Survival Analysis

Monica Alexander

January 27 2021

Contents

1 Introduction

This lab will take you through some main techniques for use in survival analysis.

1.1 What to hand in

Please push your Rmd and compiled document **in PDF form** to GitHub. **The questions for this week are dispersed throughout the lab.**

1.2 Data

In this lab we're going to use the `fert` dataset in the `eha` package. This data relates to times between births for women in Sweden in the 19th century.

As in the lecture, we're just going to look at women of parity 1: this is just demography-speak for women who have had one child already.

So the focus of our survival analysis is the time to second birth. The variable of interest is `next.ivl`, which is the number of years until the next birth. Also of interest is the `event` variable, which tells us whether the birth happened (or whether the woman is censored).

```
library(tidyverse) # the old fave
library(survival) # useful stuff for survival analysis
library(eha) # has the dataset

data(fert)
f12 <- fert %>% as_tibble() %>% filter(parity ==1)
head(f12)
```

```
## # A tibble: 6 x 9
##      id parity   age  year next.ivl event prev.ivl ses     parish
##   <dbl>  <dbl> <dbl> <dbl>    <dbl> <dbl>    <dbl> <fct>   <fct>
## 1     1      1    25  1826    22.3      0    0.411 farmer  SKL
## 2     2      1    19  1821     1.84     1    0.304 unknown SKL
## 3     3      1    24  1827     2.05     1    0.772 farmer  SKL
## 4     4      1    35  1838     1.78     0    6.79  unknown SKL
## 5     5      1    28  1832     1.63     1    3.03  farmer  SKL
## 6     6      1    25  1829     1.73     1    0.819 lower   SKL
```

Let's make a new age group variable, splitting the women by whether or not they are less than 30 years old.

```
f12 <- f12 %>%
  mutate(age_group = ifelse(age<30, "<30", "30+"))
```

2 Descriptives

2.1 Question 1

With plots or tables, give me three observations about the times to second births. At least one of these observations should be related to differences by `age_group`.

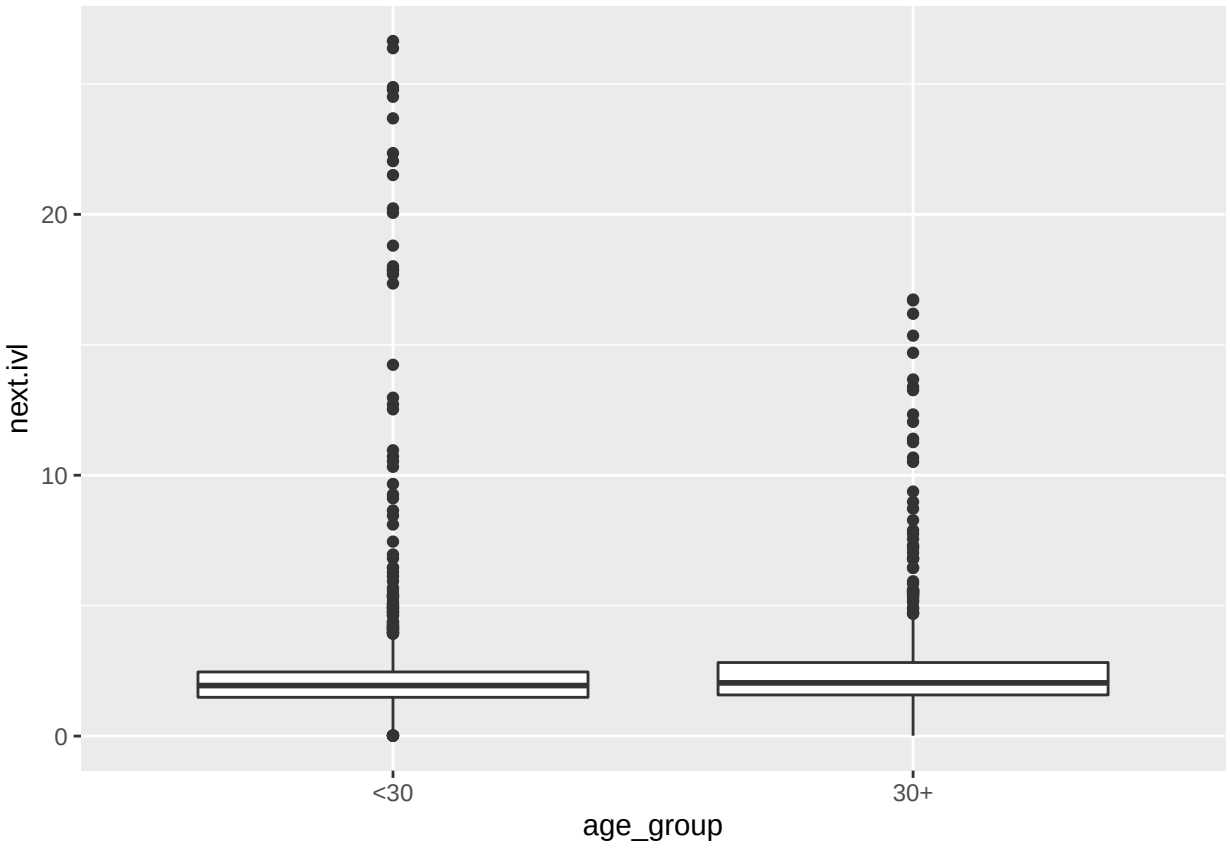
2.2 Answer

We have selected the largest next.ivl in age group < 30 (id 562), the largest next.ivl in age group 30+ (id 1712) and the first sample among all samples. We could find just by those three samples. The time to second births in <30 age group is larger than 30+ age group. From the boxplot we could confirm the same conclusion, <30 age group is larger than 30+ age group.

```
print(f12[f12$id %in% c(562,1712,1),])
```

```
## # A tibble: 3 x 10
##      id parity   age  year next.ivl event prev.ivl ses     parish age_group
##   <dbl>  <dbl> <dbl> <dbl>    <dbl> <dbl>    <dbl> <fct>   <fct>  <chr>
## 1     1      1    25  1826     22.3     0    0.411 farmer  SKL    <30
## 2   562      1    20  1845     26.6     0    0.192 unknown SKL    <30
## 3  1712      1    30  1879     16.7     0    0.873 farmer  SKL    30+
```

```
p <- ggplot(f12, aes(x=age_group, y=next.ivl)) +
  geom_boxplot()
p
```



3 Kaplan Meier

First we will calculate the non-parametric version of the survival function.

3.1 Surv objects

Surv objects are set of ordered times with the censors indicated with a plus:

```
survobject <- Surv(time = f12$next.ivl, event = f12$event)
head(survobject)
```

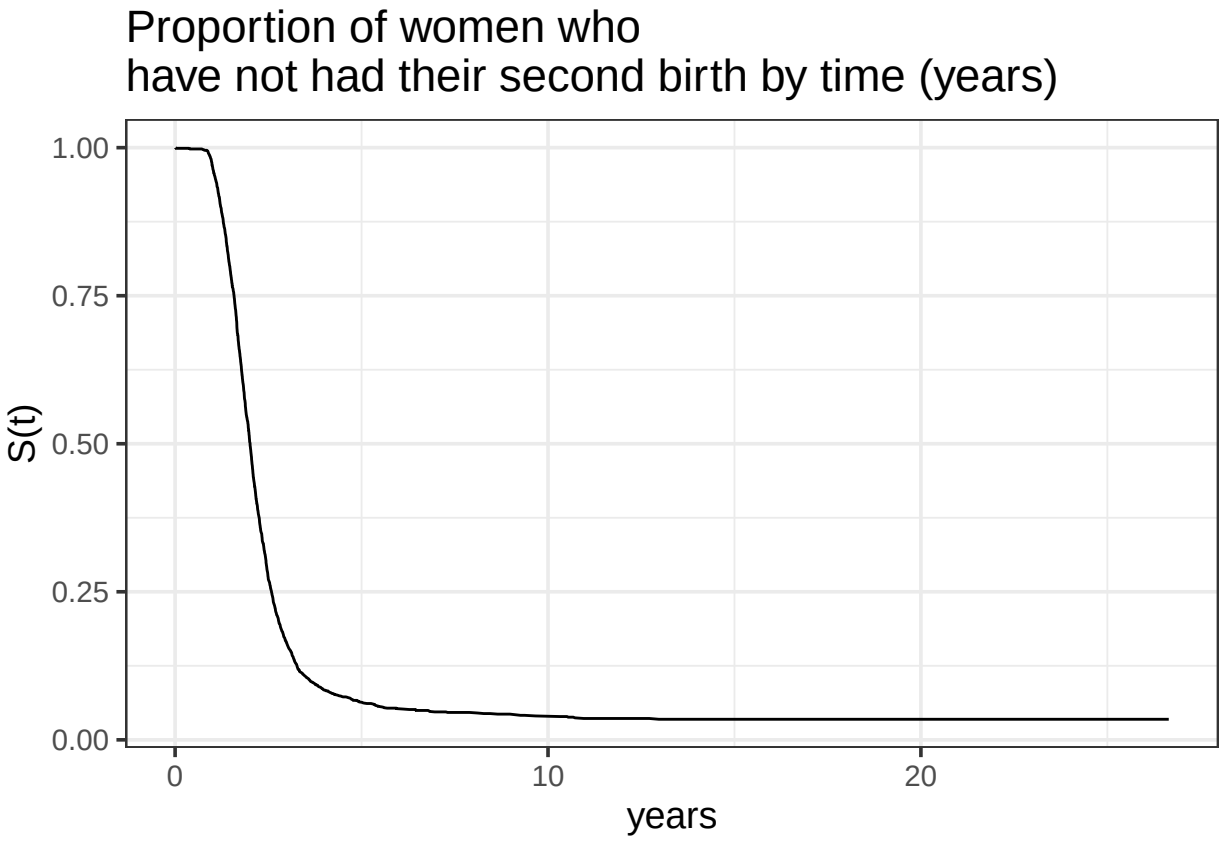
```
## [1] 22.348+  1.837   2.051   1.782+  1.629   1.730
```

These can feed into the survfit function from the survival package to estimate the KM curve:

```
fit <- survfit(Surv(next.ivl, event) ~ 1, data = f12)

fit_df <- tibble(time = fit$time, surv = fit$surv)

ggplot(aes(time, surv), data = fit_df) +
  geom_line() +
  ggtitle("Proportion of women who \nhave not had their second birth by time (years)") +
  xlab("years") + ylab("S(t)")+
  theme_bw(base_size = 14)
```



3.2 KM by hand

We can calculate Kaplan-Meier by hand fairly easily by setting up our `tibble` in the right way and calculating some new variables.

3.3 Question 2

Fill in the gaps below (denoted by XX)

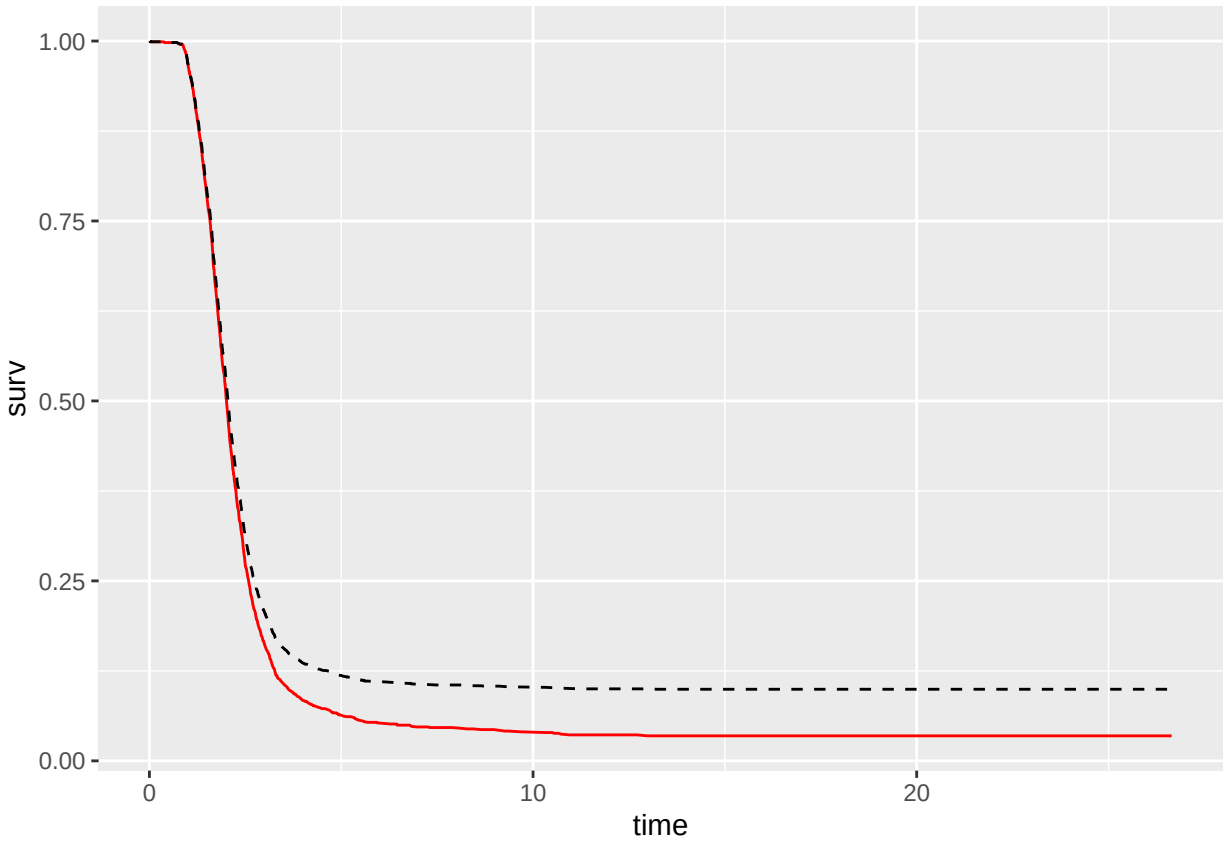
```
n <- nrow(f12)

f12 <- f12 %>%
  arrange(next.ivl) %>% # need to sort by survival times
  mutate(cumulative_people_lost = cumsum(event),
         exposure = lag(n - cumulative_people_lost, default = n),
         prob_birth = cumulative_people_lost/(cumulative_people_lost + exposure),
         prob_surv = exposure/(cumulative_people_lost + exposure),
         surv = cumprod(prob_surv))
```

If your code worked, the survival curve should be identical to what we got using `survfit`:

(NOTE: you will need to delete the `eval=F` from this chunk and the above chunk before you compile)

```
ggplot(aes(time, surv), data = fit_df) +
  geom_line(color = "red") +
  geom_line(aes(next.ivl, prob_surv), data = f12, lty = 2)
```



3.4 Question 3

When have 75% of the women had their second child?

3.5 Answer:

It was 1859. Or 2.74 for the next.ivl, which could be verified on the graph.

```
f12_greater_75 = f12[f12$prob_birth>0.75,]
print(f12_greater_75[1,])
```

```
## # A tibble: 1 x 15
##      id parity   age  year next.ivl event prev.ivl ses   parish age_group
##   <dbl>  <dbl> <dbl> <dbl>    <dbl> <dbl>    <dbl> <fct> <fct>  <chr>
## 1   908      1    29  1859     2.74     1     1.35 farm~ SKL    <30
## # ... with 5 more variables: cumulative_people_lost <dbl>, exposure <dbl>,
## #   prob_birth <dbl>, prob_surv <dbl>, surv <dbl>
```

4 Piecewise Constant Hazards

Let's now estimate a PCH model, using the same cut-points as in the lecture.

4.1 `survSplit`

To do this, we first need to get our data in the form of tracking deaths/censors in each interval. We could do this by hand, but easier with the `survSplit` function. After doing the survSplit, we then create an interval

factor (for use in regression) and an interval length variable. Make sure you understand the form of this new `f12_split` and what all these new variables are.

```
cutpoints <- c(10/12, 1.25, 1.75, 2.25, seq(3,5), seq(6, 12, by = 3))
C <- length(cutpoints) + 1

f12_split <- survSplit(formula = Surv(time = next.ivl, event = event) ~ .,
                       data = f12, cut = cutpoints) %>%
  as_tibble() %>%
  mutate(interval = factor(tstart),
         interval_length = next.ivl - tstart)
f12_split
```

```
## # A tibble: 7,625 x 18
##       id parity   age  year prev.ivl ses   parish age_group
##    <dbl>  <dbl> <dbl> <dbl>    <dbl> <fct> <fct>  <chr>
##  1  1841      1    26  1884    0.969 lower SKL    <30
##  2   456      1    28  1851    1.95  farm~ SKL    <30
##  3   942      1    21  1852    0.463 farm~ SKL    <30
##  4  1249      1    37  1875    0.778 farm~ SKL    30+
##  5   961      1    24  1856    0.126 lower SKL    <30
##  6  1076      1    41  1875    1.81  farm~ SKL    30+
##  7  1858      1    34  1898    0.882 farm~ SKL    30+
##  8  1644      1    28  1874    0.819 upper SKL    <30
##  9   238      1    27  1845    1.28  farm~ SKL    <30
## 10  1704      1    34  1882    1.92  unkn~ SKL    30+
## # ... with 7,615 more rows, and 10 more variables:
## #   cumulative_people_lost <dbl>, exposure <dbl>, prob_birth <dbl>,
## #   prob_surv <dbl>, surv <dbl>, tstart <dbl>, next.ivl <dbl>,
## #   event <dbl>, interval <fct>, interval_length <dbl>
```

Now run the regression

```
fit_ind <- glm(event ~ offset(log(interval_length))-1 + interval, data=f12_split, family = "poisson")
summary(fit_ind)
```

```
##
## Call:
## glm(formula = event ~ offset(log(interval_length)) - 1 + interval,
##     family = "poisson", data = f12_split)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.3123  -0.7946  -0.4692  -0.0941   4.1246
##
## Coefficients:
##                           Estimate Std. Error z value Pr(>|z|)
## interval0                 -5.23731    0.35353 -14.814  < 2e-16 ***
## interval0.833333333333333 -1.33094    0.07313 -18.200  < 2e-16 ***
## interval1.25              -0.45976    0.04800  -9.578  < 2e-16 ***
## interval1.75               0.05636    0.04637   1.215  0.22423
## interval2.25               0.13803    0.05227   2.641  0.00827 **
## interval3                 -0.36345    0.08771  -4.144 3.41e-05 ***
## interval4                 -1.27428    0.17678  -7.208 5.66e-13 ***
## interval5                 -1.61037    0.25000  -6.442 1.18e-10 ***
## interval6                 -2.68981    0.30151  -8.921  < 2e-16 ***
```

```
## interval9                 -2.74371    0.37796  -7.259 3.89e-13 ***
## interval12                -5.24424    1.00000  -5.244 1.57e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 10885.5  on 7625  degrees of freedom
## Residual deviance:  6361.4  on 7614  degrees of freedom
## AIC: 9697.4
##
## Number of Fisher Scoring iterations: 7
```

Alternatively, we could run the Poisson regression using the sums over each interval. The results are exactly the same:

```
E_k <- f12_split %>% group_by(interval) %>% summarise(E = sum(next.ivl-tstart)) %>% select(E) %>% pull()
D_k <- f12_split %>% group_by(interval) %>% summarise(D = sum(event)) %>% select(D) %>% pull()

intervals <- unique(f12_split$interval) # number of intervals
fit_pois <- glm(D_k ~ offset(log(E_k))-1 + intervals, family = "poisson")
summary(fit_pois)
```

```
##
## Call:
## glm(formula = D_k ~ offset(log(E_k)) - 1 + intervals, family = "poisson")
##
## Deviance Residuals:
##  [1]  0  0  0  0  0  0  0  0  0  0  0  0
##
## Coefficients:
##                            Estimate Std. Error z value Pr(>|z|)
## intervals0                 -5.23731    0.35355 -14.813  < 2e-16 ***
## intervals0.833333333333333 -1.33094    0.07313 -18.200  < 2e-16 ***
## intervals1.25              -0.45976    0.04800  -9.578  < 2e-16 ***
## intervals1.75               0.05636    0.04637   1.215  0.22423
## intervals2.25               0.13803    0.05227   2.641  0.00827 **
## intervals3                 -0.36345    0.08771  -4.144 3.41e-05 ***
## intervals4                 -1.27428    0.17678  -7.208 5.66e-13 ***
## intervals5                 -1.61037    0.25000  -6.441 1.18e-10 ***
## intervals6                 -2.68981    0.30151  -8.921  < 2e-16 ***
## intervals9                 -2.74371    0.37796  -7.259 3.89e-13 ***
## intervals12                -5.24424    1.00000  -5.244 1.57e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 4.5241e+03  on 11  degrees of freedom
## Residual deviance: 6.2839e-14  on  0  degrees of freedom
## AIC: 83.335
##
## Number of Fisher Scoring iterations: 3
```

Hazards are the transformed coefficients,

```
exp(coef(fit_pois))
```

```
##                  intervals0 intervals0.833333333333333
##                 0.005314553                 0.264229787
##               intervals1.25               intervals1.75
##                 0.631435293                 1.057979555
##               intervals2.25                  intervals3
##                 1.148011995                 0.695272681
##                  intervals4                  intervals5
##                 0.279632284                 0.199812676
##                  intervals6                  intervals9
##                 0.067894110                 0.064331140
##                 intervals12
##                 0.005277853
```

and you can get the standard errors from the output, too. To get the approximate SEs around the hazards rates, use the delta method:

```
sqrt(diag(vcov(fit_pois)))*exp(coef(fit_pois))
```

```
##                  intervals0 intervals0.833333333333333
##                 0.001878978                 0.019322396
##               intervals1.25               intervals1.75
##                 0.030309864                 0.049062627
##               intervals2.25                  intervals3
##                 0.060007548                 0.060979448
##                  intervals4                  intervals5
##                 0.049432471                 0.049953169
##                  intervals6                  intervals9
##                 0.020470844                 0.024314885
##                 intervals12
##                 0.005277827
```

4.2 Question 4

Confirm that the estimated hazards from the regression are the same as the rates of birth in each interval implied by the data.

Answer: We could confirm this is the case.

```
print(D_k/E_k)
```

```
##  [1] 0.005314553 0.264229787 0.631435293 1.057979555 1.148011995
##  [6] 0.695272681 0.279632284 0.199812676 0.067894110 0.064331140
## [11] 0.005277853
```

```
print(exp(coef(fit_pois)))
```

```
##                  intervals0 intervals0.833333333333333
##                 0.005314553                 0.264229787
##               intervals1.25               intervals1.75
##                 0.631435293                 1.057979555
##               intervals2.25                  intervals3
##                 1.148011995                 0.695272681
##                  intervals4                  intervals5
##                 0.279632284                 0.199812676
##                  intervals6                  intervals9
```

```
##                0.067894110                0.064331140
##                intervals12
##                0.005277853
```

4.3 Visualizing hazards

In the lecture, I made a step-wise plot to visualize these hazards. The first step to get this is to make a tibble with our hazard rates, SEs and cut points. I add an extra point at the end, representing the maximum time observed:

```
C <- length(cutpoints)+1
cuts <- c(0,cutpoints,max(f12$next.ivl))
hazs <- c(exp(coef(fit_pois)), exp(coef(fit_pois))[C])
ses <- c(sqrt(diag(vcov(fit_pois)))*exp(coef(fit_pois)), sqrt(diag(vcov(fit_pois)))[C]*exp(coef(fit_pois

haz_df <- tibble(cut = cuts, haz = hazs, se = ses)
```

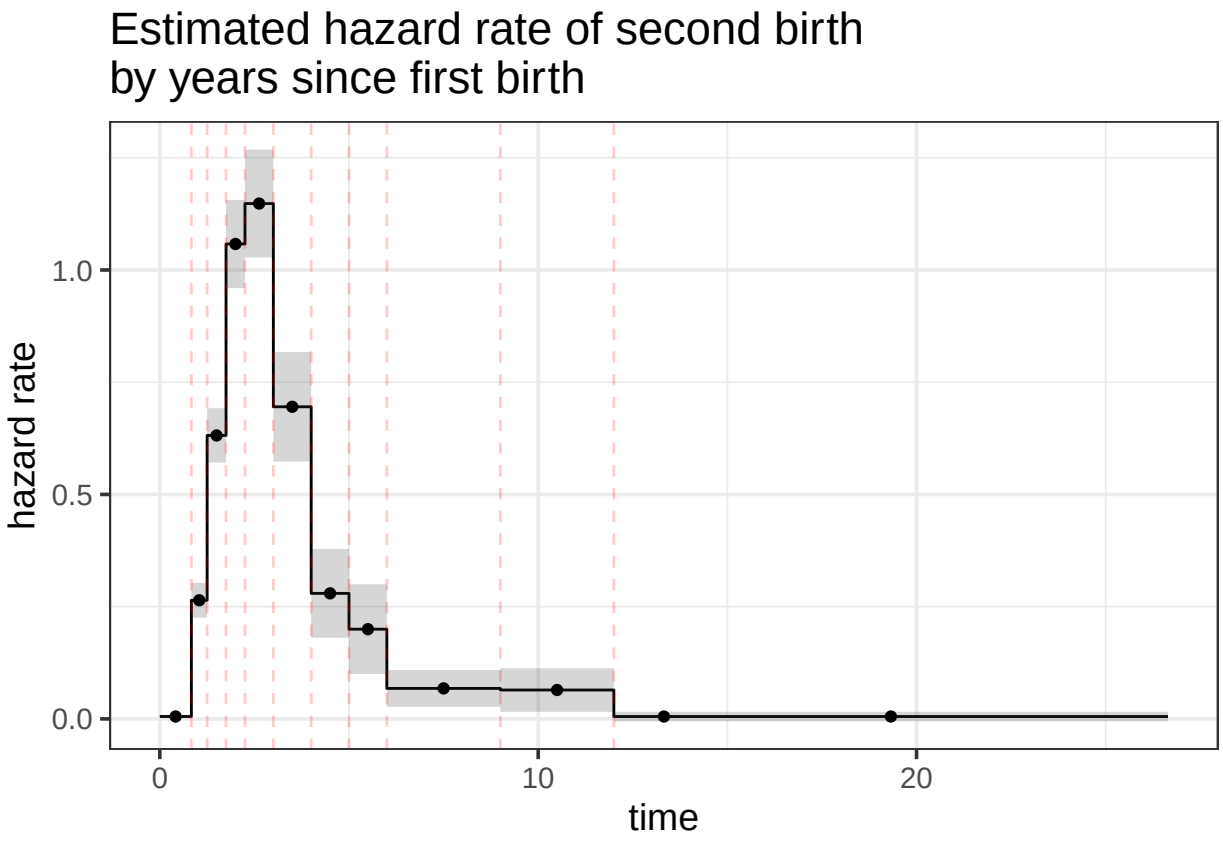
Next we want to make some 95% CIs and calculate the mid-point and end-point of each interval, for plotting purposes.

```
haz_df <- haz_df %>%
  mutate(lower = haz - 2*se,
         upper = haz + 2*se,
         midpoints = cut  + (lead(cut, default = 0) - cut)/2,
         endpoints = lead(cut, default = max(cut)))
```

Now plot!

```
haz_long <- haz_df %>%
  pivot_longer(-(haz:upper), values_to = "time", names_to = "point")

haz_long %>%
  ggplot(aes(time, haz) ) + geom_line() +
  geom_ribbon(aes(ymin = lower, ymax = upper), alpha = 0.2) +
  geom_point(aes(time, haz), data = haz_long %>% filter(point == "midpoints")) +
  geom_vline(xintercept = cutpoints, col = 2, alpha = 0.2, lty = 2) +
  theme_bw(base_size = 14) +
  ylab("hazard rate") +
  ggtitle("Estimated hazard rate of second birth\nby years since first birth")
```



4.4 Survival probabilities

Would be good to also transform these hazards into survival probabilities. Here's the start of a function that does this:

```
survival_prob <- function(lambdas,
                      cuts, # start and end times that lambdas refers to, starting at 0 and ending at ma
                      ## observation time of interest,
                      ## thus length is one more than length of lambda
                      neval = 100 # at how many points do you want to evaluate S(t) within each interval
                      ){
  lengthintervals <- rep((cuts[-1] - cuts[-length(cuts)])/neval, each = neval)
  t_seq <- c(0, cumsum(lengthintervals))
  cumulative_hazard <- cumsum(lengthintervals*rep(lambdas, each  = neval))
  surv_probs <- c(1, exp(-cumulative_hazard)) # add a 1 at the start because everyone survives at the s
  return(tibble(time = t_seq, surv = surv_probs ))
}
```

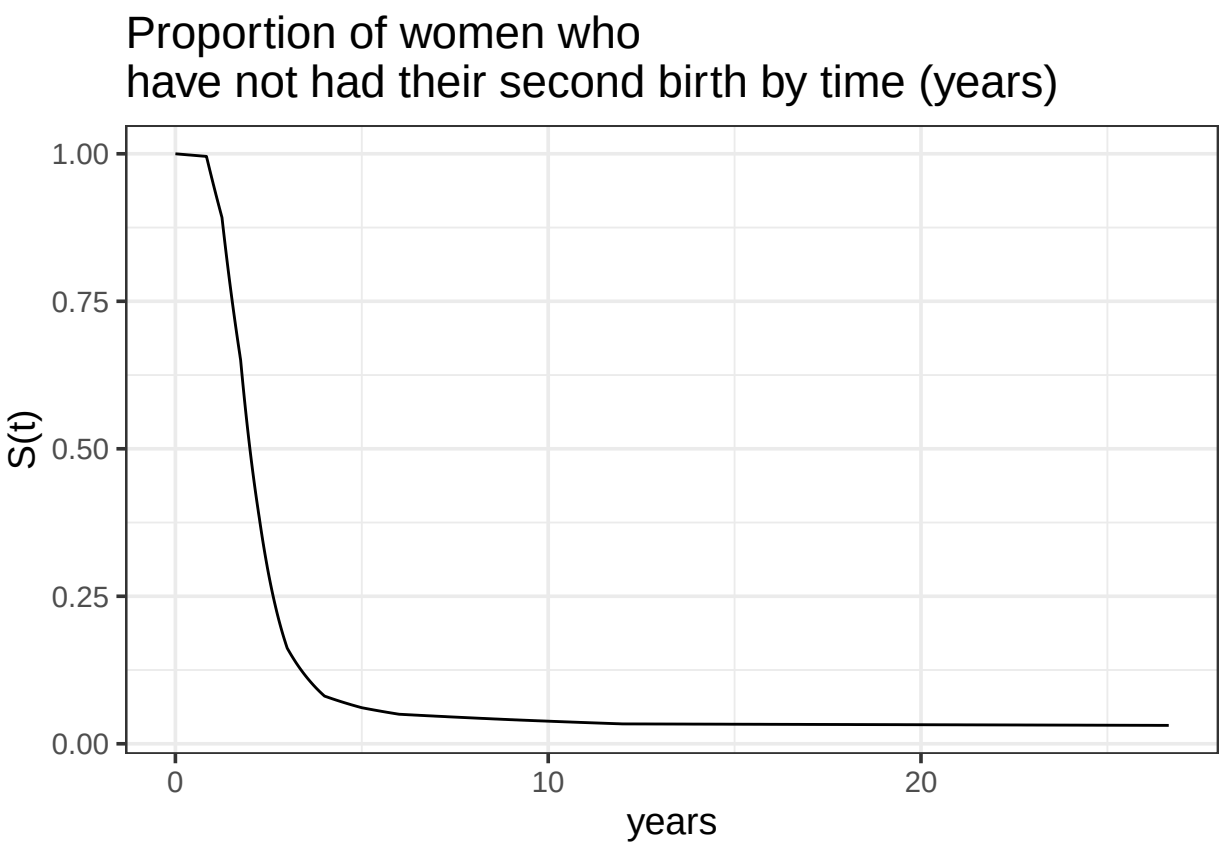
4.5 Question 5

Fill in the gaps above in the `survival_prob` function. (Note you will need to remove `eval = FALSE` again).

Now use this to plot the survival function:

```
lambdas <- exp(coef(fit_pois))
cuts <- c(0, cutpoints, max(f12_split$next.ivl))
df_surv <- survival_prob(lambdas = exp(coef(fit_pois)),
              cuts = cuts)
```

```
ggplot(aes(time, surv), data = df_surv) + geom_line()+
  ggtitle("Proportion of women who \nhave not had their second birth by time (years)") +
  xlab("years") + ylab("S(t)")+
  theme_bw(base_size = 14)
```



Question 5b (BONUS, not required) Using simulation based on estimated log hazards and SEs, calculate and plot the 95% CI around the survival curve above.

5 PCH with covariates

5.1 Question 6

Rerun the PCH regression above but with `age_group` as a covariate (Note: probability easiest just to run the individual-level regression rather than the regression on the sums).

```
age_regression <- glm(event ~ offset(log(interval_length)) - 1 + interval + age_group, data = f12_split

print(summary(age_regression))
```

```
##
## Call:
## glm(formula = event ~ offset(log(interval_length)) - 1 + interval +
##     age_group, family = poisson, data = f12_split)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.3861  -0.8130  -0.4020  -0.0806   4.1041
##
```

```
## Coefficients:
##                            Estimate Std. Error z value Pr(>|z|)
## interval0                  -5.15321    0.35369 -14.570  < 2e-16 ***
## interval0.833333333333333 -1.24604    0.07394 -16.852  < 2e-16 ***
## interval1.25               -0.37196    0.04932  -7.542 4.63e-14 ***
## interval1.75                0.14972    0.04792   3.124  0.00178 **
## interval2.25                0.24754    0.05419   4.568 4.91e-06 ***
## interval3                  -0.21940    0.08977  -2.444  0.01452 *
## interval4                  -1.10730    0.17819  -6.214 5.16e-10 ***
## interval5                  -1.44356    0.25100  -5.751 8.86e-09 ***
## interval6                  -2.54673    0.30211  -8.430  < 2e-16 ***
## interval9                  -2.62211    0.37830  -6.931 4.17e-12 ***
## interval12                 -5.20142    1.00001  -5.201 1.98e-07 ***
## age_group30+               -0.39438    0.05965  -6.611 3.80e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 10885.5  on 7625  degrees of freedom
## Residual deviance:  6314.6  on 7613  degrees of freedom
## AIC: 9652.6
##
## Number of Fisher Scoring iterations: 7
```

5.2 Question 7

Use the `survival_prob` function defined above to help you find the proportion of women aged less than 30 who have had their second birth within 5 years of their first birth.

5.3 Answer:

From the table we could find about 94.6% of women aged less than 30 had their second birth within 5 years of their first birth.

```
exp_coef <- exp(coef(age_regression))
df_surv <- survival_prob(lambdas = exp_coef[-1], cuts = cuts)

print(1 - 0.05409992)
```

```
## [1] 0.9459001
```