# DEM 7223 - Event History Analysis - Cox Proportional Hazards Model Part 2

true

October 5, 2020

## Contents

## Review of Cox Regression Assumptions

- Although the Cox PH model offers an attractive alternative to parametric models, especially when ties are present, the assumptions of the model need to be assessed
- The primary assumption we are concerned with are:
- The time-constant covariate effect, i.e. the effect does not vary with time
- Grambsch and Therneau (1994) derived a method for checking the proportionality assumption for the Cox model using the residuals from the Cox model fit
- First we need to look at the various kinds of residuals from the Cox model and their properties

### Cox Model Residuals

The basic principle for the construction of residuals is:

- Observed value – Predicted value
- This lets you get an idea of how well you are modeling your data with your covariates
- In hazard models, this principle is a bit more difficult because of censoring in the data, but despite this, we can define several types of model residuals and use them to diagnose problems with the model.

### Schoenfeld Residuals

- These are used to test the proportionality assumption of the model
- If there are $p$ covariates and $n$ observations, with observed durations, censoring indicators and covariates, then the Schoenfeld residual is defined as the observed - expected value of a covariate at a particular *failure time*
- Plotting these residuals versus time should show any time dependency in the covariate, which violates the proportionality assumption of the model.

### Martingale residuals

- These residuals can be thought of as the difference between an expected event occurring and an actual event occurring
- These use the censoring indicator for each observation and the estimate of the cumulative hazard function
- These residuals are given by:

$$M_i(t) = \delta_i(t)\check{}H_i(t)$$

- Where $H_i(t)$ is the cumulative hazard function
- This residual is derived from counting process theory and represents the difference between the observed count of failures at any time, minus those that are predicted by the cumulative hazard function.
- Martingale residuals are also useful for assessing the functional form of a covariate
- Meaning, is the effect linear or quadratic
- A plot of the martingale residuals against the values of the covariate will indicate if the covariate is being modeled correctly
- If a line fit to these residuals is a straight line, then the covariate has been modeled effectively, if it is curvilinear, you may need to enter the covariate as a quadratic
- This is usually not a problem for binary covariates

### Testing non-proportional effects

- As mentioned before, Grambsch and Therneau (1994) derived a method for checking the proportionality assumption for the Cox model using the residuals from the Cox model fit
- Now that we know what these residuals are, we can see the test
- Their test is equivalent to regressing the Schoenfeld residual on time for each covariate
- If there is a significant trend (correlation) between the residual and time, then non-proportionality is likely.

## Model stratification

- One common method for dealing with non proportionality of hazards is via model stratification.
- If one of the covariates exhibits non-proportionality we can re-specify the model so that each group will have its own baseline hazard rate
- The effect of the other covariates in the model is assumed to behave the same in both groups!
- This creates the model:

$$h_{is}(t) = h_{0s} exp(x'\beta)$$

- which allows for different baseline hazard rates for each of the $s$ strata, which should control for their unequal hazards of experiencing the event.
- This procedure is slightly different than fitting separate models for each level of the stratification variable
- This method will allow the effects of the covariates to vary between strata
- Unfortunately, when we split the data and run separate models for each level, we lose the ability to discuss "between level" effects, since each analysis is run on a different sample

## Data examples

This example will illustrate how to examine the fit of the Cox Proportional hazards model to a discrete-time (longitudinal) data set and examine various model diagnostics to evaluate the overall model fit. The data example uses data from the ECLS-K. Specifically, we will examine the transition into poverty between kindergarten and third grade.

```
options( "digits"=4)
#Load required libraries
library(foreign)
library(survival)
library(car)
```

```
## Loading required package: carData
```

```
library(survey)
```

```
## Loading required package: grid
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: ’survey’
```

```
## The following object is masked from ’package:graphics’:
##
##     dotchart
```

```
library(muhaz)
```

## 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.

First we load our data

```
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=
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.n
```

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     5.652    7.162    0    0    0       0
## 10000014.2 10000014          1    2     7.162    7.644    0    0    0       0
## 10000020.1 10000020          0    1     5.698    7.381    0    0    0       0
## 10000020.2 10000020          1    2     7.381    7.781    0    0    0       0
## 10000022.1 10000022          0    1     5.718    7.307    0    0    0       0
## 10000022.2 10000022          1    2     7.307    7.748    0    0    0       0
## 10000029.1 10000029          0    1     5.783    7.238    0    0    0       0
## 10000029.2 10000029          1    2     7.238    7.723    0    0    0       0
## 10000034.1 10000034          0    1     6.353    7.775    0    0    1       0
## 10000034.2 10000034          1    2     7.775    8.296    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
```

### Construct survey design and fit basic Cox model

Now we fit the Cox 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")

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<-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)

#Fit the model
fitl1<-svycoxph(Surv(time = time_start, time2=time, event = povtran)~mlths+race_rec, design=des2)
summary(fitl1)
```

```
## Stratified 1 - level Cluster Sampling design (with replacement)
## With (123) clusters.
## svydesign(ids = ~w6c6p_6psu, strata = ~w6c6p_6str, weights = ~w6c6p_20,
##     data = e.long, nest = T)

## Call:
## svycoxph(formula = Surv(time = time_start, time2 = time, event = povtran) ~
##     mlths + race_rec, design = des2)
##
##   n= 3938, number of events= 221
##
##                   coef exp(coef) se(coef) robust se    z Pr(>|z|)
## mlths            1.068     2.910    0.176     0.225 4.74  2.1e-06 ***
## race_rechispanic 1.215     3.370    0.162     0.287 4.23  2.3e-05 ***
## race_recnhasian  0.247     1.280    0.377     0.424 0.58  0.56076
## race_recnhblack  1.030     2.802    0.224     0.282 3.65  0.00026 ***
## race_recother    0.522     1.686    0.280     0.278 1.88  0.06033 .
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
##                  exp(coef) exp(-coef) lower .95 upper .95
## mlths                 2.91      0.344     1.871      4.52
## race_rechispanic      3.37      0.297     1.920      5.91
## race_recnhasian       1.28      0.781     0.557      2.94
## race_recnhblack       2.80      0.357     1.612      4.87
## race_recother         1.69      0.593     0.978      2.91
##
## Concordance= 0.702  (se = 0.021 )
## Likelihood ratio test= NA  on 5 df,   p=NA
## Wald test            = 88.4  on 5 df,   p=<2e-16
## Score (logrank) test = NA  on 5 df,   p=NA
##
##   (Note: the likelihood ratio and score tests assume independence of
##      observations within a cluster, the Wald and robust score tests do not).
```

### Model Residuals

There are several types of residuals for the Cox model, and they are used for different purposes.

First, we will extract the *Shoenfeld* residuals, which are useful for examining non-proportional hazards with respect to time. This means that the covariate effect could exhibit time-dependency.

First we extract the residuals from the model, then we fit a linear model to the residual and the observed (uncensored) failure times

## WE DO NOT WANT TO SEE A SIGNIFICANT MODEL HERE!!!!!

that would indicate dependence between the residual and outcome, or *nonpropotionality*, similar to doing a test for heteroskedasticity in OLS

```r
schoenresid<-resid(fitl1, type="schoenfeld")

fit.sr<-lm(schoenresid~des2$variables$time[des2$variables$povtran==1])
summary(fit.sr)
```

```
## Response mlths :
##
## Call:
## lm(formula = mlths ~ des2$variables$time[des2$variables$povtran ==
##     1])
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -0.287 -0.287 -0.167 -0.135  0.833
##
## Coefficients:
##                                                   Estimate Std. Error t value
## (Intercept)                                         0.1895     0.0854    2.22
## des2$variables$time[des2$variables$povtran == 1]  -0.1199     0.0601   -2.00
##                                                   Pr(>|t|)
## (Intercept)                                           0.027 *
## des2$variables$time[des2$variables$povtran == 1]    0.047 *
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 0.423 on 219 degrees of freedom
## Multiple R-squared:  0.0179, Adjusted R-squared:  0.0134
## F-statistic: 3.98 on 1 and 219 DF,  p-value: 0.0472
##
##
## Response race_rechispanic :
##
## Call:
## lm(formula = race_rechispanic ~ des2$variables$time[des2$variables$povtran ==
##     1])
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -0.567 -0.537  0.433  0.463  0.570
##
## Coefficients:
##                                                  Estimate Std. Error t value
## (Intercept)                                        0.1748     0.1010    1.73
## des2$variables$time[des2$variables$povtran == 1]  -0.1071     0.0711   -1.51
##                                                  Pr(>|t|)
## (Intercept)                                         0.085 .
## des2$variables$time[des2$variables$povtran == 1]    0.133
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 0.501 on 219 degrees of freedom
## Multiple R-squared:  0.0103, Adjusted R-squared:  0.00574
## F-statistic: 2.27 on 1 and 219 DF,  p-value: 0.133
##
##
## Response race_recnhasian :
##
## Call:
## lm(formula = race_recnhasian ~ des2$variables$time[des2$variables$povtran ==
##     1])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.0764 -0.0764 -0.0746 -0.0263  0.9719
##
## Coefficients:
##                                                  Estimate Std. Error t value
## (Intercept)                                        0.0909     0.0475    1.91
## des2$variables$time[des2$variables$povtran == 1]  -0.0483     0.0334   -1.45
##                                                  Pr(>|t|)
## (Intercept)                                         0.057 .
## des2$variables$time[des2$variables$povtran == 1]    0.150
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

```
##
## Residual standard error: 0.235 on 219 degrees of freedom
## Multiple R-squared:  0.00946,    Adjusted R-squared:  0.00493
## F-statistic: 2.09 on 1 and 219 DF,  p-value: 0.15
##
##
## Response race_recnhblack :
##
## Call:
## lm(formula = race_recnhblack ~ des2$variables$time[des2$variables$povtran ==
##     1])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.1350 -0.1329 -0.0834 -0.0813  0.9187
##
## Coefficients:
##                                                 Estimate Std. Error t value
## (Intercept)                                      -0.0847     0.0605   -1.40
## des2$variables$time[des2$variables$povtran == 1]   0.0515     0.0426    1.21
##                                                 Pr(>|t|)
## (Intercept)                                         0.16
## des2$variables$time[des2$variables$povtran == 1]     0.23
##
## Residual standard error: 0.3 on 219 degrees of freedom
## Multiple R-squared:  0.00664,    Adjusted R-squared:  0.00211
## F-statistic: 1.46 on 1 and 219 DF,  p-value: 0.228
##
##
## Response race_recother :
##
## Call:
## lm(formula = race_recother ~ des2$variables$time[des2$variables$povtran ==
##     1])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.0830 -0.0792 -0.0640 -0.0601  0.9399
##
## Coefficients:
##                                                 Estimate Std. Error t value
## (Intercept)                                      -0.0229     0.0510   -0.45
## des2$variables$time[des2$variables$povtran == 1]   0.0191     0.0359    0.53
##                                                 Pr(>|t|)
## (Intercept)                                         0.65
## des2$variables$time[des2$variables$povtran == 1]     0.60
##
## Residual standard error: 0.253 on 219 degrees of freedom
## Multiple R-squared:  0.00129,    Adjusted R-squared:  -0.00327
## F-statistic: 0.283 on 1 and 219 DF,  p-value: 0.595
```

From these results, it appears that the `mlths` variable is correlated with the timing of transition, while the other variables are constant over **time**

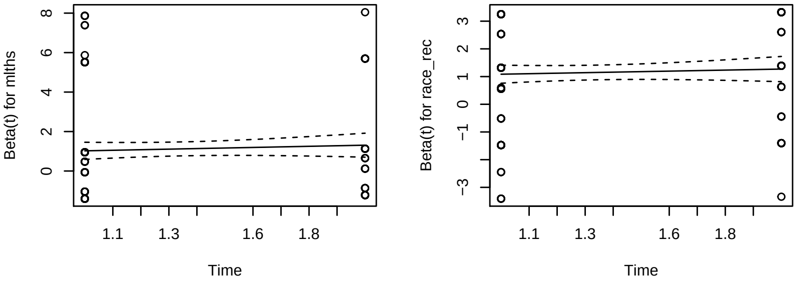
*Not so soon*

We can also get a formal test using weighted residuals in a nice pre-rolled form with a plot, a la Grambsch and Therneau (1994) :

```
fit.test<-cox.zph(fitl1)
fit.test
```

```
##             chisq df    p
## mlths        1.25  1 0.26
## race_rec     3.21  4 0.52
## GLOBAL       3.43  5 0.63
```

```
par(mfrow=c(3,3))
plot(fit.test, df=2)
par(mfrow=c(1,1))
```

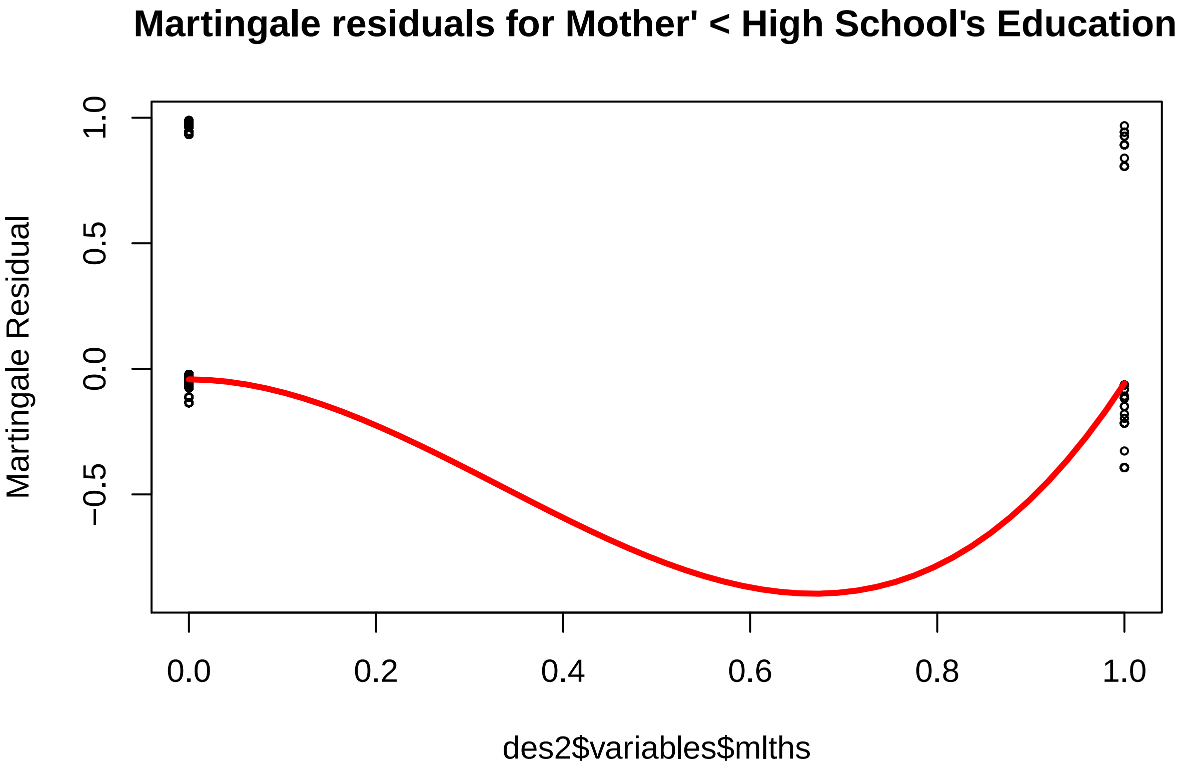


Here, we see that opposite, with no significant relationship detected by the formal test. **This is what you want to see**.

Next we examine Martingale residuals. Martingale residuals are also useful for assessing the functional form of a covariate. A plot of the martingale residuals against the values of the covariate will indicate if the covariate is being modeled correctly, i.e. linearly in the Cox model. If a line fit to these residuals is a straight line, then the covariate has been modeled effectively, if it is curvilinear, you may need to enter the covariate as a quadratic, although this is not commonly a problem for dummy variables.

```
#extract Martingale residuals
res.mar<-resid(fitl1, type="martingale")

#plot vs maternal education
scatter.smooth(des2$variables$mlths, res.mar,degree = 2,
               span = 1, ylab="Martingale Residual",
               col=1,  cex=.5, lpars=list(col = "red", lwd = 3))
title(main="Martingale residuals for Mother' < High School's Education")
```



Which shows nothing in the way of nonlinearity in this case.

### Stratification

Above, we observed evidence of non-proportional effects by education. There are a few standard ways of dealing with this in practice. The first is *stratification* of the model by the offending predictor. If one of the covariates exhibits non-proportionality we can re-specify the model so that each group will have its own baseline hazard rate. This is direct enough to do by using the `strata()` function within a model. This is of best use when a covariate is categorical, and not of direct importance for our model (i.e. a control variable).

```r
fitl2<-svycoxph(Surv(time = time_start, time2 = time, event = povtran)~race_rec+strata(mlths),
                design=des2)
summary(fitl2)
```

```
## Stratified 1 - level Cluster Sampling design (with replacement)
## With (123) clusters.
## svydesign(ids = ~w6c6p_6psu, strata = ~w6c6p_6str, weights = ~w6c6p_20,
##     data = e.long, nest = T)

## Call:
## svycoxph(formula = Surv(time = time_start, time2 = time, event = povtran) ~
##     race_rec + strata(mlths), design = des2)
##
##   n= 3938, number of events= 221
##
##                     coef exp(coef) se(coef) robust se    z Pr(>|z|)
## race_rechispanic 1.214     3.366    0.162     0.287 4.23  2.4e-05 ***
## race_recnhasian  0.247     1.280    0.377     0.424 0.58  0.56062
## race_recnhblack  1.030     2.800    0.224     0.282 3.65  0.00026 ***
## race_recother    0.521     1.685    0.280     0.278 1.87  0.06081 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##                   exp(coef) exp(-coef) lower .95 upper .95
## race_rechispanic      3.37      0.297     1.918      5.91
## race_recnhasian       1.28      0.781     0.557      2.94
## race_recnhblack       2.80      0.357     1.611      4.87
## race_recother         1.68      0.594     0.977      2.91
##
## Concordance= 0.663  (se = 0.024 )
## Likelihood ratio test= NA  on 4 df,   p=NA
## Wald test            = 23.9  on 4 df,   p=8e-05
## Score (logrank) test = NA  on 4 df,   p=NA
##
##   (Note: the likelihood ratio and score tests assume independence of
##      observations within a cluster, the Wald and robust score tests do not).
```

### Non-proportional effects with time

We can also include a time by covariate interaction term to model directly any time-dependence in the covariate effect. Different people say to do different things, some advocate for simply interacting time with the covariate, others say use a nonlinear function of time, e.g. log(time) * the covariate, others say use time-1 * covariate, which is called the "heavy side function", according to Mills.

In this example, time is so limited that it doesn't make sense to do this.

## ANOVA like tests for factors

You can use the `regTermTest()` function in the `survey()` package to do omnibus tests for variation across a factor variable.

```
fit3<-svycoxph(Surv(time = time_start, time2=time, event = povtran)~mlths+race_rec,
               design=des2)
summary(fit3)
```

```
## Stratified 1 - level Cluster Sampling design (with replacement)
## With (123) clusters.
## svydesign(ids = ~w6c6p_6psu, strata = ~w6c6p_6str, weights = ~w6c6p_20,
##     data = e.long, nest = T)
```

```
## Call:
## svycoxph(formula = Surv(time = time_start, time2 = time, event = povtran) ~
##     mlths + race_rec, design = des2)
##
##   n= 3938, number of events= 221
##
##                    coef exp(coef) se(coef) robust se    z Pr(>|z|)
## mlths             1.068     2.910    0.176     0.225 4.74  2.1e-06 ***
## race_rechispanic 1.215     3.370    0.162     0.287 4.23  2.3e-05 ***
## race_recnhasian  0.247     1.280    0.377     0.424 0.58  0.56076
## race_recnhblack  1.030     2.802    0.224     0.282 3.65  0.00026 ***
## race_recother    0.522     1.686    0.280     0.278 1.88  0.06033 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##                  exp(coef) exp(-coef) lower .95 upper .95
## mlths                 2.91      0.344     1.871      4.52
## race_rechispanic      3.37      0.297     1.920      5.91
## race_recnhasian       1.28      0.781     0.557      2.94
## race_recnhblack       2.80      0.357     1.612      4.87
## race_recother         1.69      0.593     0.978      2.91
##
## Concordance= 0.702  (se = 0.021 )
## Likelihood ratio test= NA  on 5 df,   p=NA
## Wald test            = 88.4  on 5 df,   p=<2e-16
## Score (logrank) test = NA  on 5 df,   p=NA
##
##   (Note: the likelihood ratio and score tests assume independence of
##      observations within a cluster, the Wald and robust score tests do not).
```

```
regTermTest(fit3, ~mlths, method="LRT")
```

```
## Working (Rao-Scott+F) LRT for mlths
##  in svycoxph(formula = Surv(time = time_start, time2 = time, event = povtran) ~
##     mlths + race_rec, design = des2)
## Working 2logLR =  28.96 p= 9.7e-07
## df=1;  denominator df= 72
```

```
regTermTest(fit3, ~race_rec, method="LRT")
```

```
## Working (Rao-Scott+F) LRT for race_rec
##  in svycoxph(formula = Surv(time = time_start, time2 = time, event = povtran) ~
##     mlths + race_rec, design = des2)
## Working 2logLR =  52.99 p= 6.5e-07
## (scale factors:  1.5 1.3 0.74 0.42 );  denominator df= 72
```

## DHS data example

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

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.

```
table(is.na(model.dat$bidx_01))
```

```
##
## FALSE  TRUE
##  6161  2187
```

```
#now we extract those women
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)
```

Now I need to calculate the birth intervals, both observed and censored, and the event indicator (i.e. did the women *have* the second birth?)

```
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)
```

### Create covariates

Here, we create some predictor variables: Woman's education (secondary +, vs < secondary), Woman's age^2, Partner's education (> secondary school)

```
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 )
```

### Fit the model

```
#use survey design
des<-svydesign(ids=~psu, strata = ~strata , weights=~weight, data=sub2[is.na(sub2$partnerhiedu)==F,])

cox.s<-svycoxph(Surv(secbi,b2event)~educ.high+partnerhiedu+agefb+age2,
                design=des)
summary(cox.s)
```

```
## Stratified 1 - level Cluster Sampling design (with replacement)
## With (217) clusters.
## svydesign(ids = ~psu, strata = ~strata, weights = ~weight, data = sub2[is.na(sub2$partnerhiedu) ==
##     F, ])
```

```
## Call:
## svycoxph(formula = Surv(secbi, b2event) ~ educ.high + partnerhiedu +
##     agefb + age2, design = des)
##
##   n= 5289, number of events= 4527
##
##                   coef  exp(coef)  se(coef) robust se      z Pr(>|z|)
## educ.high    -0.392111  0.675629  0.045754  0.049620  -7.90  2.7e-15 ***
## partnerhiedu -0.282617  0.753809  0.067577  0.087659  -3.22   0.0013 **
## agefb         0.205506  1.228146  0.016829  0.019935  10.31  < 2e-16 ***
## age2         -0.003409  0.996597  0.000284  0.000324 -10.54  < 2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
```

```
##              exp(coef) exp(-coef) lower .95 upper .95
## educ.high        0.676      1.480     0.613     0.745
## partnerhiedu     0.754      1.327     0.635     0.895
## agefb            1.228      0.814     1.181     1.277
## age2             0.997      1.003     0.996     0.997
##
## Concordance= 0.544  (se = 0.006 )
## Likelihood ratio test= NA  on 4 df,   p=NA
## Wald test            = 140  on 4 df,   p=<2e-16
## Score (logrank) test = NA  on 4 df,   p=NA
##
##   (Note: the likelihood ratio and score tests assume independence of
##      observations within a cluster, the Wald and robust score tests do not).
```

```
cox.s2<-svycoxph(Surv(secbi,b2event)~educ.high+partnerhiedu+agefb+age2,
                 design=des)
summary(cox.s2)
```

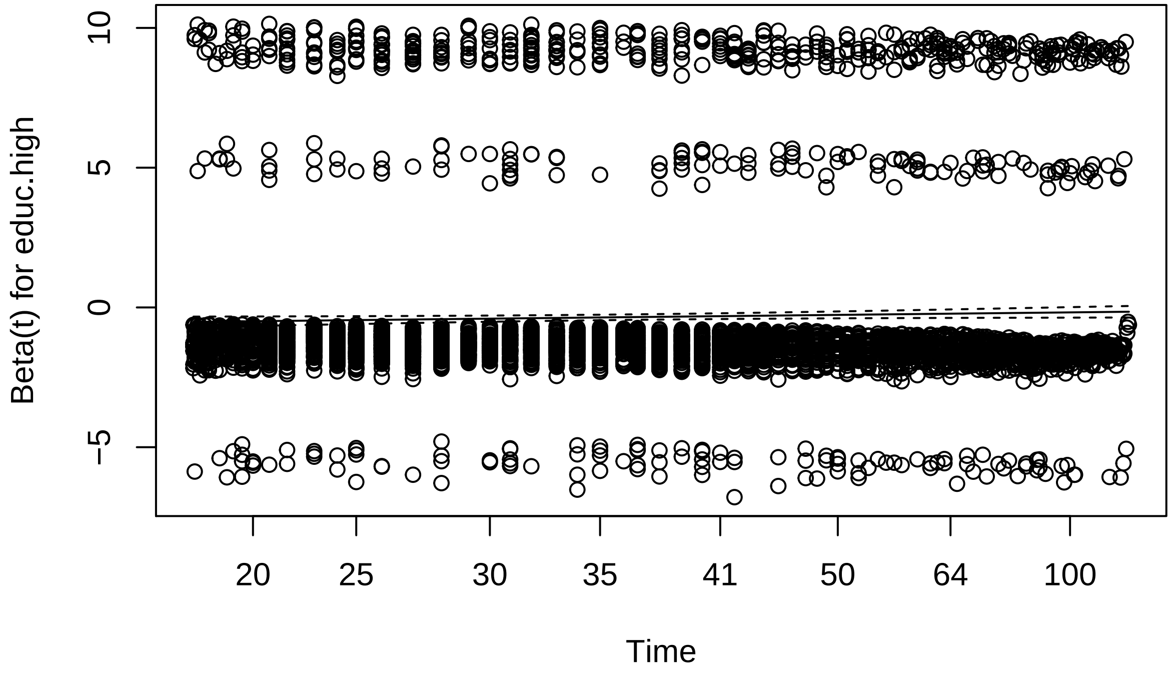
```
## Stratified 1 - level Cluster Sampling design (with replacement)
## With (217) clusters.
## svydesign(ids = ~psu, strata = ~strata, weights = ~weight, data = sub2[is.na(sub2$partnerhiedu) ==
##     F, ])
```

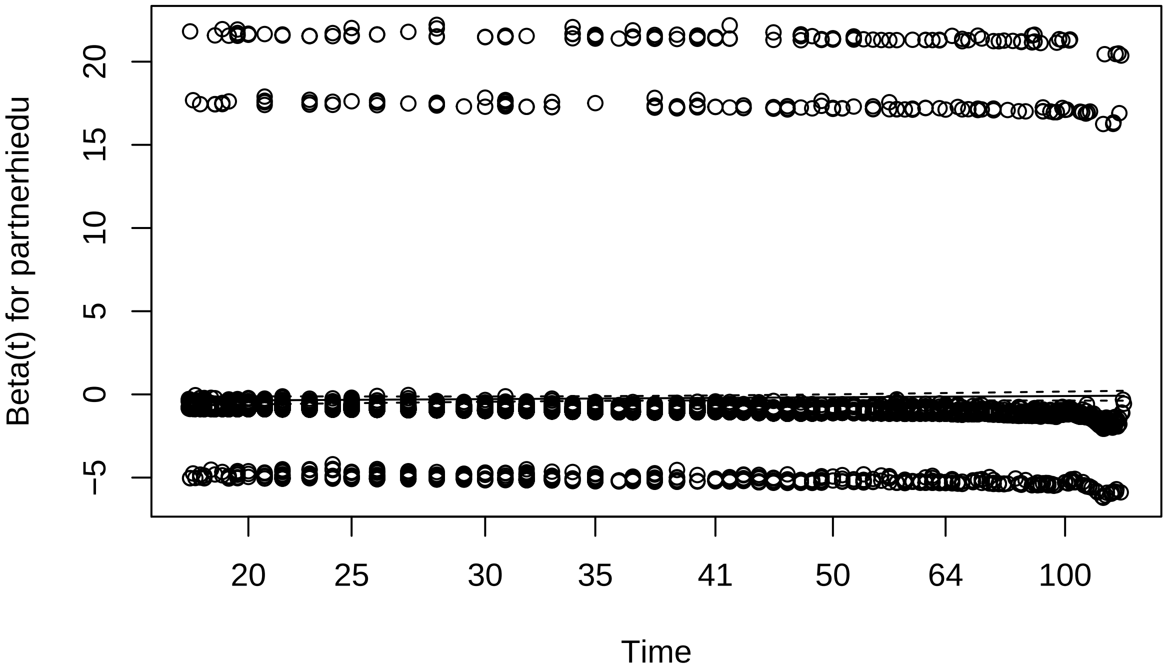
```
## Call:
## svycoxph(formula = Surv(secbi, b2event) ~ educ.high + partnerhiedu +
##     agefb + age2, design = des)
##
##   n= 5289, number of events= 4527
##
##                   coef exp(coef)  se(coef) robust se      z Pr(>|z|)
## educ.high    -0.392111  0.675629  0.045754  0.049620  -7.90  2.7e-15 ***
## partnerhiedu -0.282617  0.753809  0.067577  0.087659  -3.22   0.0013 **
## agefb         0.205506  1.228146  0.016829  0.019935  10.31  < 2e-16 ***
## age2         -0.003409  0.996597  0.000284  0.000324 -10.54  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##              exp(coef) exp(-coef) lower .95 upper .95
## educ.high        0.676      1.480     0.613     0.745
## partnerhiedu     0.754      1.327     0.635     0.895
## agefb            1.228      0.814     1.181     1.277
## age2             0.997      1.003     0.996     0.997
##
## Concordance= 0.544  (se = 0.006 )
## Likelihood ratio test= NA  on 4 df,   p=NA
## Wald test            = 140  on 4 df,   p=<2e-16
## Score (logrank) test = NA  on 4 df,   p=NA
##
##   (Note: the likelihood ratio and score tests assume independence of
##      observations within a cluster, the Wald and robust score tests do not).
```

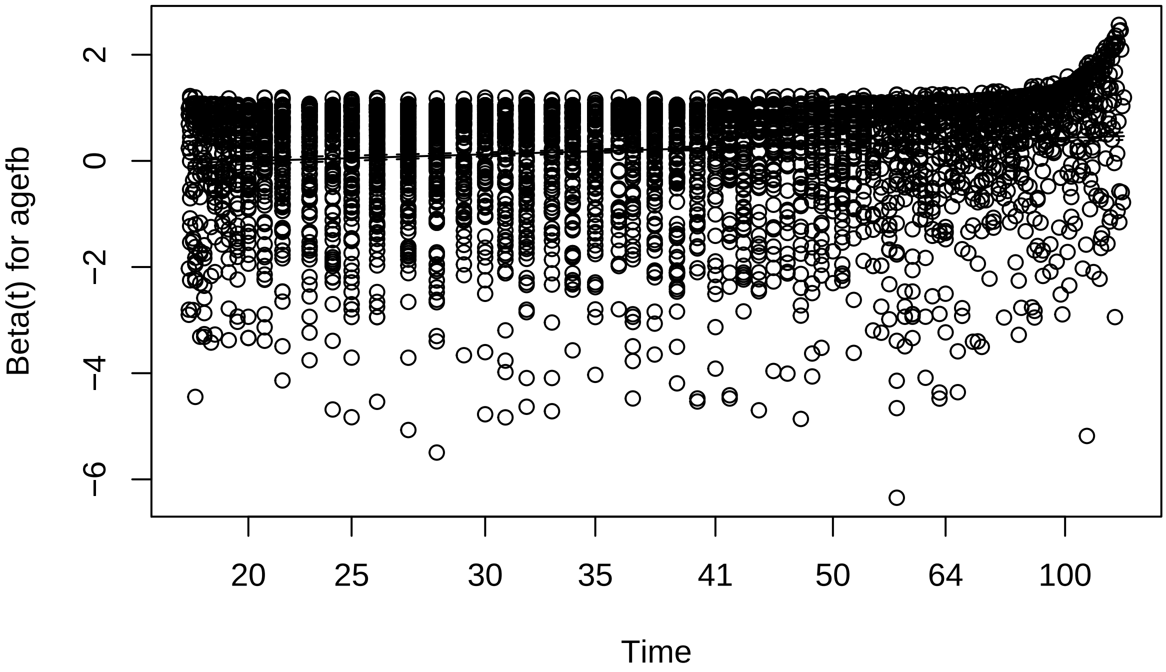
```
#Schoenfeld test
fit.test<-cox.zph(cox.s)
fit.test
```

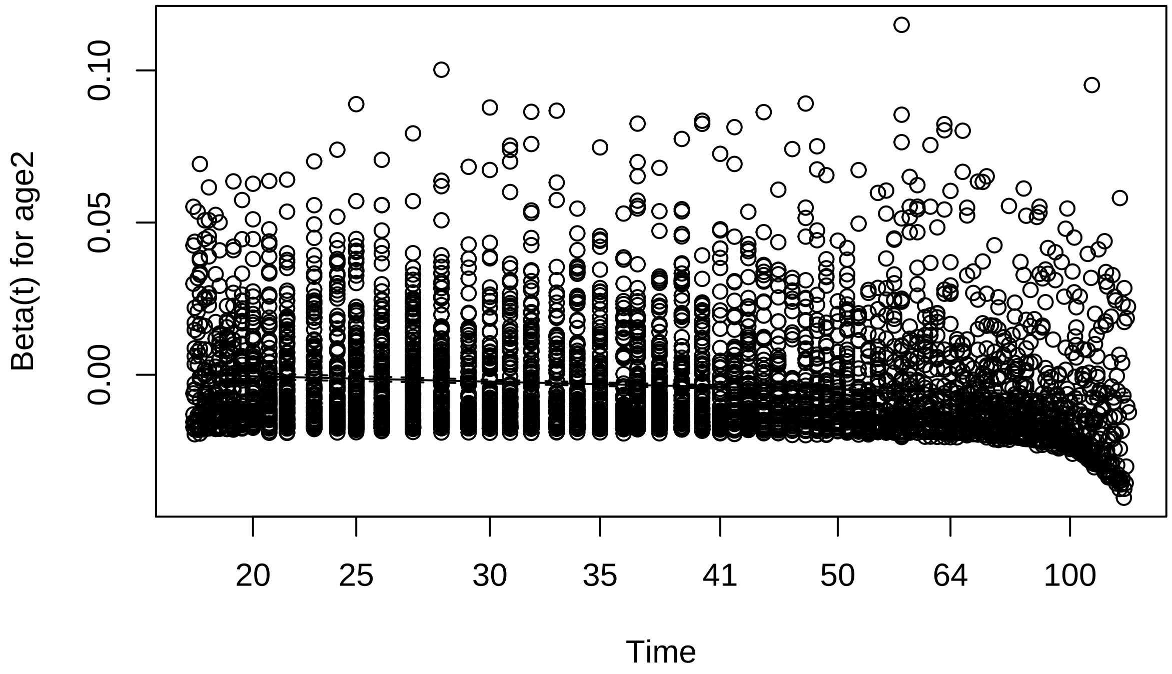
```
##                 chisq df      p
## educ.high        1.35  1   0.25
## partnerhiedu     3.28  1   0.07
## agefb          103.37  1 <2e-16
## age2            84.96  1 <2e-16
## GLOBAL         150.55  4 <2e-16
```

```
plot(fit.test, df=2)
```



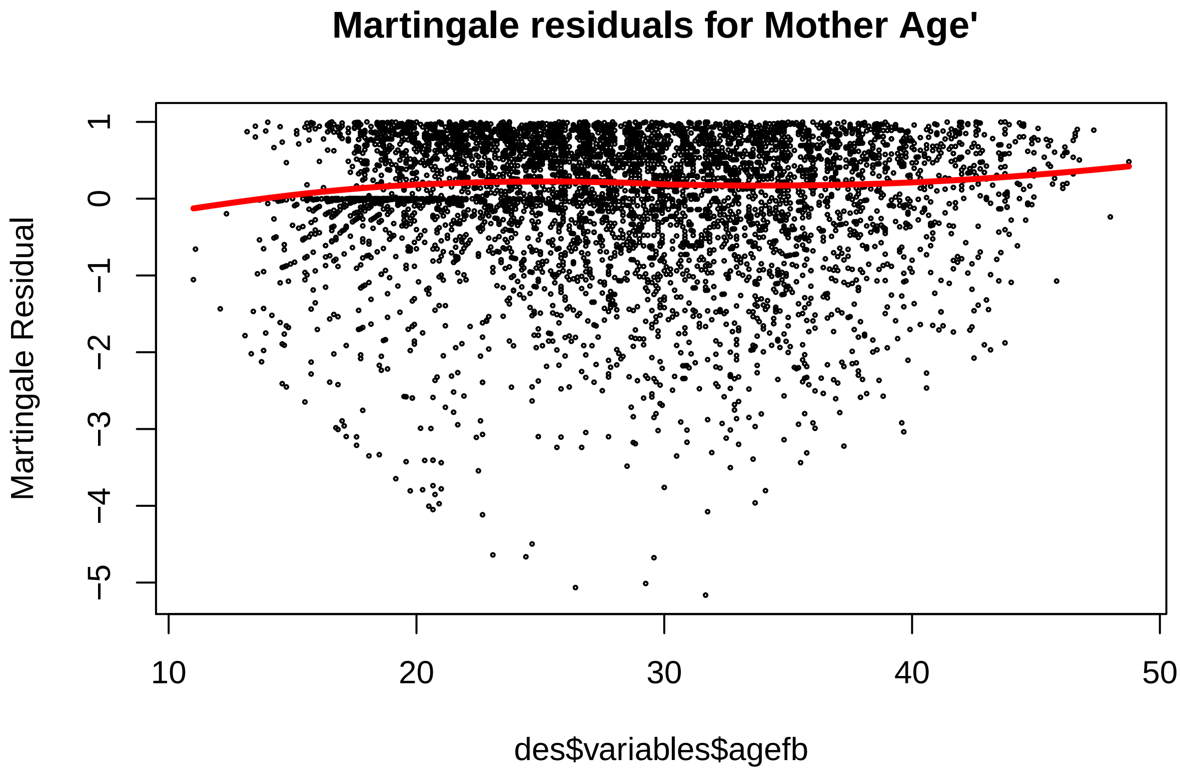

Beta(t) for partnerhiedu
20
15
10
5
0
−5
20
25
30
35
41
50
64
100
Time


Beta(t) for agefb
2
0
−2
−4
−6
20
25
30
35
41
50
64
100
Time



```
#martingale residuals
#extract Martingale residuals
res.mar<-resid(cox.s, type="martingale")

#plot vs maternal age
scatter.smooth(des$variables$agefb, res.mar,degree = 2,
               span = 1, ylab="Martingale Residual",
               col=1,  cex=.25, lpars=list(col = "red",
                                           lwd = 3))
title(main="Martingale residuals for Mother Age' ")
```



### Non-proportional effects with time

We can also include a time by covariate interaction term to model directly any time-dependence in the covariate effect. Different people say to do different things, some advocate for simply interacting time with the covariate, others say use a nonlinear function of time, e.g. log(time) $*$ the covariate, others say use time-1 $*$ covariate, which is called the "heavy side function", according to Mills. Mills cites Allison, in saying that, to interpret the heavy side function, you go with the rule : "If $\beta_2$ is positive, then the effect of the covariate x increases over time, while if $\beta_2$ is negative, the effect of x decreases over time."

```
sub.split<-survSplit(Surv(secbi, b2event)~.,
                     data= sub2[sub2$secbi>0,], cut=36, episode = "timegroup")
sub.split<-sub.split[order(sub.split$CASEID, sub.split$timegroup),]

sub.split$hv1<-sub.split$agefb*(1-sub.split$timegroup)
sub.split$hv2<-sub.split$agefb*(sub.split$timegroup)

head(sub.split, n=20)
```

```
##             CASEID int.cmc fbir.cmc sbir.cmc marr.cmc rural educ age partneredu
## 1          1  1  2    1386     1381     1349     1213     2    0  30          0
## 2          1  3  2    1386     1356       NA     1334     2    2  22          2
## 3          1  4  2    1386     1262     1230     1201     2    0  42          0
## 4          1  4  3    1386     1376     1356     1313     2    1  25          0
## 5          1  5  1    1386     1385     1352     1280     2    2  25          0
## 6          1  6  2    1386     1323     1226     1183     2    0  37          0
## 7          1  6  2    1386     1323     1226     1183     2    0  37           0
## 8          1  7  2    1386     1266       NA     1347     2    0  21           0
## 9          1  7  2    1386     1266       NA     1347     2    0  21           0
## 10         1  9  5    1386     1139     1094     1090     2    0  46           0
## 11         1  9  5    1386     1139     1094     1090     2    0  46           0
## 12         1 11  2    1386     1273     1247     1112     2    1  37           2
## 13         1 13  3    1386     1278       NA     1241     2    0  25           2
## 14         1 13  3    1386     1278       NA     1241     2    0  25           2
## 15         1 15  1    1386     1385     1368     1192     2    0  35           0
## 16         1 15  2    1386     1359       NA     1336     2    1  20           0
## 17         1 16  2    1386     1362     1313     1254     2    0  33           0
## 18         1 16  2    1386     1362     1313     1254     2    0  33           0
## 19         1 19  2    1386     1374       NA     1365     2    1  17           0
## 20         1 20  2    1386     1303     1261     1283     2    0  34           2
##    partnerage weight psu strata agefb educ.high   age2 partnerhiedu tstart
## 1          49  1.058   1     26 29.58         0  875.2            0      0
## 2          22  1.058   1     26 19.50         1  380.2            0      0
## 3          NA  1.058   1     26 31.67         0 1002.8            0      0
## 4          35  1.058   1     26 24.17         0  584.0            0      0
## 5          31  1.058   1     26 24.92         1  620.8            0      0
## 6          45  1.058   1     26 31.75         0 1008.1            0      0
## 7          45  1.058   1     26 31.75         0 1008.1            0     36
## 8          36  1.058   1     26 11.00         0  121.0            0      0
## 9          36  1.058   1     26 11.00         0  121.0            0     36
## 10         51  1.058   1     26 25.42         0  646.0            0      0
## 11         51  1.058   1     26 25.42         0  646.0            0     36
## 12         47  1.058   1     26 27.58         0  760.8            0      0
## 13         27  1.058   1     26 16.00         0  256.0            0      0
## 14         27  1.058   1     26 16.00         0  256.0            0     36
## 15         47  1.058   1     26 34.92         0 1219.2            0      0
## 16         32  1.058   1     26 17.75         0  315.1            0      0
## 17         34  1.058   1     26 31.00         0  961.0            0      0
## 18         34  1.058   1     26 31.00         0  961.0            0     36
## 19         NA  1.058   1     26 16.00         0  256.0            0      0
## 20         75  1.058   1     26 27.08         0  733.5            0      0
##    secbi b2event timegroup    hv1   hv2
## 1     32       1         1   0.00 29.58
## 2     30       0         1   0.00 19.50
## 3     32       1         1   0.00 31.67
## 4     20       1         1   0.00 24.17
## 5     33       1         1   0.00 24.92
## 6     36       0         1   0.00 31.75
## 7     97       1         2 -31.75 63.50
## 8     36       0         1   0.00 11.00
## 9    120       0         2 -11.00 22.00
## 10    36       0         1   0.00 25.42
## 11    45       1         2 -25.42 50.83
## 12    26       1         1   0.00 27.58
## 13    36       0         1   0.00 16.00
## 14   108       0         2 -16.00 32.00
## 15    17       1         1   0.00 34.92
## 16    27       0         1   0.00 17.75
## 17    36       0         1   0.00 31.00
## 18    49       1         2 -31.00 62.00
## 19    12       0         1   0.00 16.00
## 20    36       0         1   0.00 27.08
```

```
des3<-svydesign(ids=~psu, strata = ~strata ,
                weights=~weight, data=sub.split[is.na(sub.split$partnerhiedu)==F,])

cox.s2<-svycoxph(Surv(secbi,b2event)~educ.high+partnerhiedu+hv1+hv2,
                 design=des3)
summary(cox.s2)
```

```
## Stratified 1 - level Cluster Sampling design (with replacement)
## With (217) clusters.
## svydesign(ids = ~psu, strata = ~strata, weights = ~weight, data = sub.split[is.na(sub.split$partnerhi
##     F, ])
```

```
## Call:
## svycoxph(formula = Surv(secbi, b2event) ~ educ.high + partnerhiedu +
##     hv1 + hv2, design = des3)
##
##   n= 7812, number of events= 4527
##
##                  coef exp(coef) se(coef) robust se     z Pr(>|z|)
## educ.high    -0.40279   0.66845  0.04609   0.06341 -6.35  2.1e-10 ***
## partnerhiedu -0.26637   0.76616  0.06780   0.08987 -2.96    0.003 **
## hv1           0.11036   1.11668  0.00385   0.00317 34.78  < 2e-16 ***
## hv2           0.04356   1.04453  0.00233   0.00280 15.58  < 2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
##              exp(coef) exp(-coef) lower .95 upper .95
## educ.high        0.668      1.496     0.590     0.757
## partnerhiedu     0.766      1.305     0.642     0.914
## hv1              1.117      0.896     1.110     1.124
## hv2              1.045      0.957     1.039     1.050
##
## Concordance= 0.676  (se = 0.005 )
## Likelihood ratio test= NA  on 4 df,   p=NA
## Wald test            = 2792  on 4 df,   p=<2e-16
## Score (logrank) test = NA  on 4 df,   p=NA
##
##   (Note: the likelihood ratio and score tests assume independence of
##      observations within a cluster, the Wald and robust score tests do not).
```

So, for us $\beta_2$ in the heavyside function is positive, suggesting that the age effect increase over time

## Aalen's additive regression model

An alternative model proposed by Odd Aalen in 1989 and 1993 describe a model that is inherently nonparametric and models the changes in relationships in a hazard model.

```
fita<-aareg(Surv(secbi,b2event)~educ.high+partnerhiedu+agefb+age2+cluster(strata),
            sub2, weights = weight)

summary(fita)
```

```
## $table
##                      slope       coef  se(coef) robust se       z          p
## Intercept       -1.555e-02 -4.589e-04 8.678e-05 1.259e-04 -3.6435 2.689e-04
## educ.high       -1.428e-02 -1.369e-04 2.004e-05 2.005e-05 -6.8253 8.775e-12
## partnerhiedu    -8.884e-03 -8.414e-05 2.748e-05 2.576e-05 -3.2659 1.091e-03
## agefb            4.301e-03  6.367e-05 6.140e-06 8.034e-06  7.9245 2.290e-15
## age2            -7.796e-05 -1.035e-06 1.052e-07 1.228e-07 -8.4285 3.500e-17
## cluster(strata)  8.295e-05 -2.951e-07 9.343e-07 9.927e-07 -0.2973 7.663e-01
## 
## $test
## [1] "aalen"
## 
## $test.statistic
##       Intercept       educ.high    partnerhiedu           agefb            age2 
##          -25.33         -201.26          -60.03          743.33       -42444.86 
## cluster(strata) 
##         -152.22 
## 
## $test.var
##           b0                                                  
## b0    22.940   -22.14    4.963    -330.53    19087   -437.8
##      -22.137   867.98 -145.254     199.45    -7901  -1266.0
##        4.963  -145.25  384.492     -78.66     4830   -857.5
##     -330.531   199.45  -78.662    5139.27  -305542    696.0
##    19087.171 -7900.88 4830.404 -305542.16 18606189 -34790.5
##     -437.837 -1265.96 -857.481     696.00   -34791 232265.1
## 
## $test.var2
##          [,1]      [,2]     [,3]      [,4]     [,5]     [,6]
## [1,]    48.31    -62.69    60.35    -626.5    32561   -414.6
## [2,]   -62.69    869.53  -161.46     697.5   -27466  -2499.6
## [3,]    60.35   -161.46   337.86    -801.6    38475    677.9
## [4,]  -626.52    697.45  -801.64    8798.5  -467332  -4376.9
## [5,] 32560.82 -27465.57 38474.69 -467332.0 25359756 383455.6
## [6,]  -414.56  -2499.59   677.95   -4376.9   383456 262224.1
## 
## $chisq
##       [,1]
## [1,] 149.1
## 
## $n
## [1] 5289  171  171
## 
## attr(,"class")
## [1] "summary.aareg"
```

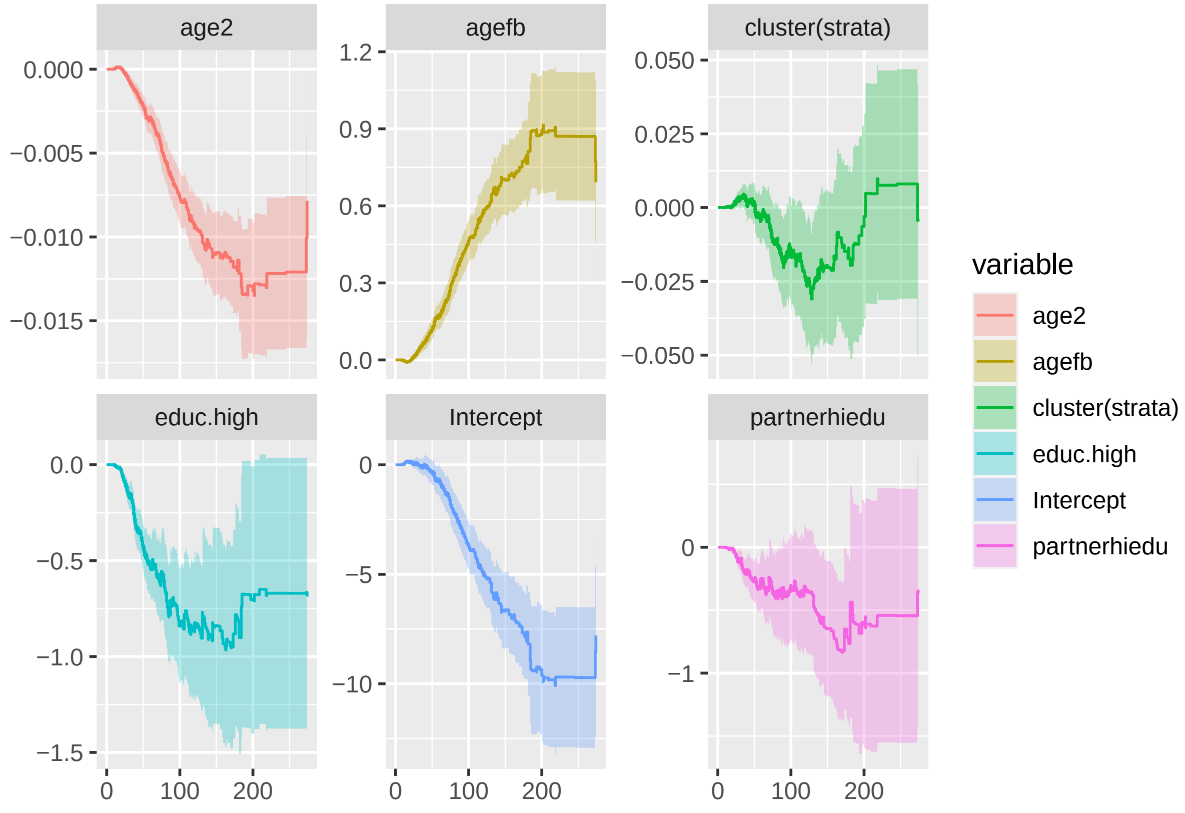
```
library(ggfortify)
```

```
## Loading required package: ggplot2
```

```
autoplot(fita)
```

```
## Warning: ‘group_by_()‘ is deprecated as of dplyr 0.7.0.
## Please use ‘group_by()‘ instead.
## See vignette('programming') for more help
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_warnings()‘ to see where this warning was generated.
```

```
## Warning: ‘mutate_()‘ is deprecated as of dplyr 0.7.0.
## Please use ‘mutate()‘ instead.
## See vignette('programming') for more help
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_warnings()‘ to see where this warning was generated.
```



What is seen in the plots are the time-varying coefficients of the hazard model. For example the effect of `educ.high` is globally negative, suggesting higher education decreases the hazard, as we saw in the Cox model above. In the plot, the regression function initially decreases sharply but then plateaus, suggesting the education effect is really only time varying until about 100 months after the first birth.