Lab 4 (707) - Time to Event Analysis

2022-06-01

## LAB MATERIALS

### Lab 4 Goals

* Describe survival time data utilizing summary statistics and graphical methods
* Fit Cox proportional hazards regression models with fixed and time-varying covariates
* Assess appropriateness of the assumptions of the Cox model
* Calculate incidence rates and ratios using Poisson methods

### Lab 4 Grading scheme

| Competency | Points |
| --- | --- |
| .Rmd file runs without error | 10 |
| Table 1 | 5 |
| Table 2 | 12 |
| Table 3 | 8 |
| Table 4 | 15 |
| Figures 1 - 10 | 30 (3 points each) |
| Short answer questions | 20 (5 points each) |
| **Total** | **100** |

### Data and assignment

The assignment and dataset are both available on [Sakai](https://youtu.be/dQw4w9WgXcQ)

**NOTE ON THE DATASET:** As you load the data into R, be sure to notice that this is the *original*, DHS dataset, with 22,534 observations (i.e. from before we closed the cohort).

### Packages

* {tidyverse}
* {survival}
* {survminer}
* {epiR}
* {fmsb}
* {biostat3}

## Survival Analysis

The methods we have used so far have assumed a closed cohort—no loss to follow-up or censoring. In the event that we have loss to follow up (or even if we don’t), we can analyze the time-to-death using methods from survival analysis. The most common approach for this is to utilize the Cox proportional hazards regression model. We will first focus on how to describe time-to-event data, then we will turn to how to analyze and check assumptions for the Cox proportional hazards model.

### Describing survival data

Survival analysis in R is similar to other types of analyses in R with the exception that you have to first tell R that you are working with time-to-event data. Before we begin to analyze survival data and generate Kaplan-Meier curves, we need to create a Surv() data object.

This is a special data type, unique to the {survival} package. It simultaneously indicates an observation’s time at risk, and whether it was censored.

The Surv() function takes two objects from our data frame: the subject’s follow-up time and whether or not they experienced the event.

In the present analysis, that information is stored in our variables time (how many months the child has survived after birth) and death, respectively. We can create this object as a new variable in our data frame using mutate():

kenya <- kenya %>%  
 mutate(surv\_data = Surv(time = time, event = death))

Be sure to inspect that new Surv() column. Here’s how it appears next to our variables for time and death:

| time | death | surv\_data |
| --- | --- | --- |
| 13 | 0 | 13+ |
| 21 | 0 | 21+ |
| 38 | 0 | 38+ |
| 205 | 0 | 205+ |
| 153 | 0 | 153+ |
| 262 | 0 | 262+ |
| 38 | 0 | 38+ |
| 192 | 0 | 192+ |
| 301 | 0 | 301+ |
| 325 | 0 | 325+ |
| 79 | 0 | 79+ |
| 158 | 0 | 158+ |

Note that Surv() can also be created as a standalone object and does not need to be attached to a data frame as a new column.

We can then use our Surv() object to build a survfit() time-to-event model, which is parameterized just like in the function lm().

surv\_bord5 <- survfit(surv\_data ~ bord5, data = kenya)

If we inspect this model, we’ll see that it is the tabular representation of a survival curve. Use summary() to inspect it for yourself.

summary(surv\_bord5)

Seriously, run this summary() code for yourself and see. The output is too long for the website. Once you do, notice how these calculations are similar to the ones your performed manually in the take-home assignment from class.

The summary of surv\_bord5 will show us a tabulation of the cohort’s survival probability at each time-step, along with how many were at risk and the number of events in that interval.

### Describing Survival Data

Survival data can be described both graphically and analytically. Measures such as total time at risk, number of subjects and median survival time are common values reported in survival analysis reports.

If we print() the surv\_fit() object generated above, we receive a brief set of summary statistics that would suffice for a good portion of Table 1. But median values are NA. And there’s no mean. Oh no!

## Call: survfit(formula = surv\_data ~ bord5, data = kenya)  
##   
## n events median 0.95LCL 0.95UCL  
## bord5=0 1559 141 NA NA NA  
## bord5=1 441 54 NA NA NA

[This Stats Exchange post](https://stats.stackexchange.com/questions/402432/why-median-is-na-for-some-of-the-group-outcomes-in-survival-analysis) explains why. Basically, our data don’t have enough events to calculate a median.

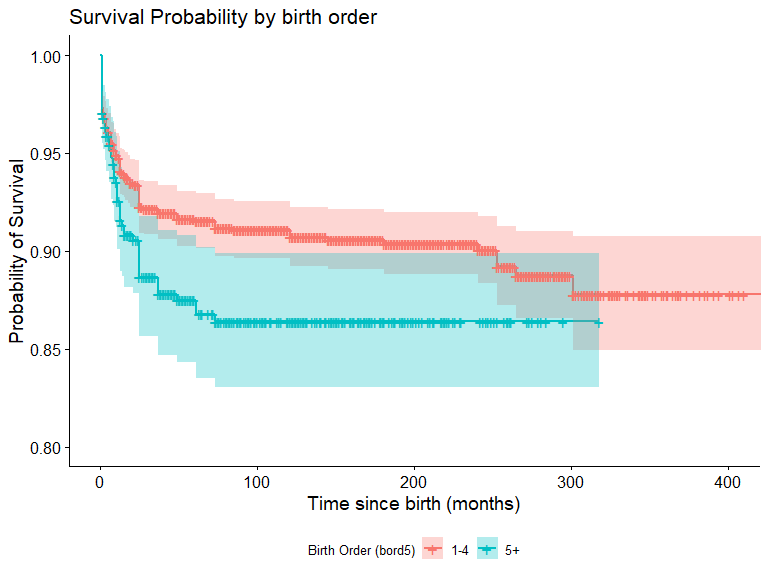
In view of this, to fill out Table 1, we will need to rely on the summarize() function to get summary data for the variables time and death

### Kaplan-Meier Curves

Graphically, Kaplan-Meier curves are plots of the probability of survival as a function of time. These are often presented for each level of the primary exposure variable and are usually unadjusted (although adjusted KM curves are possible).

The package {survminer} contains plotting functions that interact with survfit() objects to plot survival curves. The syntax is a bit different than {ggplot2}, but the concept is the same.

ggsurvplot(fit = surv\_bord5, data = kenya,  
 ylim = c(.8, 1),  
 conf.int = TRUE,  
 xlab = "Time since birth (months)",  
 ylab = "Probability of Survival",  
 title = "Survival Probability by birth order",  
 legend = "bottom", # this specifies legend position  
 legend.title = "Birth Order (bord5)",  
 legend.labs = c("1-4", "5+")  
 )



For more plotting options of ggsurvplot(), check out [this link](http://www.sthda.com/english/wiki/survminer-r-package-survival-data-analysis-and-visualization)

### Cox Proportional Hazards Models

#### Reading - Hazard Ratios

(Click on the thumbnail for further reading.)

In R, Cox PH models are parameterized just like the survfit() function, in that they take a Surv() object as the response variable. However, they’re different in their output. The output is similar to the familiar regression output, with coefficients.

cox\_bord5 <- coxph(Surv(time, death) ~ bord5, data = kenya, ties = "breslow")  
  
summary(cox\_bord5)

## Call:  
## coxph(formula = Surv(time, death) ~ bord5, data = kenya, ties = "breslow")  
##   
## n= 2000, number of events= 195   
##   
## coef exp(coef) se(coef) z Pr(>|z|)   
## bord5 0.3874 1.4732 0.1606 2.413 0.0158 \*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## exp(coef) exp(-coef) lower .95 upper .95  
## bord5 1.473 0.6788 1.075 2.018  
##   
## Concordance= 0.537 (se = 0.016 )  
## Likelihood ratio test= 5.48 on 1 df, p=0.02  
## Wald test = 5.82 on 1 df, p=0.02  
## Score (logrank) test = 5.89 on 1 df, p=0.02

#### Adjusted models

To estimate the effect of bord5, adjusting for maternal education (coded with disjoint indicator variables), we would use the following code:

cox\_bord5\_ed <- coxph(Surv(time, death) ~ bord5 +   
 education\_c2 + education\_c3,   
 ties = "breslow",  
 data = kenya)

Alternatively, we can parameterize education as a factor-type variable to obtain the same result:

cox\_bord5\_ed <- coxph(Surv(time, death) ~ bord5 + factor(education),   
 ties = "breslow",  
 data = kenya)

#### Violation of the Proportional Hazards Assumption

The standard Cox proportional hazards model assumes that the effect of a covariate on the time to an event (e.g. bord5) remains the same across follow-up (i.e., is proportional or constant over time). This might not be a reasonable assumption. We can evaluate if our data are consistent with this assumption through graphical or test-based methods.

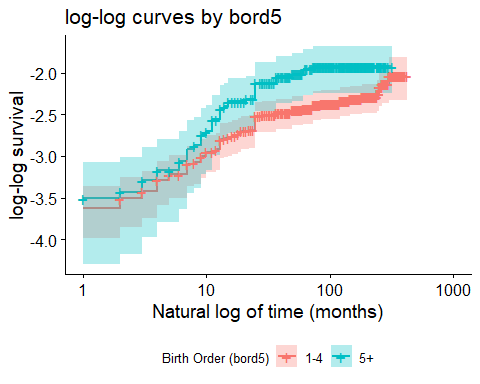
A common graphical method is to plot a transformation of survival for categories of a variable (S(t), the probability of not having the event) against the natural logarithm of follow-up time. In particular, the ln(-ln(S(t)) for two categories of a given variable should remain fairly parallel in a loose sense (never cross, although the lines may not be perfectly straight).

If two lines cross, or appear to be converging (or diverging) even outside of the range of time used, then this can indicate that the proportional hazards assumption may not be reasonable and you should explore estimating an effect that varies over time.

These plots are only useful for categorical variables and cannot be utilized for continuous variables.

In R, we generate this type of plot similar to a Kaplan Meier, except we need to specify our function (fun). Its default gives the survival probability, but here we will set fun = "cloglog".

ggsurvplot(surv\_bord5, fun = "cloglog",  
 conf.int = TRUE,  
 xlab = "Natural log of time (months)",  
 ylab = "log-log survival",  
 title = "log-log curves by bord5",  
 legend = "bottom", # this specifies legend position  
 legend.title = "Birth Order (bord5)",  
 legend.labs = c("1-4", "5+"))



Note that we have used the original survival object, surv\_bord5 to generate this plot. The coxph() model object cannot be used for plotting, unfortuneately.

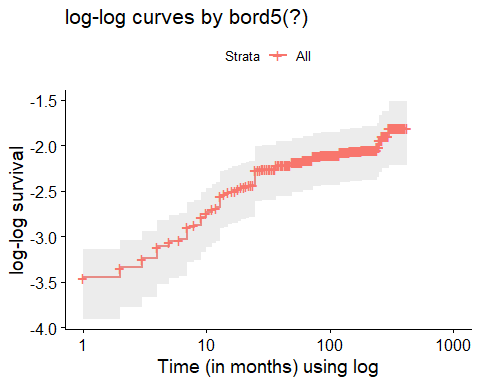
#### Plotting adjusted models

To plot an adjusted model, we’ll need to convert it to a survfit() object:

surv\_bord5\_ed <- survfit(cox\_bord5\_ed, data = kenya)

You might notice that if we try and plot the adjusted model, the plot appears unstratified:

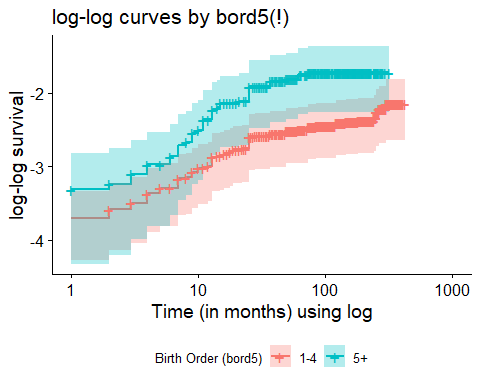
ggsurvplot(surv\_bord5\_ed, fun = "cloglog",  
 conf.int = TRUE,  
 xlab = "Time (in months) using log",  
 ylab = "log-log survival",  
 title = "log-log curves by bord5(?)")



The strata() function allows us to indicate which variable we wish to evaluate for the proportional hazards assumptions. This function is used *within* our model equation.

So to get a proper stratified “cloglog” plot from our adjusted Cox model, we must re-build the model and specify which variable to stratify with strata(), then convert to survfit(), and ***only then*** will it plot properly:

cox\_bord5\_ed\_strata <- coxph(Surv(time, death) ~ strata(bord5) +   
 education\_c2 + education\_c3 + male + mage + mage2,   
 ties = "breslow",  
 data = kenya)  
  
surv\_bord5\_ed\_strata <- survfit(cox\_bord5\_ed\_strata, data = kenya)  
  
ggsurvplot(surv\_bord5\_ed\_strata, fun = "cloglog",  
 conf.int = TRUE,  
 xlab = "Time (in months) using log",  
 ylab = "log-log survival",  
 title = "log-log curves by bord5(!)",  
 legend = "bottom", # this specifies legend position  
 legend.title = "Birth Order (bord5)",  
 legend.labs = c("1-4", "5+"))



If we compared these adjusted log-log survival curves with those of the plain model, where bord5 is the only predictor (formula = Surv(time, death) ~ bord5), we would find that the plots are quite alike, but contain subtle differences in the alignment of the curves. In evaluating the proportionality assumption in Cox regression, we only need the assumption to hold, conditional on modeled covariates; so plotting these adjusted curves is useful.

### Estimating Time-varying effects

In the event that you find the proportional hazards assumption violated for a particular covariate, you should report the effect of the exposure at several time points (you can think of this as effect modification of the exposure by follow-up time). The specification of a Cox model with a time-varying effect for the covariate is:

Notice that this is in a form similar to our original Cox regression formula, but contains an interaction term between our main effect and , time.

To estimate the hazard ratio for vs.  at time :

You can use the Cox regression model that contains the time-varying effect lincom() command to generate time-varying effects. Remember to ‘eform’!

#### Formal statistical test of proportional hazards

We can use R to formally test the significance of this time-varying effect as it pertains to the assumption of proportional hazards.

By allowing the hazard ratio to vary across time, we test if this **time-varying effect** is significant. To perform this, we use the tt() function in our model formula, and then use the argument tt = to specify the function by which time is interacting with bord5 in our model:

cox\_tt <- coxph(Surv(time, death) ~ bord5 + male + rural + education\_c2 + education\_c3 + mage + mage2 + tt(bord5),  
 data = kenya,   
 ties = "breslow",  
 tt = function(x, time,...)x\*time)

## Calculating incidence rates and related measures of association

Just as the generalized linear model can be used to estimate risks, risk differences and risk ratios, we can also use this methodology to estimate incidence rates and incidence rate ratios from the counts of outcomes over time. The Poisson distribution is used to model counts or rates and has the general form:

where **family = “poisson”** and **link = “log”**.

Notice that the specification of the Poisson model includes the number of deaths and the logarithm of the exposure time. Rearranging the above equation and a bit of algebra shows how this can be used to derive the incidence rate (IR):

To estimate incidence rates for :

To estimate the incidence rate ratio (IRR) for vs. :

IRs and IRRs can be estimated through linear combinations of the regression parameters as we have done with previous linear models, using lincom().

### R commands for the tabular analysis of rates

You can use the pyears() function from the {survival} package to generate values that are useful in calculating incidence rates and incidence rate ratios

ptime\_bord5 <- pyears(Surv(time, death) ~ strata(bord5),   
 data = kenya, scale = 1)  
  
summary(ptime\_bord5)

## Call: pyears(formula = Surv(time, death) ~ strata(bord5), data = kenya,   
## scale = 1)  
##   
## number of observations = 2000  
##   
## strata(bord5) N Events Time   
## --------------- ------ -------- --------   
## bord5=0 1559 141 203041   
## bord5=1 441 54 40760

You can then access values within the pyears() object like this:

# events of death in strata bord5 = 0  
ptime\_bord5$event[1]

## bord5=0   
## 141

# person time in strata bord5 = 0  
ptime\_bord5$pyears[1]

## bord5=0   
## 203041

# events of death in strata bord5 = 1  
ptime\_bord5$event[2]

## bord5=1   
## 54

# person time in strata bord5 = 1  
ptime\_bord5$pyears[2]

## bord5=1   
## 40760

You can then use the functions ratedifference() and rateratio() from the package {fmsb} to calculate the incidence rate difference and incidence rate ratio, respectively.

Both functions take the same arguments:

ratedifference(a, b, PT1, PT0, conf.level = 0.95)  
rateratio(a, b, PT1, PT0, conf.level = 0.95)

Where ***a*** is the number of disease occurence among the exposed,

***b*** is the number of diease occurence among the unexposed,

***PT1*** is the observed person-time in the exposed cohort,

And ***PT0*** is the observed person-time in the unexposed cohort.

### R Commands for Poisson Regression Models

The command for a Poisson regression model has the following form:

poisson\_bord5 <- glm(death ~ bord5 + offset(log(time)),   
 family = "poisson"(link = "log"), data = kenya)

Access the model summary just like any normal glm() model:

summary(poisson\_bord5)

Note that the exposure time is explicitly specified in the GLM options using offset(). We must use the natural log of time to generate the appropriate estimates since our model is fit on the log scale (link = "log").

To generate estimates of Incidence Rates (IR) and Incidence Rate Ratios (IRR), you can use the lincom() command with your poisson models.

# IR at intercept  
lincom(poisson\_bord5, "(Intercept)", eform = TRUE)  
  
# IR at bord5 == "5+"  
lincom(poisson\_bord5, "(Intercept) + bord5", eform = TRUE)  
  
# IRR and CI  
lincom(poisson\_bord5, "bord5", eform = TRUE)

To have IR/IRR reported (instead of beta coefficients), be sure to set eform = TRUE in your lincom() function.