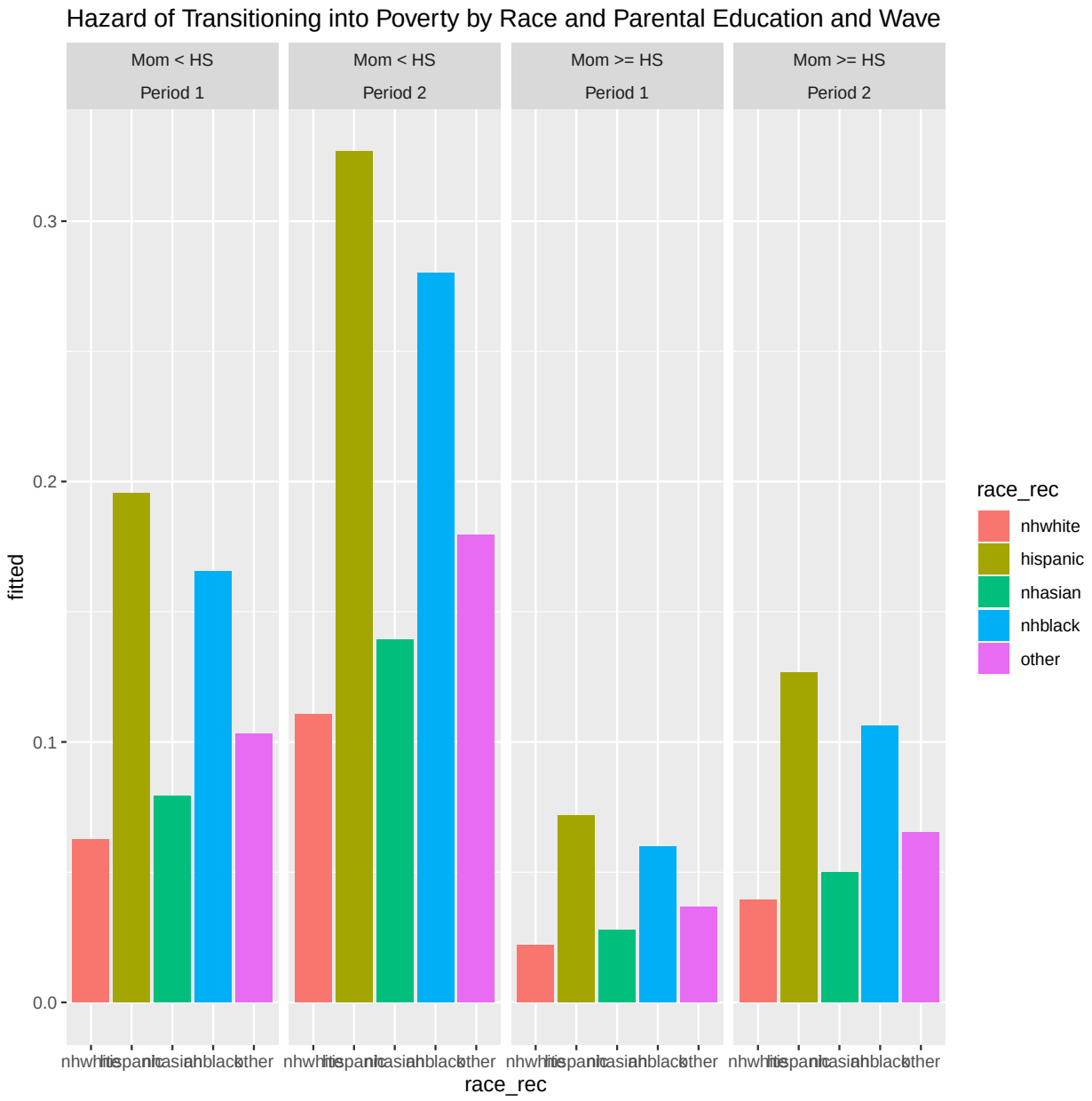
```
dat3<-expand.grid(time_start=as.factor(c(0,1)),mlths=c(0,1),
                  race_rec=levels(des2$variables$race_rec))
#unfortunately, expand.grid makes some unrealistic cases sometimes, get rid of those.

dat3$fitted<-as.numeric(predict(fitl1, dat3, type="response"))
head(dat3)
```

```
##   time_start mlths race_rec     fitted
## 1          0     0  nhwhite 0.03930971
## 2          1     0  nhwhite 0.02182251
## 3          0     1  nhwhite 0.11066899
## 4          1     1  nhwhite 0.06249086
## 5          0     0 hispanic 0.12652247
## 6          1     0 hispanic 0.07172290
```

Do some plots, these aren't very cool, since there is only 2 time periods.

```
library(ggplot2)
dat3%>%
  mutate(momedu = ifelse(mlths==1, "Mom < HS", "Mom >= HS"),
         group = paste(momedu, race_rec, sep="-"),
         period=ifelse(time_start==1, "Period 1", "Period 2"))%>%
  ggplot()+
  geom_bar(aes(y=fitted,x=race_rec,fill=race_rec, group=race_rec),stat="identity", position="dodge")+
  facet_grid(~momedu+period)+
  ggtitle(label="Hazard of Transitioning into Poverty by Race and Parental Education and Wave")
```



Continuous duration outcome - DHS data

load the data

```
library(haven)
model.dat<-read_dta("https://github.com/coreysparks/data/blob/master/ZZIR62FL.DTA?raw=true")
model.dat<-zap_labels(model.dat)
```

Event - Second birth occurrence

In the DHS individual recode file, information on every live birth is collected using a retrospective birth history survey mechanism.

Since our outcome is time between first and second birth, we must select as our risk set, only women who have had a first birth.

The `bidx` variable indexes the birth history and if `bidx_01` is not missing, then the woman should be at risk of having a second birth (i.e. she has had a first birth, i.e. `bidx_01==1`).

I also select only non-twin births (`b0 == 0`).

The DHS provides the dates of when each child was born in Century Month Codes.

To get the interval for women who *actually had* a second birth, that is the difference between the CMC for the first birth `b3_01` and the second birth `b3_02`, but for women who had not had a second birth by the time of the interview, the censored time between births is the difference between `b3_01` and `v008`, the date of the interview.

We have 6161 women who are at risk of a second birth.

```
#We form a subset of variables
sub<-subset(model.dat, model.dat$bidx_01==1&model.dat$b0_01==0)

#Here I keep only a few of the variables for the dates, and some characteristics of the women, and deta
sub2<-data.frame(CASEID=sub$caseid,
                 int.cmc=sub$v008,
                 fbir.cmc=sub$b3_01,
                 sbir.cmc=sub$b3_02,
                 marr.cmc=sub$v509,
                 rural=sub$v025,
                 educ=sub$v106,
                 age=sub$v012,
                 partneredu=sub$v701,
                 partnerage=sub$v730,
                 weight=sub$v005/1000000,
                 psu=sub$v021, strata=sub$v022)

sub2$agefb = (sub2$age - (sub2$int.cmc - sub2$fbir.cmc)/12)

#censoring indicator for death by age 5, in months (<=60 months)
sub2$secbi<-ifelse(is.na(sub2$sbir.cmc)==T,
                   ((sub2$int.cmc))-((sub2$fbir.cmc)),
                   (sub2$fbir.cmc-sub2$sbir.cmc))
sub2$b2event<-ifelse(is.na(sub2$sbir.cmc)==T,0,1)
table(sub2$b2event)
```

```
##
##    0    1
## 1237 4789
```

```
sub2$educ.high<-ifelse(sub2$educ %in% c(2,3), 1, 0)
sub2$age2<-(sub2$agefb)^2
sub2$partnerhiedu<-ifelse(sub2$partneredu<3,0,
                          ifelse(sub2$partneredu%in%c(8,9),NA,1 ))
```

```
options(survey.lonely.psu = "adjust")
des<-svydesign(ids=~psu, strata=~strata, data=sub2[sub2$secbi>0,], weight=~weight )
```

Create the person-period file

The distinction between the way we have been doing things and the discrete time model, is that we treat time discretely, versus continuously. This means that we transform the data from the case-duration data format to the person-period format. For this example, a natural choice would be year, since we have intervals of equal length (12 months each).

R provides a useful function called `survSplit()` in the `survival` library that will split a continuous duration into discrete periods.

```
#make person period file
pp<-survSplit(Surv(secbi, b2event)~. , data = sub2[sub2$secbi>0,],
              cut=c(0,12, 24, 36, 48, 60, 72),  episode="year_birth")

pp$year <- pp$year_birth-1
pp<-pp[order(pp$CASEID, pp$year_birth),]
head(pp[, c("CASEID", "secbi", "b2event", "year", "educ.high", "agefb")], n=20)
```

```
##             CASEID secbi b2event year educ.high    agefb
## 1          1  1  2    12       0    1         0 29.58333
## 2          1  1  2    24       0    2         0 29.58333
## 3          1  1  2    32       1    3         0 29.58333
## 4          1  3  2    12       0    1         1 19.50000
## 5          1  3  2    24       0    2         1 19.50000
## 6          1  3  2    30       0    3         1 19.50000
## 7          1  4  2    12       0    1         0 31.66667
## 8          1  4  2    24       0    2         0 31.66667
## 9          1  4  2    32       1    3         0 31.66667
## 10         1  4  3    12       0    1         0 24.16667
## 11         1  4  3    20       1    2         0 24.16667
## 12         1  5  1    12       0    1         1 24.91667
## 13         1  5  1    24       0    2         1 24.91667
## 14         1  5  1    33       1    3         1 24.91667
## 15         1  6  2    12       0    1         0 31.75000
## 16         1  6  2    24       0    2         0 31.75000
## 17         1  6  2    36       0    3         0 31.75000
## 18         1  6  2    48       0    4         0 31.75000
## 19         1  6  2    60       0    5         0 31.75000
## 20         1  6  2    72       0    6         0 31.75000
```

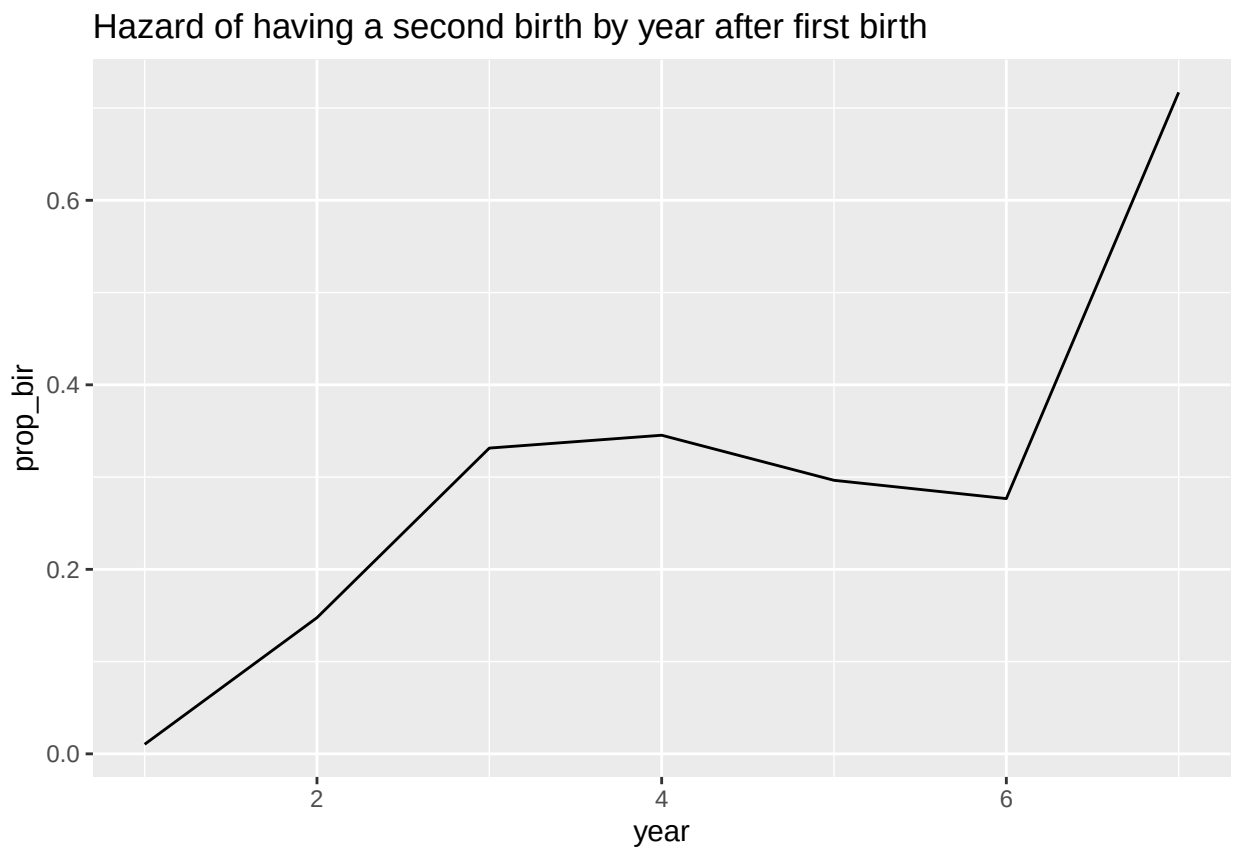
We see that each child is not in the data for multiple "risk periods", until they experience the event (death) or age out of the risk set (year 6 in this case).

Descriptive analysis

```
pp%>%
  group_by(year)%>%
```

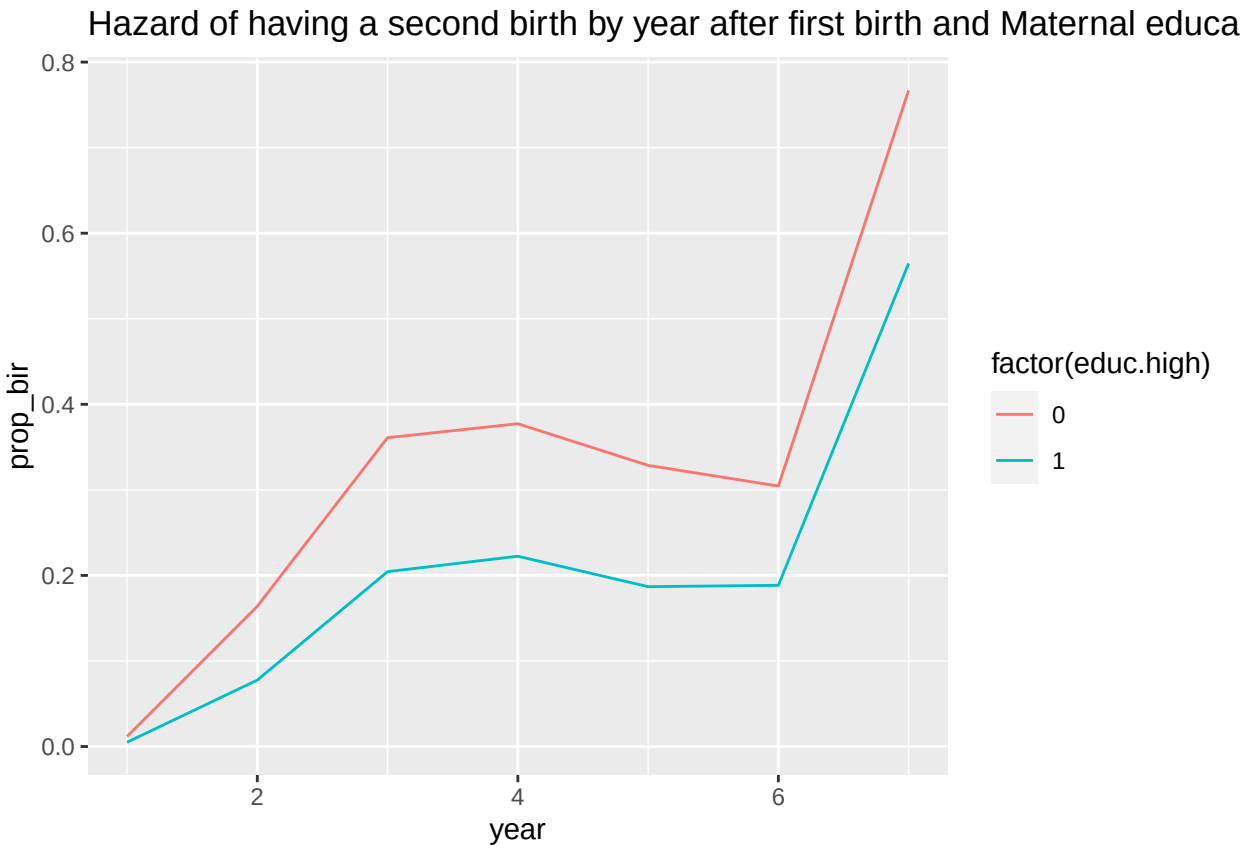
```
  summarise(prop_bir=mean(b2event, na.rm=T))%>%
  ggplot(aes(x=year, y=prop_bir))+
  geom_line()+
  ggtitle(label = "Hazard of having a second birth by year after first birth")
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```




```
pp%>%
  group_by(year, educ.high)%>%
  summarise(prop_bir=mean(b2event, na.rm=T))%>%
  ggplot(aes(x=year, y=prop_bir))+
  geom_line(aes(group=factor(educ.high), color=factor(educ.high) ))+
  ggtitle(label = "Hazard of having a second birth by year after first birth and Maternal education")
```

```
## `summarise()` regrouping output by 'year' (override with `.groups` argument)
```



Discrete time model

So, the best thing about the discrete time model, is that it's just logistic regression. Each risk period is treated as a single Bernoulli trial, and the child can either fail (y=1) or not (y=0) in the period. This is how we get the hazard of the event, as the estimated probability of failure in each discrete time period. So, any method you would like to use to model this probability would probably work (logit, probit models), but I will show two standard approaches. We will use the complementary log-log link. This is used because it preserves the proportional hazards property of the model, as in the Cox model.

```
#generate survey design

des<-svydesign(ids=~psu, strata = ~strata , weights=~weight, data=pp)

#Fit the basic logistic model with ONLY time in the model
#I do -1 so that no intercept is fit in the model, and we get a hazard estimate for each time period
fit.0<-svyglm(b2event~as.factor(year)-1,design=des , family="binomial")
```

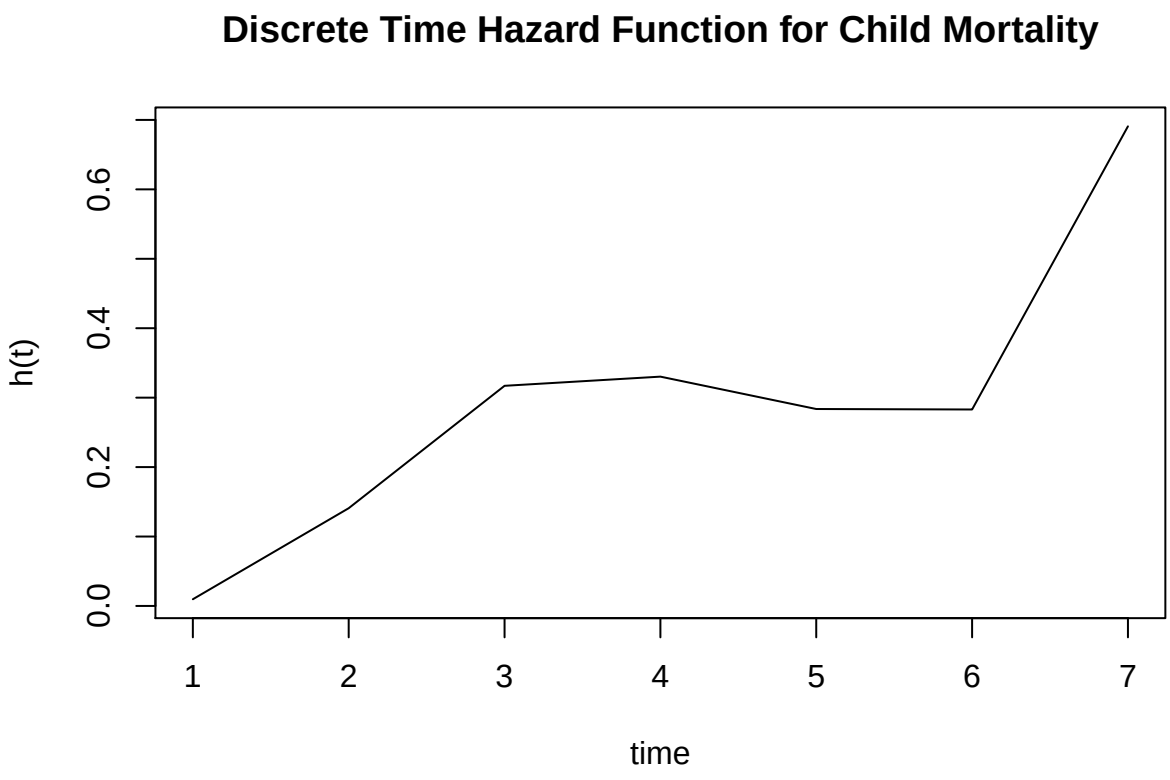
```
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
```

```
summary(fit.0)
```

```
##
## Call:
## svyglm(formula = b2event ~ as.factor(year) - 1, design = des,
```

```
##     family = "binomial")
##
## Survey design:
## svydesign(ids = ~psu, strata = ~strata, weights = ~weight, data = pp)
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## as.factor(year)1 -4.62968    0.17640 -26.245  < 2e-16 ***
## as.factor(year)2 -1.80896    0.05286 -34.222  < 2e-16 ***
## as.factor(year)3 -0.76732    0.04603 -16.669  < 2e-16 ***
## as.factor(year)4 -0.70704    0.06203 -11.399  < 2e-16 ***
## as.factor(year)5 -0.92652    0.06158 -15.045  < 2e-16 ***
## as.factor(year)6 -0.92979    0.08186 -11.359  < 2e-16 ***
## as.factor(year)7  0.80355    0.10880   7.386  5.1e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1.000044)
##
## Number of Fisher Scoring iterations: 7
```

```r
#Plot the hazard function on the probability scale
haz<-1/(1+exp(-coef(fit.0)))
time<-seq(1,7,1)
plot(haz~time, type="l", ylab="h(t)")
title(main="Discrete Time Hazard Function for Child Mortality")
```



Now we include a single predictor and examine the proportional hazards

```
fit.1<-svyglm(b2event~as.factor(year)+educ.high-1,design=des , family=binomial(link="cloglog"))
```

```
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
```

```
summary(fit.1)
```

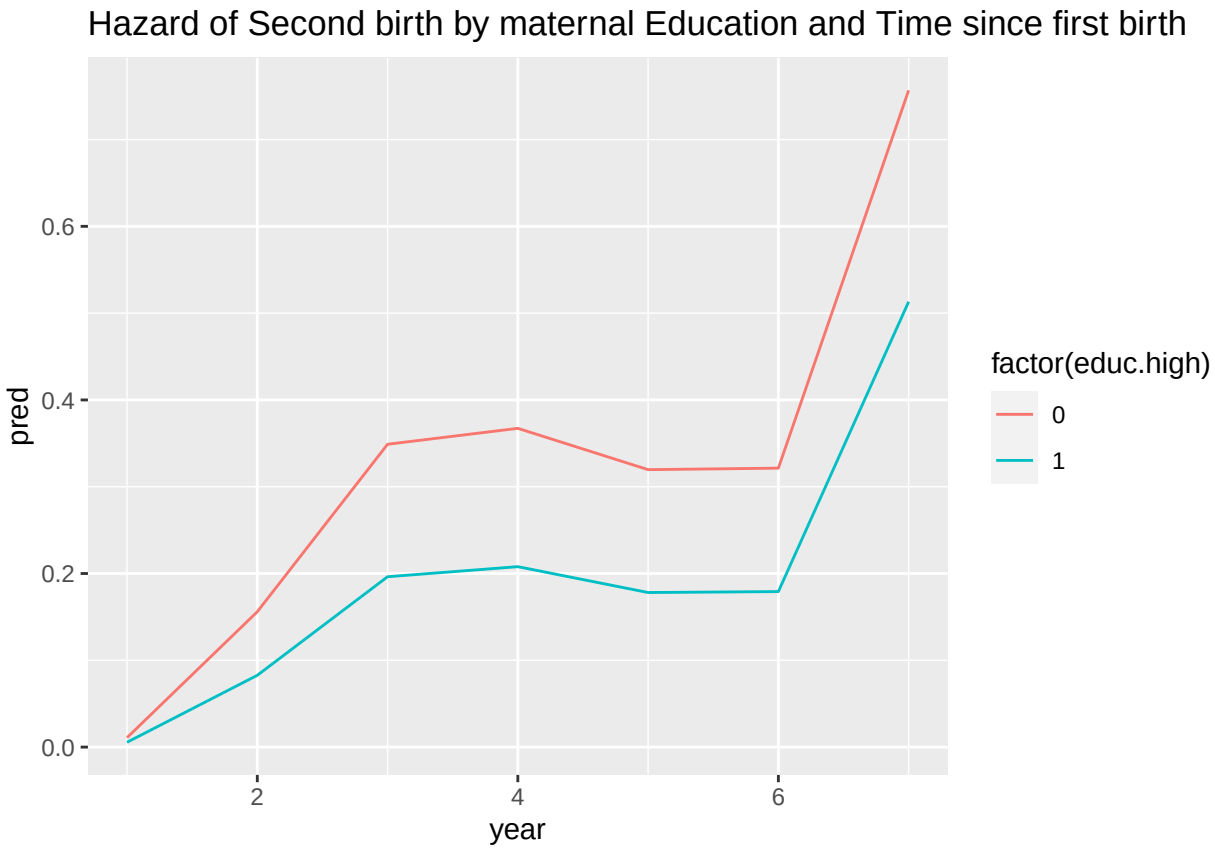
```
##
## Call:
## svyglm(formula = b2event ~ as.factor(year) + educ.high - 1, design = des,
##     family = binomial(link = "cloglog"))
##
## Survey design:
## svydesign(ids = ~psu, strata = ~strata, weights = ~weight, data = pp)
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## as.factor(year)1 -4.52122    0.17466 -25.885  < 2e-16 ***
## as.factor(year)2 -1.77441    0.04801 -36.957  < 2e-16 ***
## as.factor(year)3 -0.84604    0.03879 -21.812  < 2e-16 ***
## as.factor(year)4 -0.78154    0.04903 -15.940  < 2e-16 ***
## as.factor(year)5 -0.95405    0.05108 -18.677  < 2e-16 ***
## as.factor(year)6 -0.94747    0.06365 -14.885  < 2e-16 ***
## as.factor(year)7  0.34610    0.06032   5.737 3.92e-08 ***
## educ.high        -0.67489    0.05940 -11.361  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 0.9899529)
##
## Number of Fisher Scoring iterations: 7
```

Which shows a lower risk of having a child for highly educated mothers. In fact, it's a -49.08% lower risk

Next, I do the plots of the hazard functions the Singer and Willett way. I got this code from Singer and Willett's example from Ch 11

```
dat4<-expand.grid(year=1:7, educ.high=c(0,1))
dat4$pred<-as.numeric(predict(fit.1, newdata=dat4, type="response"))

dat4%>%
  ggplot(aes(x=year, y=pred,color=factor(educ.high), group=factor(educ.high) ))+
  geom_line()+
  ggtitle(label="Hazard of Second birth by maternal Education and Time since first birth")
```



See, same thing.