Lab Exercise - Jan29th - Survival Analysis

Luis Correia - Student No. 1006508566

January 29th 2020

Introduction

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

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.**

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.

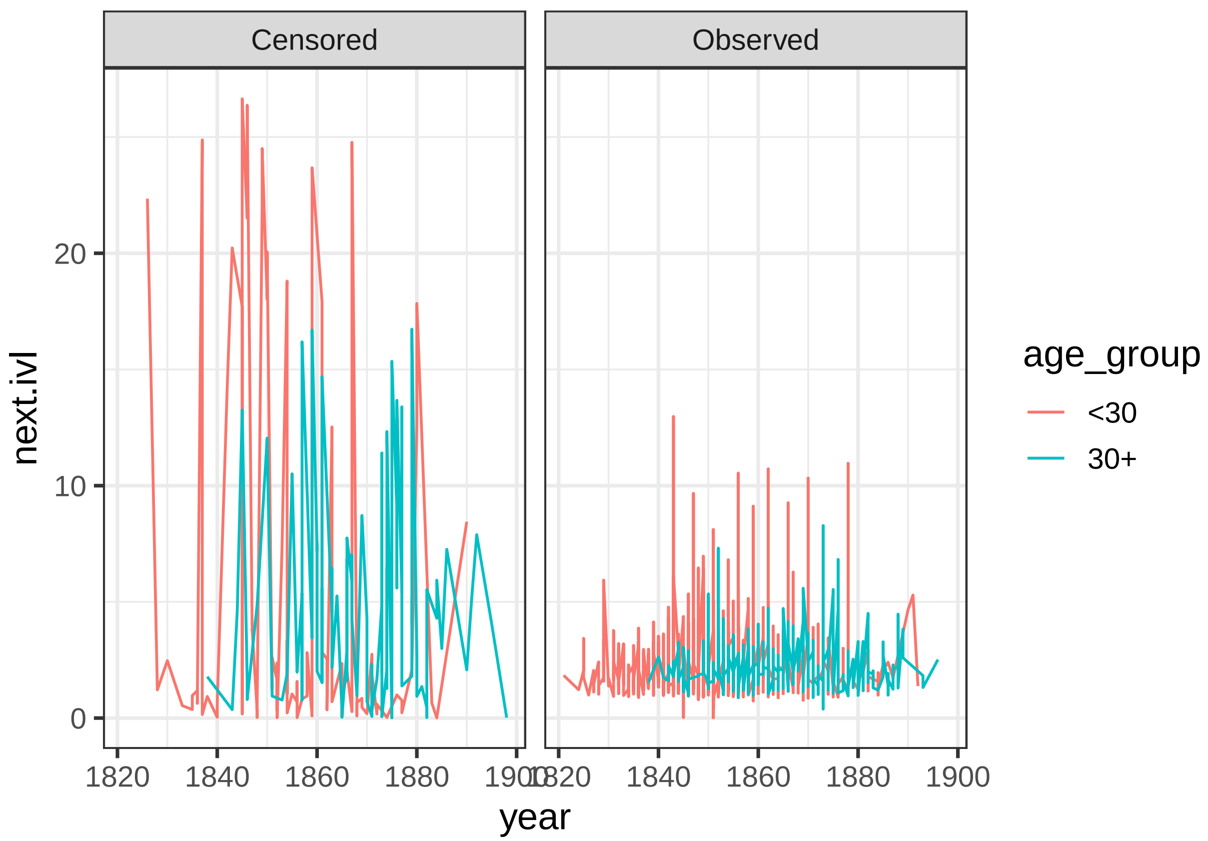
```
f12 <- f12 %>%
  mutate(age_group = ifelse(age<30, "<30", "30+"))
```

Descriptives

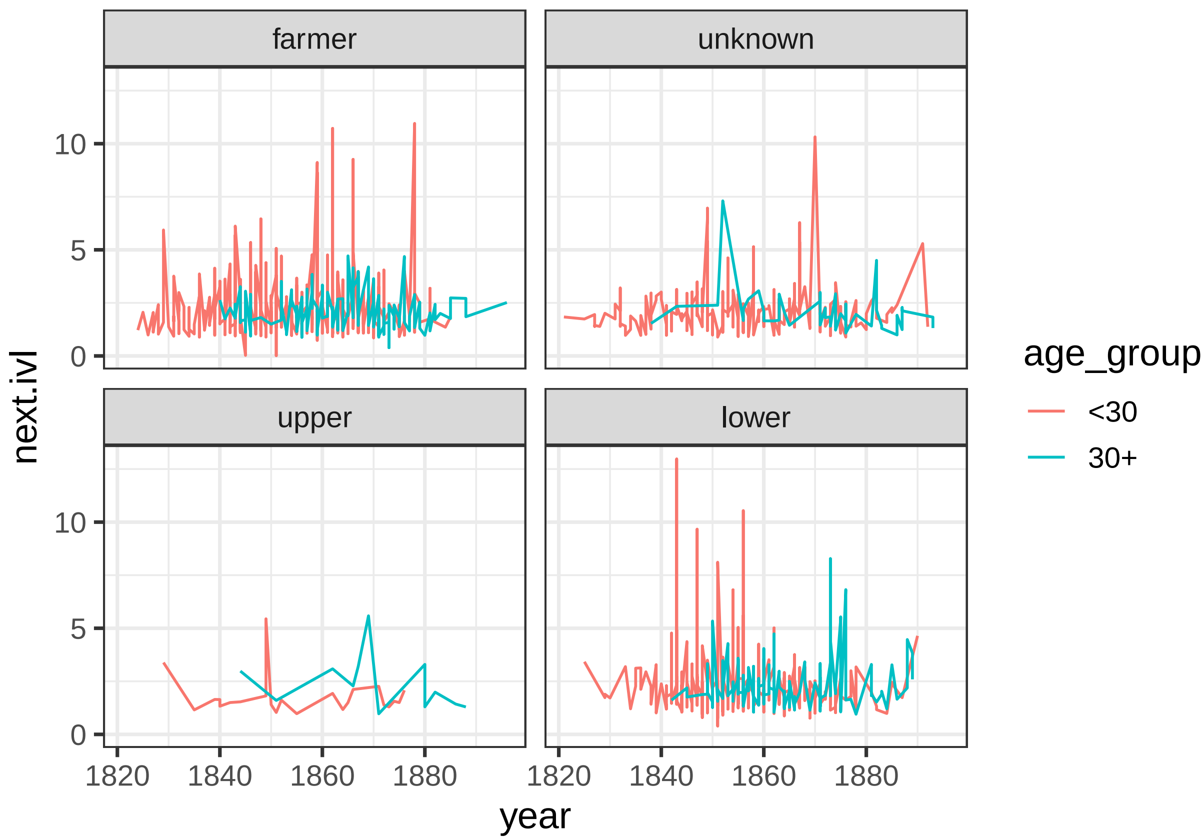
Question 1

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`.

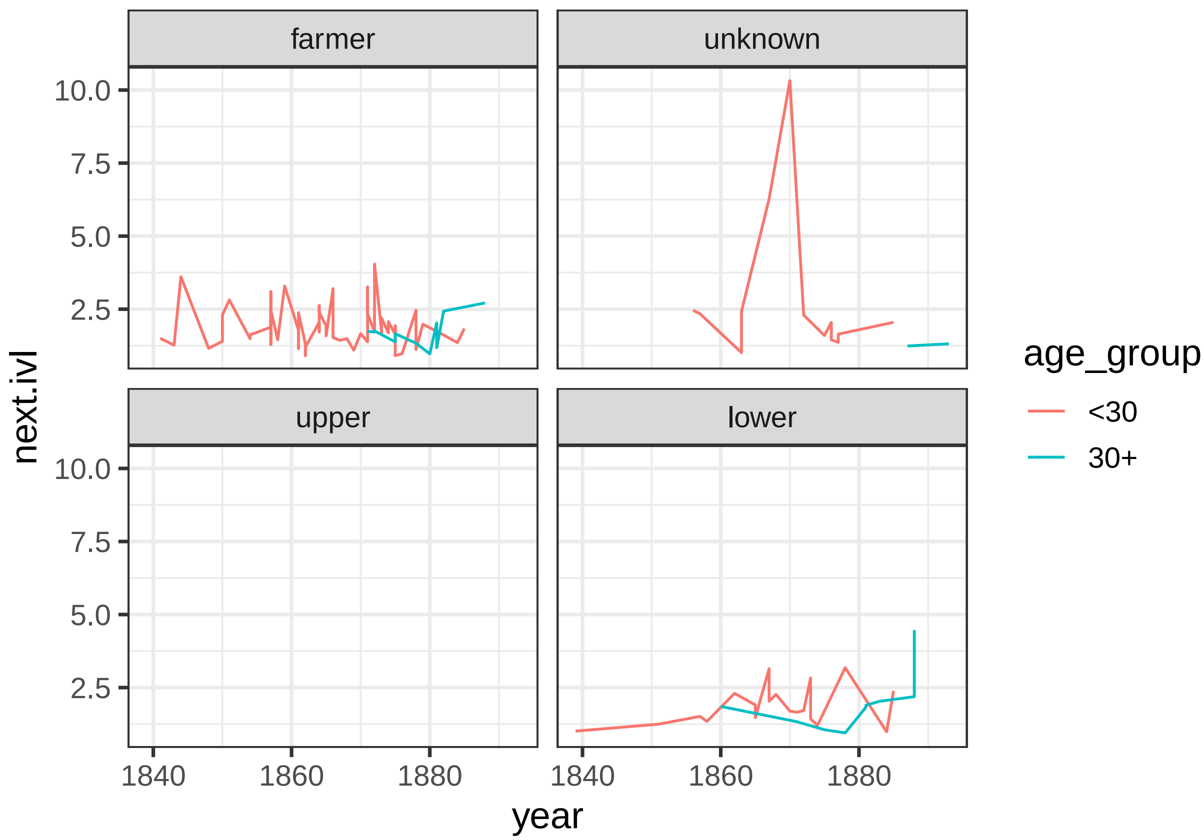
```
f12 %>%
  mutate(typeevent = ifelse(event == 1, "Observed", "Censored")) %>%
  ggplot(aes(x=year, y=next.ivl, color=age_group))+
  geom_line() +
  facet_wrap(~typeevent)+
  theme_bw(base_size = 14)
```




```
f12 %>%
  filter(event == 1) %>%
  ggplot(aes(x=year, y=next.ivl, color=age_group))+
  geom_line() +
  facet_wrap(~ses)+
  theme_bw(base_size = 14)
```




```
f12 %>%
  filter(event == 1, parish != "SKL") %>%
  ggplot(aes(x=year, y=next.ivl, color=age_group))+
  geom_line() +
  facet_wrap(~ses)+
  theme_bw(base_size = 14)
```



{**Q1-Answer:**}

- The variability of years expected until 2nd birth for women with age <30 appears to be larger than for group 30+. This might reflect, as fertility cycle is larger on group <30, women expect longer to have a 2nd. baby when compared with other group.
- Socio-economic status shows differences patterns between 'upper' and 'lower' classes: people from 'upper' class tends to have a smaller waiting time to 2nd birth than other group. The same behaviour appears when comparing 'upper' with 'farmer';
- I both groups (<30 and 30+) the waiting time seems to decrease during the years, with higher rates in early 1800's.
- Parishes of Jorn and Norsjo have a different pattern than Skelleftea with lower waiting time to 2nd birth. They are both mainly constituted from farmers and lower social economic groups.

Kaplan Meier

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

`Surv` objects

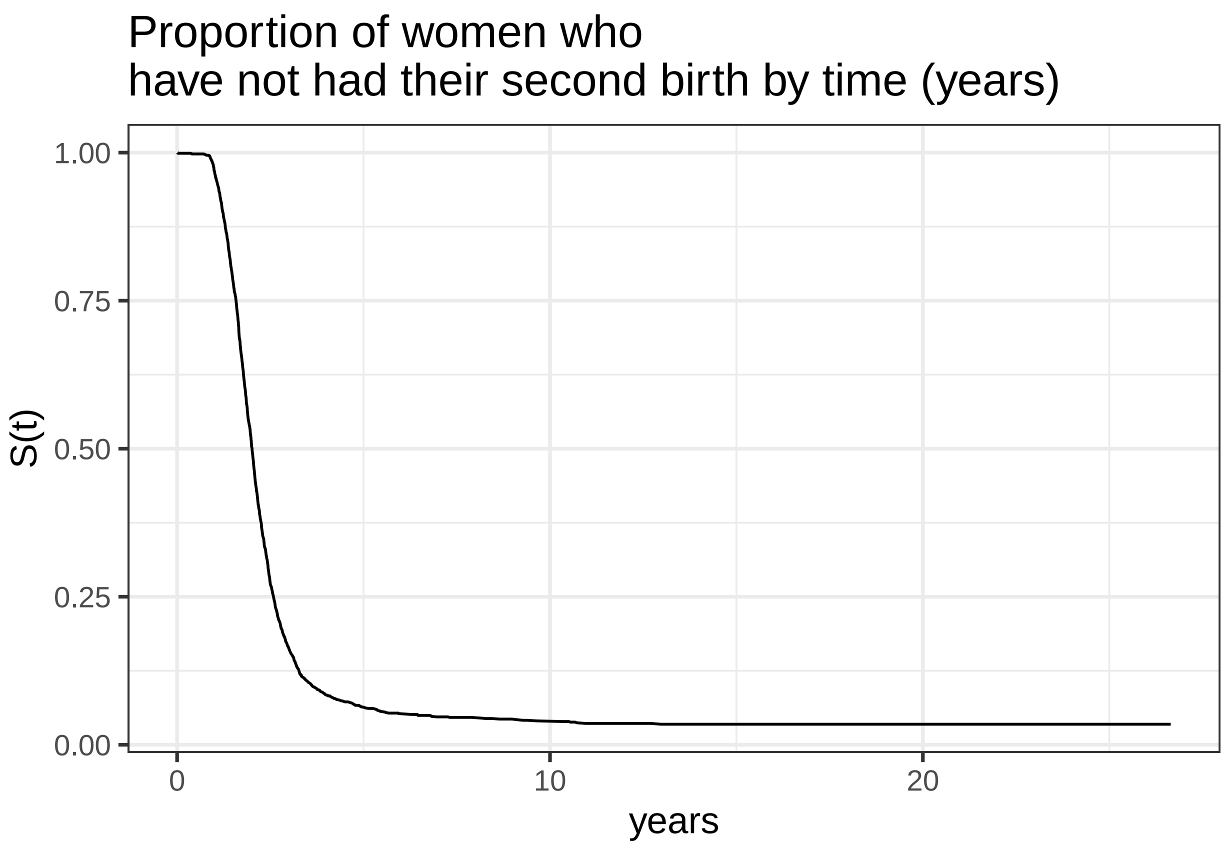
`Surv` objects are set of ordered times with the censors indicated with a plus:

```
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)
```



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.

Question 2

Fill in the gaps below (denoted by XX)

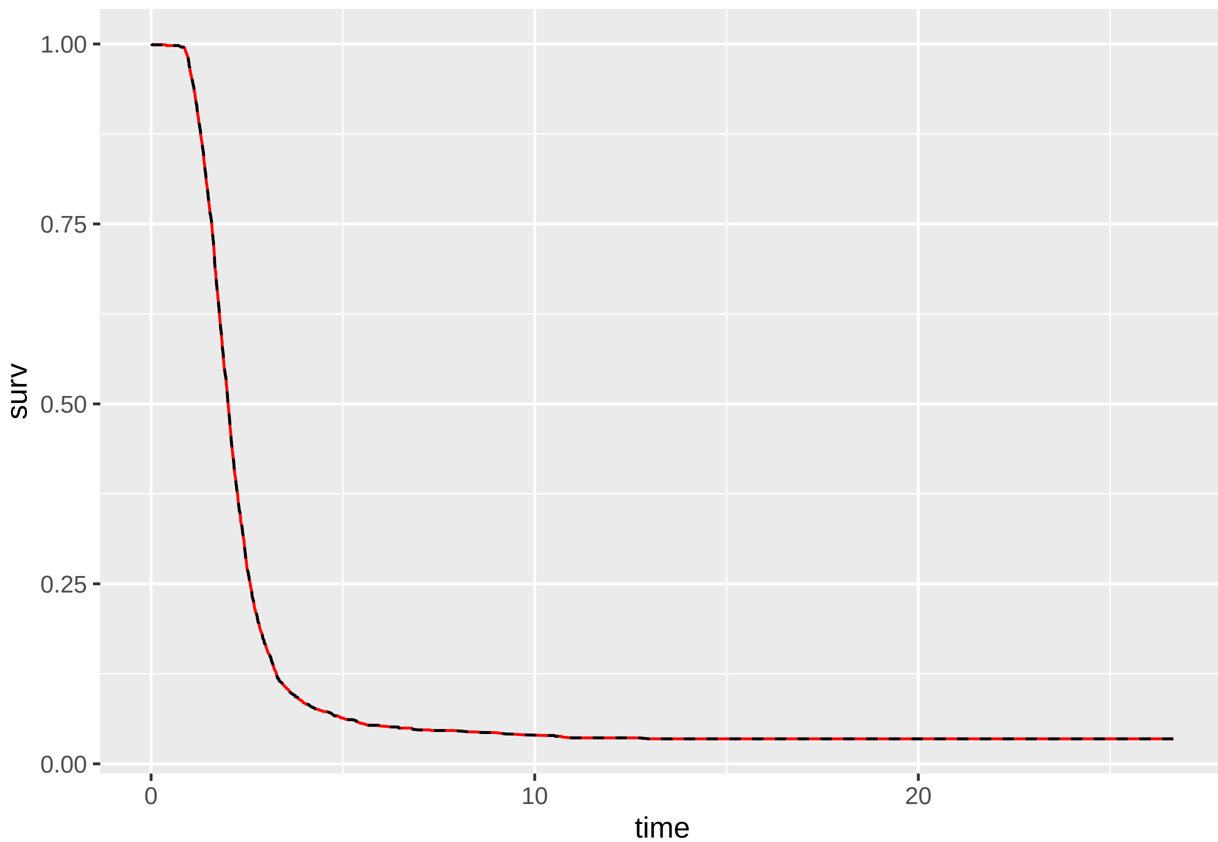
```
n <- nrow(f12)

f12 <- f12 %>%
  arrange(next.ivl) %>% # need to sort by survival times
  mutate(cumulative_people_lost = 1:n(),
         exposure = lag(n - cumulative_people_lost, default = n),
```

```
         prob_death = event/exposure,
         prob_surv = 1-prob_death,
         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, surv), data = f12, lty = 2)
```



Question 3

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

{**Q3-Answer:**}

```
p <- f12 %>% filter(surv<0.75)
cat("\n\nQ6:The proportion of women who have had their 2nd birth within 5 years is",p[1,]$next.ivl,"\n"
```

```
##
##
## Q6:The proportion of women who have had their 2nd birth within 5 years is 1.585
```

Piecewise Constant Hazards

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

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 cumulative_peop...
##    <dbl>  <dbl> <dbl> <dbl>    <dbl> <fct> <fct>  <chr>                 <int>
##  1  1841      1    26  1884    0.969 lower SKL    <30                       1
##  2   456      1    28  1851    1.95  farm... SKL    <30                       2
##  3   942      1    21  1852    0.463 farm... SKL    <30                       3
##  4  1249      1    37  1875    0.778 farm... SKL    30+                       4
##  5   961      1    24  1856    0.126 lower SKL    <30                       5
##  6  1076      1    41  1875    1.81  farm... SKL    30+                       6
##  7  1858      1    34  1898    0.882 farm... SKL    30+                       7
##  8  1644      1    28  1874    0.819 upper SKL    <30                       8
##  9   238      1    27  1845    1.28  farm... SKL    <30                       9
## 10  1704      1    34  1882    1.92  unkn... SKL    30+                      10
## # ... with 7,615 more rows, and 9 more variables: exposure <int>,
## #   prob_death <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(E = sum(event)) %>% select(E) %>% 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
##
## 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,

```r
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:

```r
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
```

Question 4

Confirm that the estimated hazards from the regression are the same as `D/E` in each interval.

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:

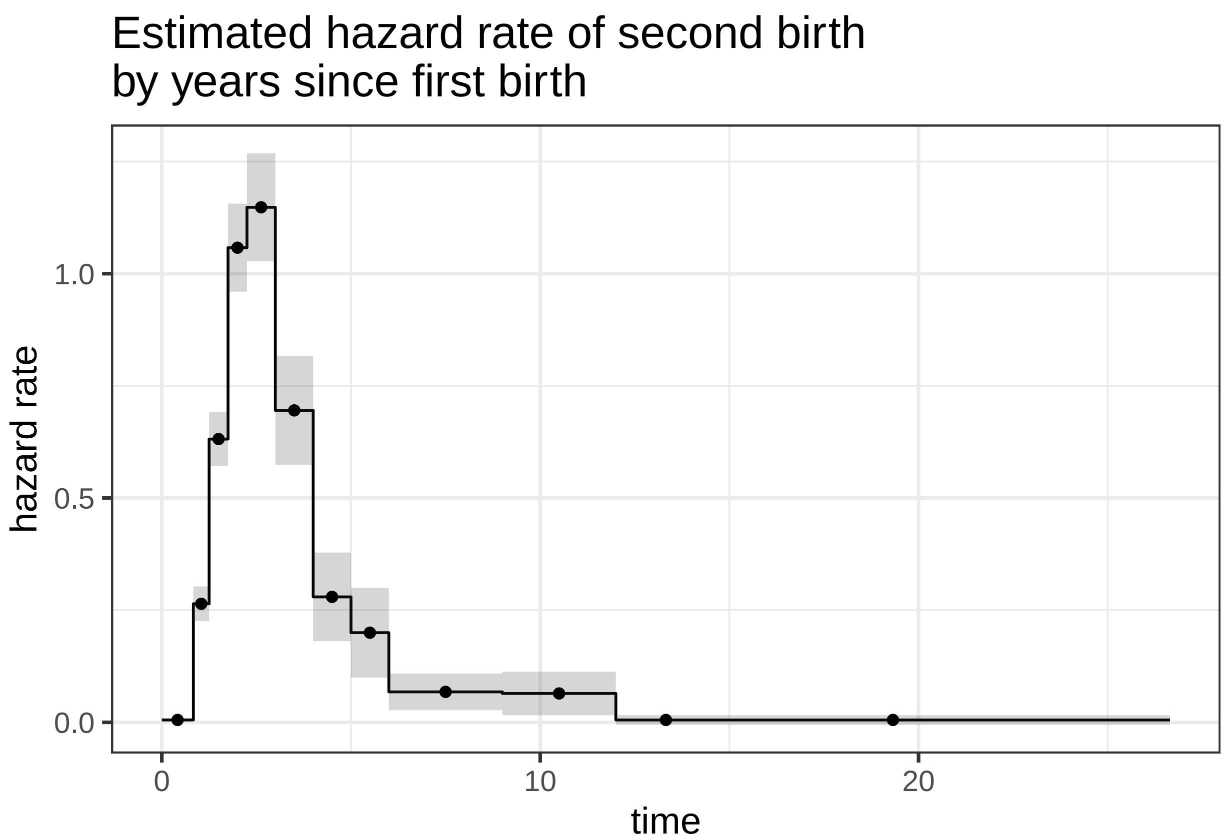
```r
C <- length(cutpoints)+1
cuts <- c(0,cutpoints,max(f12$next.ivl))
hazs <- c(exp(coef(fit_pois)), exp(coef(fit_pois))[C])
ses <- c(sqrt(diag(vcov(fit_pois)))*exp(coef(fit_pois)), sqrt(diag(vcov(fit_pois)))[C]*exp(coef(fit_pois

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")
```



Survival probabilities

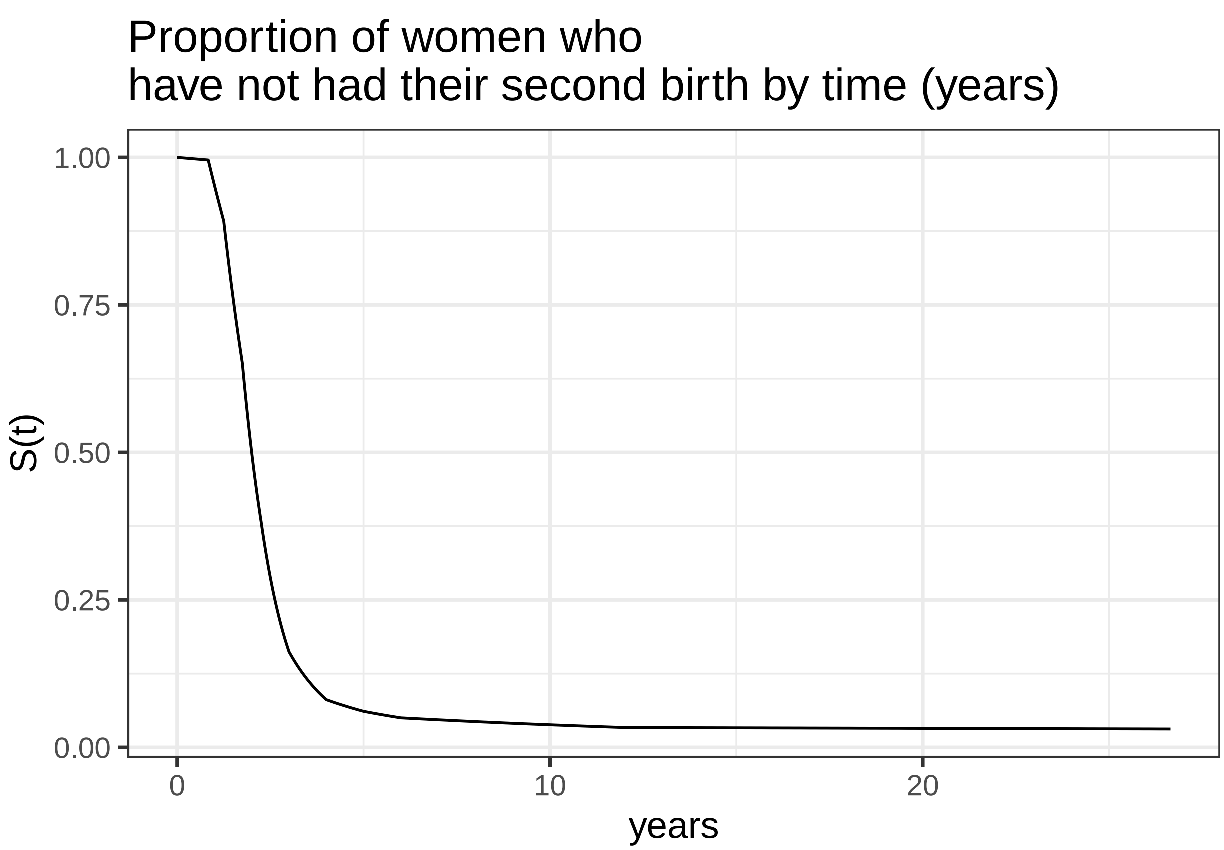
Would be good to also transform these hazards into survival probabilities. Here's a function that does this:

```
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))
  return(tibble(time = t_seq, surv = surv_probs ))
}
```

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)
```



PCH with covariates

Question 5

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).

{**Q5-Answer:**}

```
fit_ind2 <- glm(event ~ offset(log(interval_length))-1 + interval + as.factor(age_group)
                , data=f12_split, family = "poisson")
summary(fit_ind2)
```

```
##
## Call:
## glm(formula = event ~ offset(log(interval_length)) - 1 + interval +
##     as.factor(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 ***
## as.factor(age_group)30+   -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
```

Question 6

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.

{**Q6-Answer:**}

```
cf <- exp(coef(fit_ind2))
lambdas <- cf[1:length(cf)-1]
cuts <- c(0, cutpoints, max(f12_split$next.ivl))
```

```
df_surv <- survival_prob(lambdas = lambdas, cuts = cuts)

p <- df_surv %>%
  filter(time<=5.0) %>%
  arrange(-time)
cat("\n\nQ6:The proportion of women who have had their 2nd birth within 5 years is",1-p$surv[1],"\n")
```

```
##
##
## Q6:The proportion of women who have had their 2nd birth within 5 years is 0.9568996
```