# DEM 7223 - Event History Analysis - Models of Frailty

true

October26, 2020

## Contents



## Notes

### Unobserved Heterogeneity

- Often in hazards analysis we are faced with the possibility that we may not be able to measure directly all of the factors that could influence an individual's (or a group's) hazard/duration time
- This is the idea of unobserved heterogeneity (or variance)
- If there are factors inherent to the individual that makes him/her more/less likely to experience the event, then we must be prepared to examine this possibility in our analysis

#### Levels of heterogeneity

#### Heterogeneity == Frailty

- In hazards models we typically deal with unobserved heterogeneity through the concept of frailty
- Frailty is the idea that some individuals (or groups) have an inherently higher hazard rate (although we cannot directly measure why) leading to that individual/or group to have shorter duration time (higher risk==high frailty: lower risk==low frailty)

# Sastry 1997

| Level | Genetic | Behavioral | Environmental |
|---|---|---|---|
| Individual | Idiosyncratic genetic factors | individual specific behavioral factors and care | |
| Family | genetic factors among siblings/parents | Parental competence, care of children common among siblings | Household environment |
| Community | | Shared preferences/cultural values | Infrastructure, climate, physical environment |

Figure 1: Sastry 1997

**Frailty in populations**

- Consider a population composed of 2 (or more) sub-populations, with the same size and with constant mortality rates, $\mu_1$ and $\mu_2$
- If $\mu_1$ and $\mu_2$ are not equal, then the one with the higher rate can be consider to have higher *frailty* compared to the other.
- After x time periods, there will be:

$e^{-\mu_1 x}$ and $e^{-\mu_2 x}$ remaining individuals in the two sub-populations

- Since the two groups started with the same population size, the initial mortality rate in the total population will be the mean of the two rates:

$$\mu = \frac{\mu_1 + \mu_2}{2}$$

and the overall mortality rate at each time x will be:

$$\mu_{total} = \frac{e^{-\mu_1 x}\mu_1 + e^{-\mu_2 x}\mu_2}{e^{-\mu_1 x} + e^{-\mu_2 x}}$$

* Which is less than the initial rate, because the population with the higher rate of mortality will be smaller at time x, because of it's higher rate of mortality.

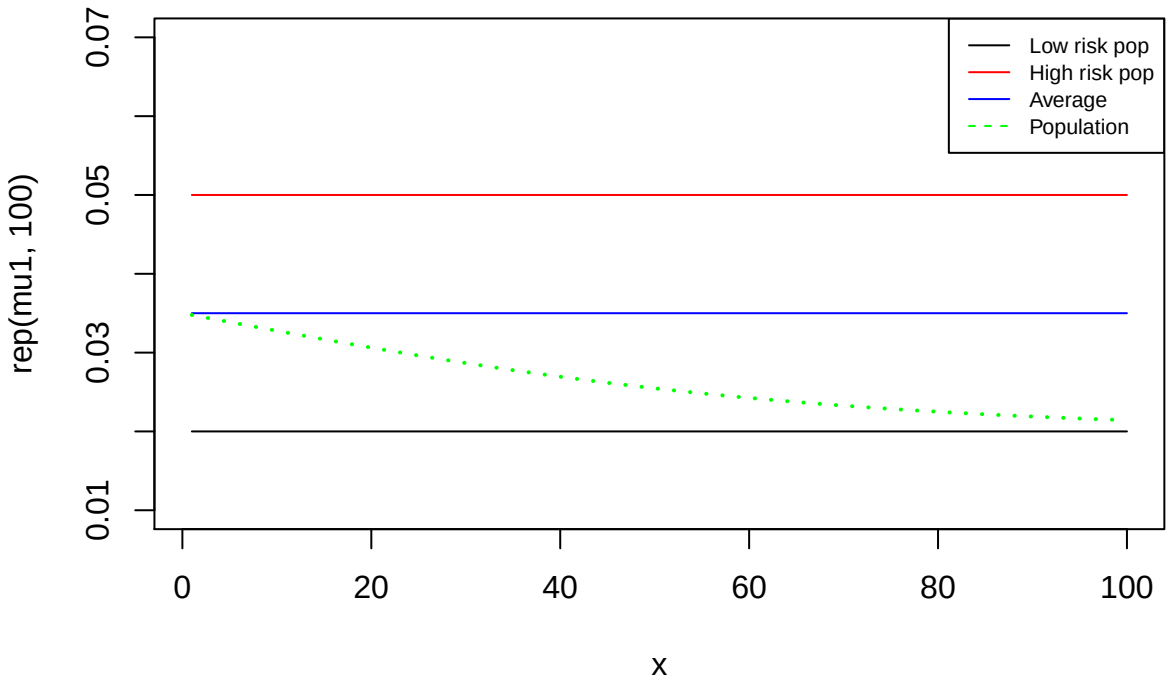
- The population will show a decreasing mortality rate over time, because of the increased death rate of the group with the higher frailty.
- This is called *differential frailty*

```
x<-1:100
n1<-n2<-100
mu1<-.02; mu2<-.05

mrate<-mean(c(mu1, mu2))

trate<-((exp(-mu1*x)*mu1)+(exp(-mu2*x)*mu2))/(exp(-mu1*x)+exp(-mu2*x))

plot(y=rep(mu1, 100), x=x, type="l" ,ylim=c(.01, .07))
lines(y=rep(mu2, 100), x=x,  col="red")
lines(y=rep(mrate, 100), x=x, col="blue")
lines(trate, x=x, lty=3, col="green", lwd=2)
legend("topright",
      legend= c("Low risk pop", "High risk pop", "Average", "Population"),
      col=c("black", "red", "blue", "green"),
      lty=c(1,1,1,2),
      cex=.7)
```




* This example shows the population-level effect of not controlling for the possibility of individual or group difference in frailty

- If we leave off this possibility, our observed rates may not be correct

## Levels of frailty

There are generally two ways of incorporating frailty in models. These correspond to two assumed levels at which frailty can operate: Individual and Shared.

### Individual frailty

- If we assume individual-based frailty, the we assume that that unmeasured heterogeneity affects an individual's risk of experiencing the event in question. This modifies the hazard function to be:

$$h_i(t) = h_0 exp(x'\beta + w_i)$$

Where the $w_i \sim Normal(0, \sigma_w)$. If $\sigma_w = 0$ then we have the standard proportional hazards model.

This implies that frailty contributes an independent, additive term to the linear mean function for the proportional hazards model.

### Shared frailty

Similarly to individual frailty, *shared frailty* likewise introduces an extra term in the hazard model. Instead of being specific to each individual, the term is specific to each sub-group in our analysis. These sub-groups can be communities, families or schools. The model is then:

$$h_i(t) = h_0 exp(x'\beta + u_j)$$

Where the $u_j \sim Normal(0, \sigma_u)$. If $\sigma_u = 0$ then we have the standard proportional hazards model.

This model specifies an additive increase or decrease in the mean function for the PH model, where individuals within communities with positive frailty values face higher risk of experiencing the event in question, and vice versa.

### Shared frailty is a multi-level model

- When a shared frailty model is used, we are basically specifying a multi-level model
- This means, you can include individual and group-level predictors, and we can also incorporate structure to the groups to measure spatial or temporal correlations in risk.

## Examples

This example will illustrate how to fit the extended Cox Proportional hazards model with Gaussian frailty to continuous duration data (i.e. person-level data) and a discrete-time (longitudinal) data set. In this example, I will use 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 birth history recode file. This file contains information for all births to women in the survey.

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

```
#Load required libraries
library(foreign)
library(survival)
library(car)
library(survey)
library(coxme)
library(knitr)
library(lme4)
library(car)
opts_chunk$set(tidy.opts=list(width.cutoff=60),tidy=TRUE)
```

```
# load the data
model.dat <- readRDS("~/OneDrive - University of Texas at San Antonio/classes/dem7223/dem7223_20/data/h
names(model.dat) <- tolower(names(model.dat))
```

```
# We form a subset of variables
sub <- data.frame(CASEID = model.dat$caseid, v008 = model.dat$v008,
    bord = model.dat$bidx, csex = model.dat$b4, b2 = model.dat$b2,
    b3 = model.dat$b3, b5 = model.dat$b5, b7 = model.dat$b7,
    ibint = model.dat$b11, rural = model.dat$v025, educ = model.dat$v106,
```

```
    age = model.dat$v012, partneredu = model.dat$v701, partnerage = model.dat$v730,
    hhses = model.dat$v190, weight = model.dat$v005/1e+06, psu = model.dat$v021,
    strata = model.dat$v022, region = model.dat$v023)

sub$death.age <- ifelse(sub$b5 == 1, ((((sub$v008)) + 1900) -
    (((sub$b3)) + 1900)), sub$b7)

# censoring indicator for death by age 5, in months (<=60
# months)
sub$d.event <- ifelse(is.na(sub$b7) == T | sub$b7 > 60, 0, 1)
sub$d.eventfac <- factor(sub$d.event)
levels(sub$d.eventfac) <- c("Alive at Age 5", "Dead by Age 5")
table(sub$d.eventfac)
```

```
##
## Alive at Age 5  Dead by Age 5
##          25339           2470
```

```
# recodes
sub$male <- ifelse(sub$csex == 1, 1, 0)
sub$educ.high <- ifelse(sub$educ %in% c(2, 3), 1, 0)
sub$age2 <- sub$age^2
sub$partnerhiedu <- ifelse(sub$partneredu < 3, 0, ifelse(sub$partneredu %in%
    c(8, 9), NA, 1))
```

### Fit the ordinary Cox model

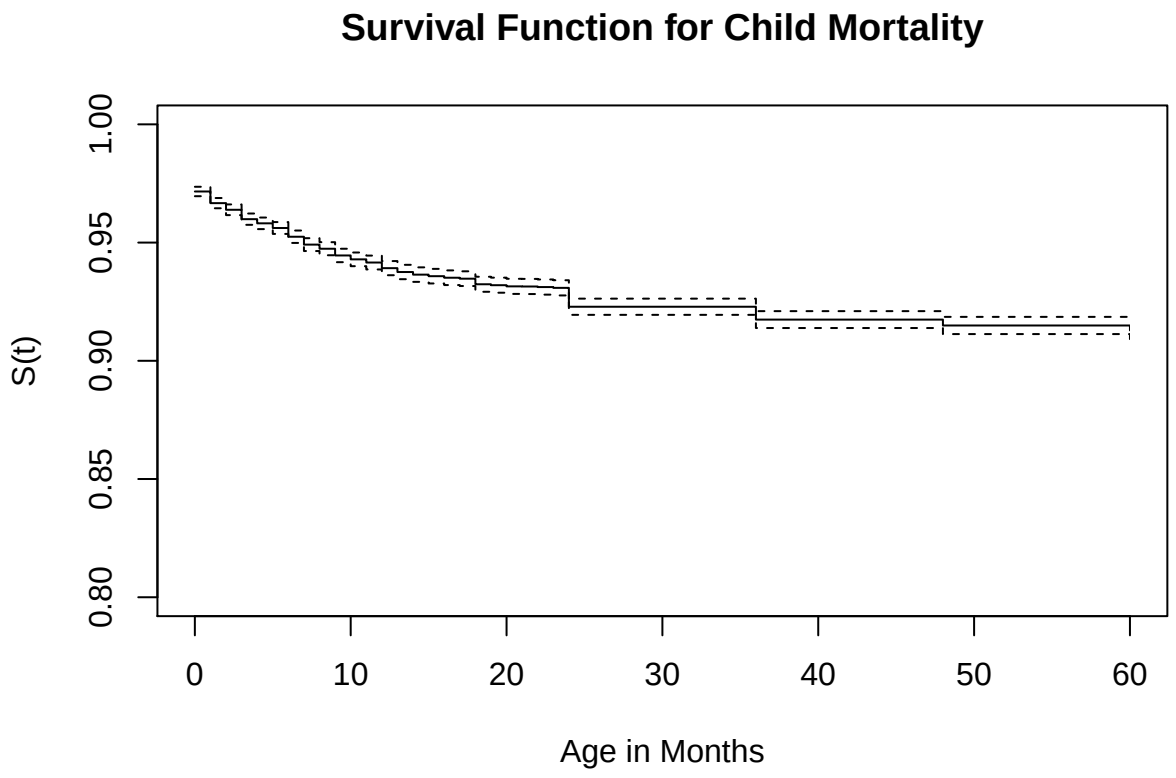
Here I fit the ordinary Cox model without frailty, just for comparison sake.

```
# using coxph in survival library
fit.cox2 <- coxph(Surv(death.age, d.event) ~ bord + male + educ.high +
    I(age/5) + I(hhses > 3), data = sub, weights = weight)
summary(fit.cox2)
```

```
## Call:
## coxph(formula = Surv(death.age, d.event) ~ bord + male + educ.high +
##     I(age/5) + I(hhses > 3), data = sub, weights = weight)
##
##   n= 27809, number of events= 2470
##
##                       coef exp(coef)  se(coef) robust se      z Pr(>|z|)
## bord              0.150384  1.162280  0.009397  0.011103 13.545  < 2e-16 ***
## male              0.141260  1.151724  0.041176  0.048359  2.921  0.00349 **
## educ.high        -0.471578  0.624017  0.061558  0.074134 -6.361    2e-10 ***
## I(age/5)         -0.054422  0.947033  0.015080  0.019457 -2.797  0.00516 **
## I(hhses > 3)TRUE  0.122010  1.129766  0.050040  0.063999  1.906  0.05659 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##                  exp(coef) exp(-coef) lower .95 upper .95
## bord                 1.162     0.8604    1.1373    1.1878
## male                 1.152     0.8683    1.0476    1.2662
```

```
## educ.high            0.624     1.6025    0.5396    0.7216
## I(age/5)             0.947     1.0559    0.9116    0.9838
## I(hhses > 3)TRUE     1.130     0.8851    0.9966    1.2808
##
## Concordance= 0.611  (se = 0.007 )
## Likelihood ratio test= 388.1  on 5 df,   p=<2e-16
## Wald test            = 311.2  on 5 df,   p=<2e-16
## Score (logrank) test = 426  on 5 df,   p=<2e-16,   Robust = 239  p=<2e-16
##
##   (Note: the likelihood ratio and score tests assume independence of
##      observations within a cluster, the Wald and robust score tests do not).
```

```
plot(survfit(fit.cox2), ylim = c(0.8, 1), xlim = c(0, 60), ylab = "S(t)",
    xlab = "Age in Months")
title(main = "Survival Function for Child Mortality")
```




## Fit the Cox model with frailty at the regional level

The `coxme()` function in the `coxme` library Link will fit the Cox model with shared frailty, assuming a Gaussian frailty term. The model would look like:

$$h_j(t) = h_{0j} e^{(x'\beta + u_j)}$$

where

$$u_j \sim N(0, \sigma^2)$$

is a Normally distributed random effect, identical for each person in the jth group. This term raises or lowers the average hazard function the same way for each person within each group, but allows the overall risk for people in different groups to be different. This would be considered to be a random intercept model, if we were considering an ordinary linear or generalized linear model.

```
fit.cox.f <- coxme(Surv(death.age, d.event) ~ bord + male + educ.high +
    I(age/5) + I(hhses > 3) + (1 | region), data = sub, weights = weight)
summary(fit.cox.f)
```

```
## Cox mixed-effects model fit by maximum likelihood
##   Data: sub
##   events, n = 2470, 27809
##   Iterations= 7 38
##                     NULL Integrated    Fitted
## Log-likelihood -23991.86  -23759.08 -23734.05
##
##                    Chisq   df p    AIC   BIC
## Integrated loglik 465.57  6.0 0 453.57 418.7
##  Penalized loglik 515.64 22.3 0 471.03 341.4
##
## Model:  Surv(death.age, d.event) ~ bord + male + educ.high + I(age/5) +      I(hhses > 3) + (1 | reg
## Fixed coefficients
##                          coef exp(coef)    se(coef)     z       p
## bord              0.14545326 1.1565637 0.009423884 15.43 0.0e+00
## male              0.14377160 1.1546204 0.041185309  3.49 4.8e-04
## educ.high        -0.47186815 0.6238358 0.061800269 -7.64 2.3e-14
## I(age/5)         -0.05046630 0.9507860 0.015082703 -3.35 8.2e-04
## I(hhses > 3)TRUE  0.05258302 1.0539901 0.057833748  0.91 3.6e-01
##
## Random effects
##  Group  Variable  Std Dev   Variance
##  region Intercept 0.2088971 0.0436380
```

This gives us the variance in child mortality by region, which is honestly pretty substantial. We can use a likelihood ratio test to see if the frailty model is significantly better than the ordinary Cox model:

```
anova(fit.cox.f, fit.cox2)
```

```
## Analysis of Deviance Table
##  Cox model: response is  Surv(death.age, d.event)
##  Model 1: ~bord + male + educ.high + I(age/5) + I(hhses > 3) + (1 | region)
##  Model 2: ~bord + male + educ.high + I(age/5) + I(hhses > 3)
##   loglik  Chisq Df P(>|Chi|)
## 1 -23759
## 2 -23798 77.447  1 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
AIC(fit.cox.f)
```

```
## [1] 47512.7
```

```
AIC(fit.cox2)
```

```
## [1] 47605.6
```

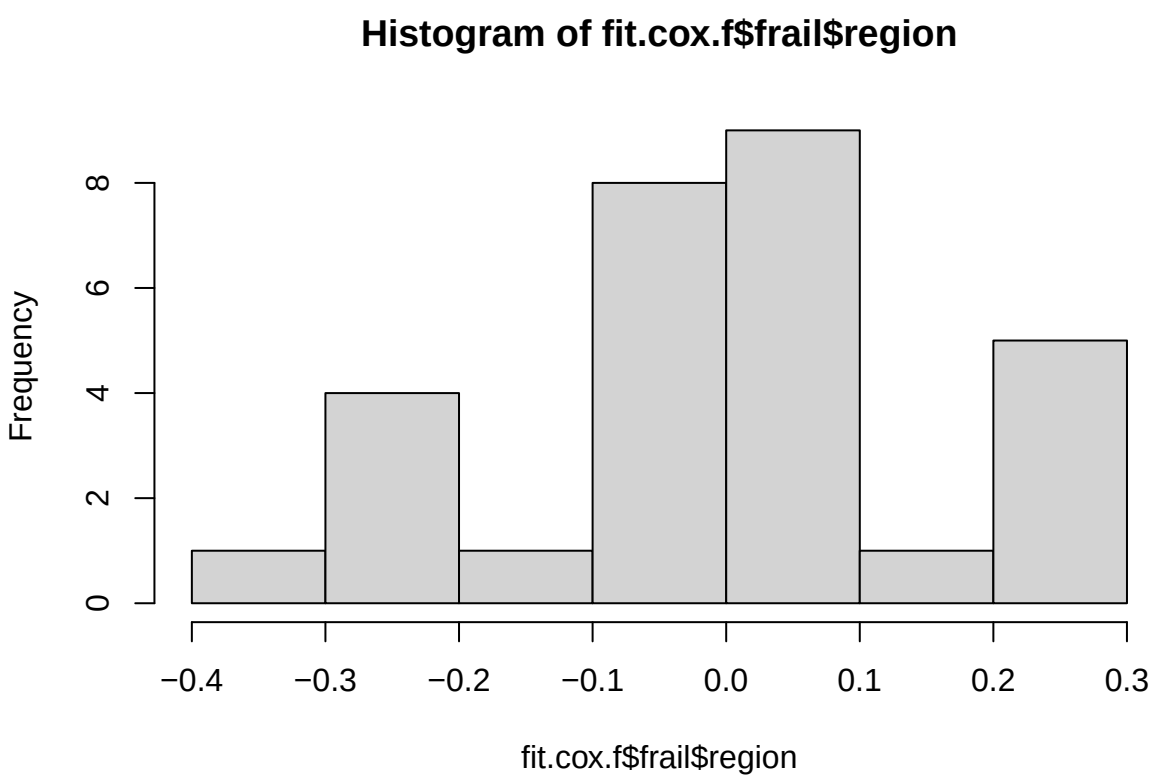
Which it is, and this is supported by the AIC difference between the two models of 92.9 points.

So, what are the frailties in this case? We can get those from the `frail` portion of the model structure:

```
fit.cox.f$frail
```

```
## $region
##            1            2            3            4            5            6
##  0.203360878  0.016819290  0.036654608  0.217173945  0.275839793 -0.093426618
##            7            8            9           10           11           12
## -0.358978732  0.121715532  0.235113399 -0.045783692 -0.282862148  0.003879599
##           13           14           15           16           17           18
## -0.010011234 -0.086047981 -0.230403270  0.045897537  0.087466014 -0.003997729
##           19           20           21           22           23           24
## -0.215003292  0.020788429 -0.092393569 -0.103155497  0.294171335  0.070191103
##           25           26           27           28           29
## -0.225918492  0.075830696 -0.013663474  0.098451995 -0.041708427
```

```
hist(fit.cox.f$frail$region)
```




Which shows the region region.23 has the highest frailty, which means that the average level of childhood mortality is highest in that region, while the lowest frailty is in region.7.

### Random slopes

If we were interested in whether a predictor variable had heterogeneous effects across the various groups within our data, we could include that in our model as well, and we would have effectively a random slope model:

$$h_j(t) = h_{0j} e^{(x'\beta + u_j + \gamma_j' x)}$$

where $\gamma_j$ is a group-specific effect of a particular predictor variable, and these two random effects will be distributed as:

$$\begin{bmatrix} u_j \\ \gamma_j \end{bmatrix} \sim \text{MVN}(0, \Sigma)$$

```
# See if higher birth order children face equal disadvantage
# in all regions
fit.cox.f2 <- coxme(Surv(death.age, d.event) ~ bord + male +
    educ.high + I(age/5) + I(hhses > 3) + (1 + bord | region),
    data = sub, weights = weight)
summary(fit.cox.f2)
```

```
## Cox mixed-effects model fit by maximum likelihood
##   Data: sub
##   events, n = 2470, 27809
##   Iterations= 13 68
##                     NULL Integrated    Fitted
## Log-likelihood -23991.86   -23754.5 -23723.86
##
##                    Chisq    df p    AIC    BIC
## Integrated loglik 474.72  8.00 0 458.72 412.23
##  Penalized loglik 536.00 26.77 0 482.46 326.85
##
## Model:  Surv(death.age, d.event) ~ bord + male + educ.high + I(age/5) +      I(hhses > 3) + (1 + bor
## Fixed coefficients
##                          coef exp(coef)   se(coef)     z       p
## bord              0.15914283 1.1725054 0.01197622 13.29 0.0e+00
## male              0.14214578 1.1527447 0.04119280  3.45 5.6e-04
## educ.high        -0.46950097 0.6253142 0.06188703 -7.59 3.3e-14
## I(age/5)         -0.05170775 0.9496063 0.01509542 -3.43 6.1e-04
## I(hhses > 3)TRUE  0.06891910 1.0713495 0.05820030  1.18 2.4e-01
##
## Random effects
##  Group  Variable  Std Dev       Variance      Corr
##  region Intercept  0.2894157608  0.0837614826 -0.8276868110
##         bord       0.0289425343  0.0008376703
```

```
anova(fit.cox.f, fit.cox.f2)
```

```
## Analysis of Deviance Table
##  Cox model: response is  Surv(death.age, d.event)
##  Model 1: ~bord + male + educ.high + I(age/5) + I(hhses > 3) + (1 | region)
##  Model 2: ~bord + male + educ.high + I(age/5) + I(hhses > 3) + (1 + bord |      region)
##    loglik  Chisq Df P(>|Chi|)
## 1 -23759
## 2 -23755 9.1528  2   0.01029 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

And it looks like there is a significant regional variation in this effect, because the model with the additional term fits the data significantly better than the model with only the random "intercept".

## 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 First we load our data

```
eclskk5 <- readRDS("C:/Users/ozd504/OneDrive - University of Texas at San Antonio/classes/dem7223/dem72
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 months. So, I fix
# the age at the midpoint of the interval they give, and make
# it into years by dividing by 12
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='nh
    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
```

```
# make an id that is the combination of state and strata
e.long$sampleid <- paste(e.long$w6c6p_6str, e.long$w6c6p_6psu)
# within each sampling unit, sum the weights
wts <- tapply(e.long$w6c6p_20, e.long$sampleid, sum)
# make a data frame from this
wts <- data.frame(id = names(unlist(wts)), wt = unlist(wts))
# get the unique sampling location ids'
t1 <- as.data.frame(table(e.long$sampleid))
# put all of this into a data set
wts2 <- data.frame(ids = wts$id, sumwt = wts$wt, jn = t1$Freq)
```

```r
# merge all of this back to the original data file
e.long <- merge(e.long, wts2, by.x = "sampleid", by.y = "ids",
    all.x = T)
# In the new data set, multiply the original weight by the
# fraction of the sampling unit total population each person
# represents
e.long$swts <- e.long$w6c6p_20 * (e.long$jn/e.long$sumwt)
```

### 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.

```r
# Fit the model
library(survival)
fitl1 <- coxph(Surv(time = time, event = povtran) ~ mlths + mgths +
    race_rec, data = e.long, weights = swts)
summary(fitl1)
```

```
## Call:
## coxph(formula = Surv(time = time, event = povtran) ~ mlths +
##     mgths + race_rec, data = e.long, weights = swts)
##
##   n= 3938, number of events= 221
##    (57 observations deleted due to missingness)
##
##                     coef exp(coef) se(coef) robust se      z Pr(>|z|)
## mlths             0.4842    1.6229   0.1864    0.2150  2.253  0.02429 *
## mgths            -1.2299    0.2923   0.1544    0.1826 -6.736 1.63e-11 ***
## race_rechispanic  0.9508    2.5877   0.1700    0.2207  4.308 1.65e-05 ***
## race_recnhasian   0.3127    1.3671   0.3439    0.3391  0.922  0.35642
## race_recnhblack   0.9279    2.5293   0.2291    0.2805  3.308  0.00094 ***
## race_recother     0.4838    1.6222   0.2929    0.3427  1.412  0.15801
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##                  exp(coef) exp(-coef) lower .95 upper .95
## mlths               1.6229     0.6162    1.0649    2.4732
## mgths               0.2923     3.4208    0.2044    0.4181
## race_rechispanic    2.5877     0.3864    1.6790    3.9882
## race_recnhasian     1.3671     0.7314    0.7033    2.6574
## race_recnhblack     2.5293     0.3954    1.4595    4.3830
## race_recother       1.6222     0.6165    0.8288    3.1751
##
## Concordance= 0.746  (se = 0.02 )
## Likelihood ratio test= 205.6  on 6 df,   p=<2e-16
## Wald test            = 192.7  on 6 df,   p=<2e-16
## Score (logrank) test = 284.4  on 6 df,   p=<2e-16,   Robust = 96.38  p=<2e-16
##
##   (Note: the likelihood ratio and score tests assume independence of
##      observations within a cluster, the Wald and robust score tests do not).
```

## Cox model with additive frailty

Now we fit the Cox model and doing frailty by the school the child attends. I use the weights calculated above, standardized to the within cluster sample size.

```
library(coxme)
# Fit the model
fitl2 <- coxme(Surv(time = time, event = povtran) ~ mlths + mgths +
    race_rec + (1 | s2_id), e.long, weights = swts)
summary(fitl2)
```

```
## Cox mixed-effects model fit by maximum likelihood
##   Data: e.long
##   events, n = 221, 3938 (57 observations deleted due to missingness)
##   Iterations= 7 47
##                     NULL Integrated    Fitted
## Log-likelihood -1852.682  -1739.942 -1649.332
##
##                    Chisq    df p    AIC    BIC
## Integrated loglik 225.48  7.00 0 211.48 187.69
##  Penalized loglik 406.70 86.68 0 233.34 -61.20
##
## Model:  Surv(time = time, event = povtran) ~ mlths + mgths + race_rec +      (1 | s2_id)
## Fixed coefficients
##                         coef exp(coef)  se(coef)     z       p
## mlths              0.4341737 1.5436870 0.2057034  2.11 3.5e-02
## mgths             -1.0785657 0.3400829 0.1628668 -6.62 3.5e-11
## race_rechispanic   0.9737078 2.6477435 0.1871651  5.20 2.0e-07
## race_recnhasian    0.4159912 1.5158725 0.3746802  1.11 2.7e-01
## race_recnhblack    0.9314125 2.5380917 0.2510170  3.71 2.1e-04
## race_recother      0.3648314 1.4402711 0.3346136  1.09 2.8e-01
##
## Random effects
##  Group Variable  Std Dev   Variance
##  s2_id Intercept 0.7655108 0.5860068
```

```
anova(fitl1, fitl2)
```

```
## Analysis of Deviance Table
##  Cox model: response is  Surv(time = time, event = povtran)
##  Model 1: ~mlths + mgths + race_rec
##  Model 2: ~mlths + mgths + race_rec + (1 | s2_id)
##    loglik  Chisq Df P(>|Chi|)
## 1 -1749.9
## 2 -1739.9 19.918  1 8.083e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

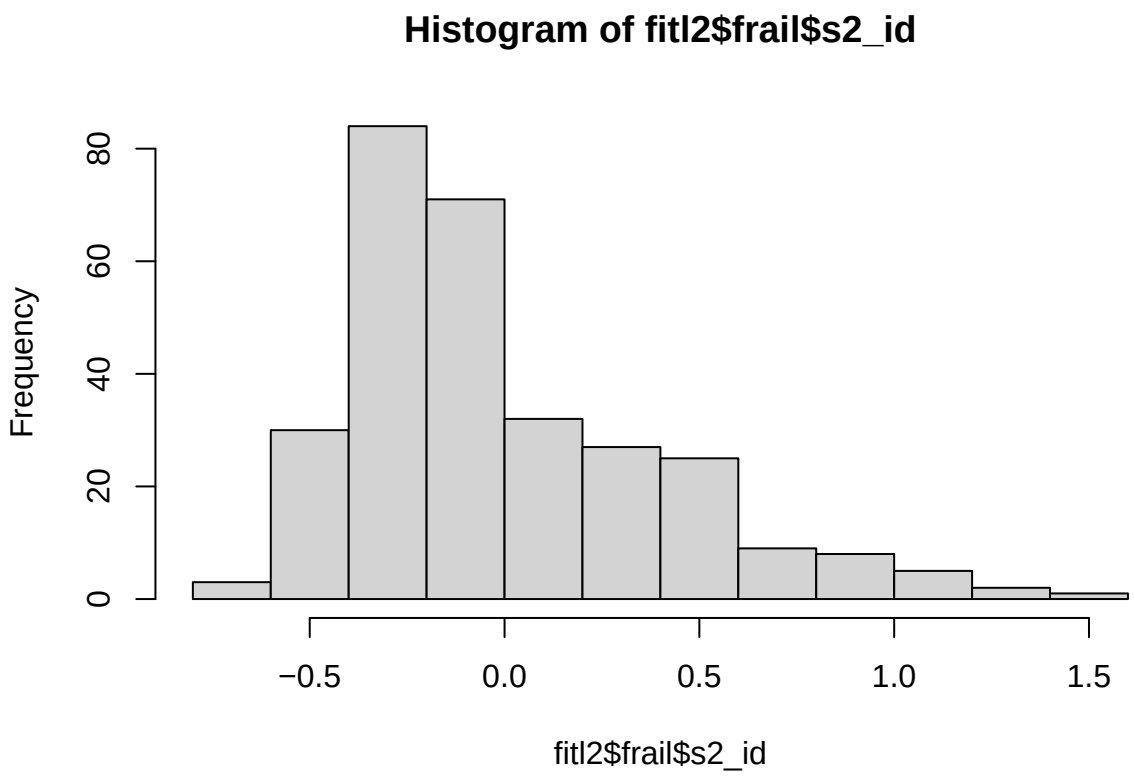
```
AIC(fitl1)
```

```
## [1] 3511.802
```

```
AIC(fitl2)
```

```
## [1] 3472.021
```

```
hist(fitl2$frail$s2_id)
```




So, we see that the frailty model has a large variance, and that the AIC is lower than the ordinary Cox model, suggesting better model fit. The model likelihood ratio test also confirms that the frailty model fits better.

# Discrete time frailty model

## Basic Discrete time model

First, we fit the basic discrete time model for our outcome. Here I'm going to use the new standardized weights I created above in a regular `glm()` model:

```
fit1 <- glm(povtran ~ as.factor(time) + mlths + mgths + race_rec -
    1, data = e.long, weights = swts, family = binomial(link = "cloglog"))
```

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

```
arm::display(fit1, detail = T)
```

```
## glm(formula = povtran ~ as.factor(time) + mlths + mgths + race_rec -
##     1, family = binomial(link = "cloglog"), data = e.long, weights = swts)
##                   coef.est coef.se z value Pr(>|z|)
## as.factor(time)1  -2.34     0.17  -14.12    0.00
## as.factor(time)2  -2.90     0.18  -15.81    0.00
## mlths              0.46     0.19    2.47    0.01
## mgths             -1.22     0.15   -7.89    0.00
## race_rechispanic   0.95     0.17    5.60    0.00
## race_recnhasian    0.31     0.34    0.89    0.37
## race_recnhblack    0.92     0.23    4.02    0.00
## race_recother      0.49     0.29    1.65    0.10
## ---
##   n = 3938, k = 8
##   residual deviance = 1537.7, null deviance = 7630.8 (difference = 6093.1)
```

### Discrete time model with shared frailty

For the discrete time model, if the logit link is used, then we are effectively fitting a multilevel model for our outcome. The model with only a group frailty component will have the exact same form as the multilevel logit model with a random intercept at the group level: Fit the basic random intercept model using the complementary log-log link function:

$$log(-log(1 - h(t))) = \beta_{0j} + x'\beta + Z\gamma'$$

with

$\beta_{0j} = \beta_0 + Z\gamma' + u_j$

and

$u_j \sim N(0, \sigma_u^2)$

Where the intercepts ($u_j$) for each group vary randomly around the overall mean ($\beta_0$).

The individual level predictors are incorporated into $x$, while the group level predictors (if any are measured) are included in $Z$. If only a random intercept is specified, then $Z$ is a vector of 1's.

```
# this will take a lot longer to fit compared to the ordinary
# logit I also had to include some extra optimization details
# because the default maximizer didn't return a satisfactory
# model fit criteria.

fit2 <- glmer(povtran ~ as.factor(time) + mlths + mgths + race_rec -
    1 + (1 | s2_id), data = e.long, weights = swts, family = binomial(link = "cloglog"),
    control = glmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 2e+05)))
```

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

```
arm::display(fit2, detail = T)
```

```
## glmer(formula = povtran ~ as.factor(time) + mlths + mgths + race_rec -
##     1 + (1 | s2_id), data = e.long, family = binomial(link = "cloglog"),
##     control = glmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 2e+05)),
##     weights = swts)
##                  coef.est coef.se z value Pr(>|z|)
## as.factor(time)1  -2.62     0.20  -12.80    0.00
## as.factor(time)2  -3.13     0.21  -14.56    0.00
## mlths              0.40     0.21    1.93    0.05
## mgths             -1.11     0.17   -6.62    0.00
## race_rechispanic   0.99     0.19    5.27    0.00
## race_recnhasian    0.49     0.40    1.21    0.23
## race_recnhblack    0.93     0.25    3.67    0.00
## race_recother      0.41     0.34    1.20    0.23
##
## Error terms:
##  Groups   Name        Std.Dev.
##  s2_id    (Intercept) 0.73
##  Residual             1.00
## ---
## number of obs: 3938, groups: s2_id, 297
## AIC = 1511.5, DIC = 1219.8
## deviance = 1356.6
```

AIC comparison of the two models:

```
AIC(fit1)  #Regular GLM
```

```
## [1] 1638.724
```

```
AIC(fit2)  #GLMM
```

```
## [1] 1511.47
```