DSTI Survey - Daisuke KUWABARA & Nesrine BENANTEUR

Disclaimer: the pdf report is more than 10 pages, because we decided to knit it directly from R Studio: lots of pages have huge blank spaces, also we couldn't find a way to reduce the size of the graphs, or of the outputs of certain functions. Other than that, our study would've probably fit the limit in pages. Please don't penalize us :(.

```
library(tidyverse)
library(lubridate)
library(broom)
library(survival)
library(ggplot2)
library(survminer)
library(ranger)
library(asaur)
```

Data import

```
raw <- read_csv("DSTI_survey.csv")
```

```
################################### INTERNSHIP ANALYSIS ###############
```

Your report should answer these basic questions:

How many students partecipated in the interview?

After data preparation, how many samples are usable for data analysis? How many samples were dropped (if any), and why?

How long does it take to obtain an internship? Please report the median time (with a confidence interval), total number of students at the baseline, the total number of events observed, and the total number of censored observations.

Of these variables, which ones have the most impact on the time to obtain an internship, and in which direction: cohort, age, educational background, having or not having children.

Bonus question: can you build a predictive model to identify students at high risk of a long search? How well does your model perform?

- Number of students participating in the interview:

```
nrow(raw)
```

```
## [1] 82
```

82 students participated in the interview.

DATA PREPARATION RELATIVE TO FINDING AN INTERNSHIP

Have you found an intership?

```
table(raw$`Have you found an internship?`, useNA = "always")
```

```
##
##   No  Yes <NA>
##   49   26    7
```

```
d_foundInt <- raw %>%
  mutate(foundInt = `Have you found an internship?` != "No")
table(d_foundInt$foundInt)
```

```
##
## FALSE  TRUE
##    49    26
```

Time taken to find an intership?

```
d_searchtime <-
  raw %>%
  mutate(sd = as.POSIXct(raw$`When did you start looking for an internship`, format = "%m/%d/%Y"),
         ed = as.POSIXct(raw$`When did you stopped looking for an internship`, format = "%m/%d/%Y"),
         st = 12 * (year(ed) - year(sd)) + (month(ed) - month(sd)))
```

```
table(raw$`When did you stopped looking for an internship`, useNA = "always")
```

```
##
##   1/1/2020  10/1/2020 10/10/2020 10/11/2019 10/19/2020 10/30/2020 10/31/2020
##          1          2          1          1          1          1          1
##  11/1/2020  11/2/2020 12/31/2018   2/1/2020  2/28/2020   3/1/2020  3/13/2020
##          1          2          1          2          1          1          1
##  3/15/2020   3/8/2020  4/15/2021   5/1/1980  5/11/2018   6/1/2021  6/26/2015
##          1          1          1          1          1          1          1
##   6/9/2020   7/1/2020  7/31/2020  9/29/2020       <NA>
##          1          1          1          1         54
```

```
d_searchtime <- d_searchtime %>%
  mutate(foundInt = `Have you found an internship?` != "No")
d_searchtime %>% mutate_if(is.character, as.numeric)
```

```
## Warning in mask$eval_all_mutate(quo): NAs introduits lors de la conversion
## automatique

## Warning in mask$eval_all_mutate(quo): NAs introduits lors de la conversion
```

```
## automatique

## Warning in mask$eval_all_mutate(quo): NAs introduits lors de la conversion
## automatique

## Warning in mask$eval_all_mutate(quo): NAs introduits lors de la conversion
## automatique

## Warning in mask$eval_all_mutate(quo): NAs introduits lors de la conversion
## automatique

## Warning in mask$eval_all_mutate(quo): NAs introduits lors de la conversion
## automatique

## Warning in mask$eval_all_mutate(quo): NAs introduits lors de la conversion
## automatique

## Warning in mask$eval_all_mutate(quo): NAs introduits lors de la conversion
## automatique

## Warning in mask$eval_all_mutate(quo): NAs introduits lors de la conversion
## automatique

## # A tibble: 82 x 17
##    Timestamp `Year of birth` `Were you ever a~ `Year when firs~ `Year when stop~
##        <dbl>           <dbl>             <dbl>            <dbl>            <dbl>
##  1        NA            1992                NA               NA               NA
##  2        NA            1993                NA             2011               NA
##  3        NA            1990                NA               NA               NA
##  4        NA            1986                NA               NA               NA
##  5        NA            1993                NA               NA               NA
##  6        NA            1992                NA             2019               NA
##  7        NA            1995                NA               NA               NA
##  8        NA            1992                NA             2010               NA
##  9        NA            1993                NA             2013             2018
## 10        NA            1989                NA               NA               NA
## # ... with 72 more rows, and 12 more variables:
## #   When did you start looking for an internship <dbl>, Sex <dbl>,
## #   When did you stopped looking for an internship <dbl>,
## #   Have you found an internship? <dbl>,
## #   Education: background (pick a main one you identify with) <dbl>,
## #   Years of education <dbl>, Do you have children? <dbl>, Cohort <dbl>,
## #   sd <dttm>, ed <dttm>, st <dbl>, foundInt <lgl>
```

```
table(d_searchtime$foundInt)
```

```
##
## FALSE  TRUE
##    49    26
```

Total number of students at the baseline: 82

Total number of events observed: 24

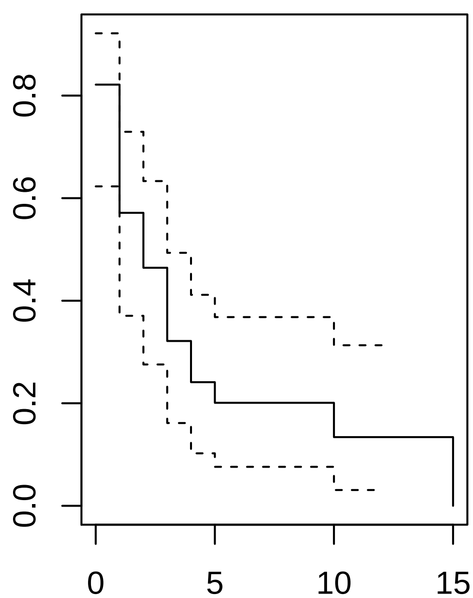
Total number of censored observations: 58 students did not give the date they stopped looking for an internship.

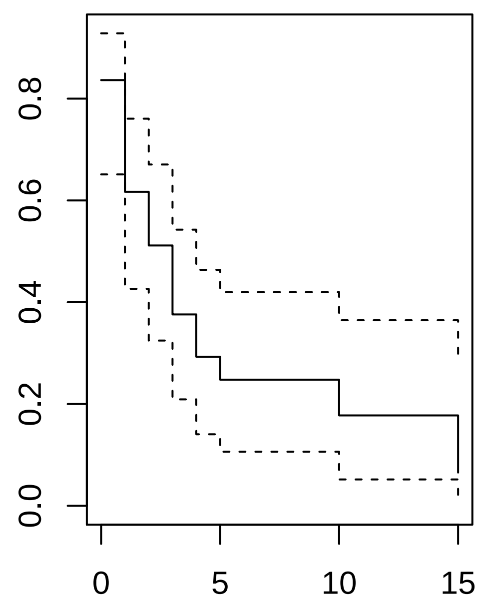
We will then use non-parametric methods to study the survival, taking into consideration that some data is missing, so that we don't have to drop any data on our side.

How long does it take to obtain an internship?

```
fit.KM <- survfit(Surv(st, foundInt) ~ 1, data = d_searchtime, type='kaplan-meier',conf.type='log-log' )
fit.NA <- survfit(Surv(st, foundInt) ~ 1, data = d_searchtime, type='fleming-harrington',conf.type='log-

par(mfrow = c(1, 2))
plot(fit.KM)
plot(fit.NA)
```






```
summary(fit.KM)
```

```
## Call: survfit(formula = Surv(st, foundInt) ~ 1, data = d_searchtime,
##     type = "kaplan-meier", conf.type = "log-log")
##
## 54 observations deleted due to missingness
##  time n.risk n.event survival std.err lower 95% CI upper 95% CI
##     0     28       5    0.821  0.0724       0.6230        0.921
```

```
##     1     23       7    0.571  0.0935       0.3706        0.729
##     2     16       3    0.464  0.0942       0.2756        0.633
##     3     13       4    0.321  0.0883       0.1615        0.493
##     4      8       2    0.241  0.0825       0.1023        0.411
##     5      6       1    0.201  0.0779       0.0760        0.368
##    10      3       1    0.134  0.0754       0.0308        0.313
##    15      1       1    0.000     NaN           NA           NA
```

```r
fit.KM
```

```
## Call: survfit(formula = Surv(st, foundInt) ~ 1, data = d_searchtime,
##     type = "kaplan-meier", conf.type = "log-log")
##
##    54 observations deleted due to missingness
##       n  events  median 0.95LCL 0.95UCL
##      28      24       2       1       3
```

The Kaplan-Meier estimator tells us that the median time to find an intership is 2 months, with a confidence interval CI= 1-3 months.

```r
fit.NA
```

```
## Call: survfit(formula = Surv(st, foundInt) ~ 1, data = d_searchtime,
##     type = "fleming-harrington", conf.type = "log-log")
##
##    54 observations deleted due to missingness
##       n  events  median 0.95LCL 0.95UCL
##      28      24       3       1       4
```

The Fleming-Harrington estimator tells us that the median time to find an intership is 3 months, with a confidence interval CI= 1-4 months.

Of these variables, which ones have the most impact on the time to obtain an internship? and in which direction: cohort, age, educational background, having or not having children.

Impact of having children or not on the search Period

```r
table(raw$`Do you have children?`, useNA = "always")
```

```
##
##   No  Yes <NA>
##   58   24    0
```

```r
d_searchtime <- d_searchtime %>%
  mutate(children = `Do you have children?` != "No")
d_searchtime %>% mutate_if(is.factor, as.numeric)
```

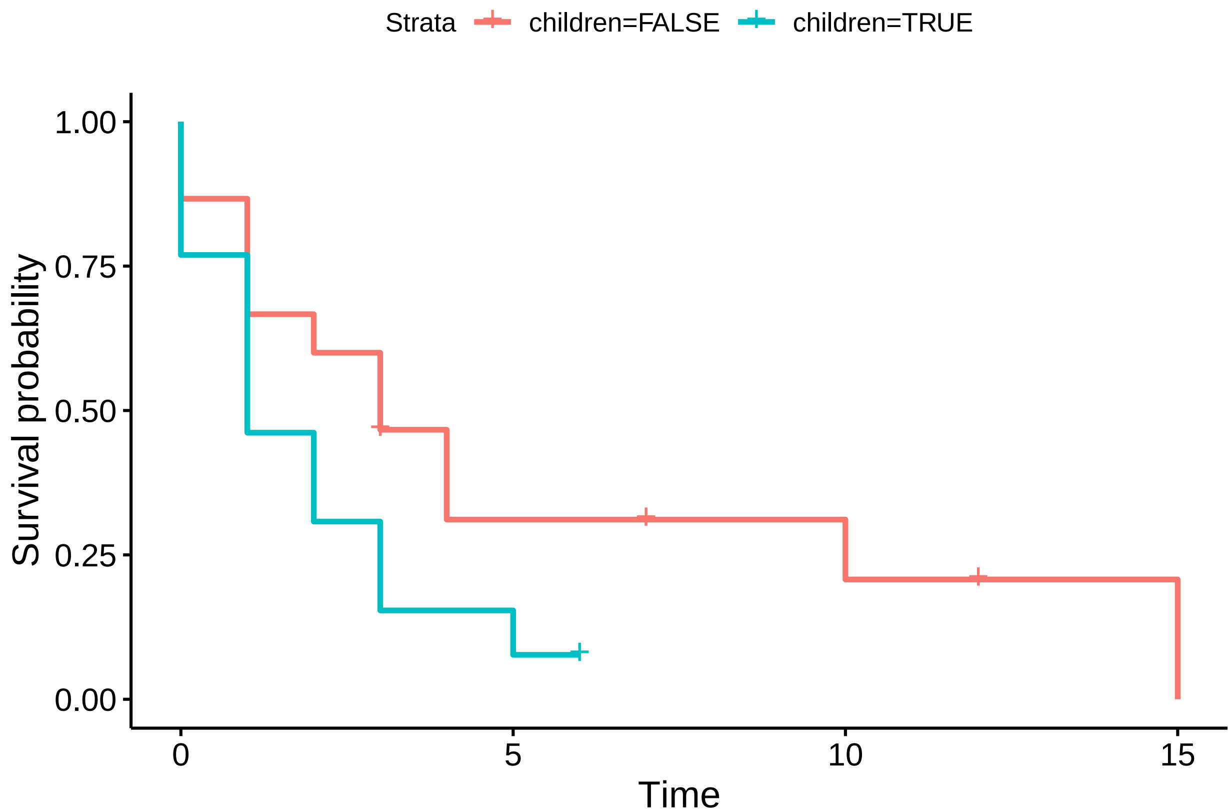
```
## # A tibble: 82 x 18
##    Timestamp  `Year of birth` `Were you ever ~ `Year when firs~ `Year when stop~
##    <chr>                <dbl> <chr>                       <dbl>            <dbl>
##  1 11/2/2020~            1992 No                             NA               NA
##  2 11/2/2020~            1993 Yes, and I'm cu~             2011               NA
##  3 11/2/2020~            1990 No                             NA               NA
##  4 11/2/2020~            1986 No                             NA               NA
##  5 11/2/2020~            1993 No                             NA               NA
##  6 11/2/2020~            1992 Yes, and I'm cu~             2019               NA
##  7 11/2/2020~            1995 No                             NA               NA
##  8 11/2/2020~            1992 Yes, and I'm cu~             2010               NA
##  9 11/2/2020~            1993 Yes, and I stop~             2013             2018
## 10 11/2/2020~            1989 No                             NA               NA
## # ... with 72 more rows, and 13 more variables:
## #   When did you start looking for an internship <chr>, Sex <chr>,
## #   When did you stopped looking for an internship <chr>,
## #   Have you found an internship? <chr>,
## #   Education: background (pick a main one you identify with) <chr>,
## #   Years of education <dbl>, Do you have children? <chr>, Cohort <chr>,
## #   sd <dttm>, ed <dttm>, st <dbl>, foundInt <lgl>, children <lgl>
```

```
table(d_searchtime$children)
```

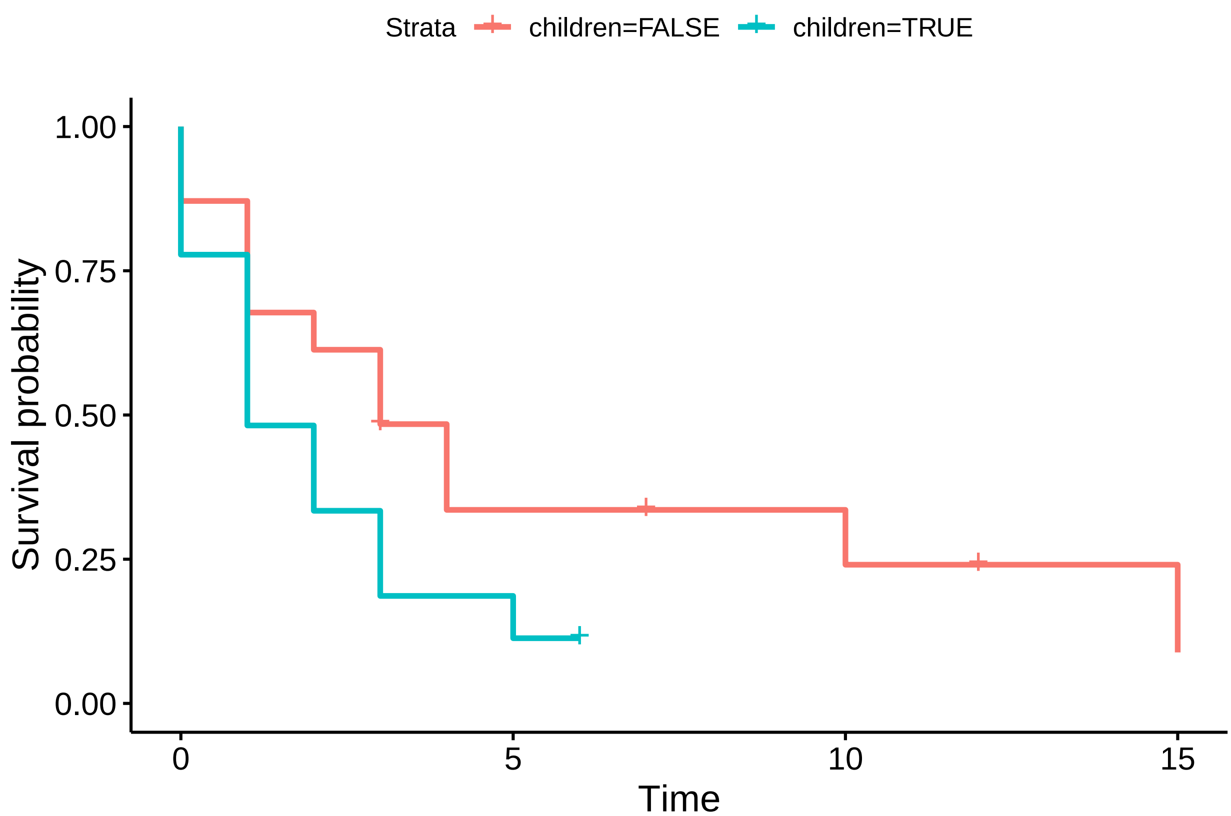
```
##
## FALSE  TRUE
##    58    24
```

```
fit.KM.children <- survfit(Surv(st, foundInt) ~ children, data = d_searchtime, type='kaplan-meier',conf
fit.NA.children <- survfit(Surv(st, foundInt) ~ children, data = d_searchtime, type='fh',conf.type='log
```

```
ggsurvplot(fit.KM.children)
```




```
ggsurvplot(fit.NA.children)
```



The Kaplan-Meier and Fleming-Harrington estimators seem to give the same survival curves.

```
survdiff <- survdiff(Surv(st, foundInt) ~ children, data = d_searchtime)
survdiff
```

```
## Call:
## survdiff(formula = Surv(st, foundInt) ~ children, data = d_searchtime)
##
## n=28, 54 observations deleted due to missingness.
##
##                 N Observed Expected (O-E)^2/E (O-E)^2/V
## children=FALSE 15       12    15.45     0.769      2.89
## children=TRUE  13       12     8.55     1.388      2.89
##
##  Chisq= 2.9  on 1 degrees of freedom, p= 0.09
```

The log rank test for difference in survival gives a p-value of $p = 0.09$, indicating that having children or not doesn't seem to influence significantly the duration of the internship search.

However, even if the test results are not significant, we can see a tendency that seems to show that students having children seem to find intership quicker than students without children.

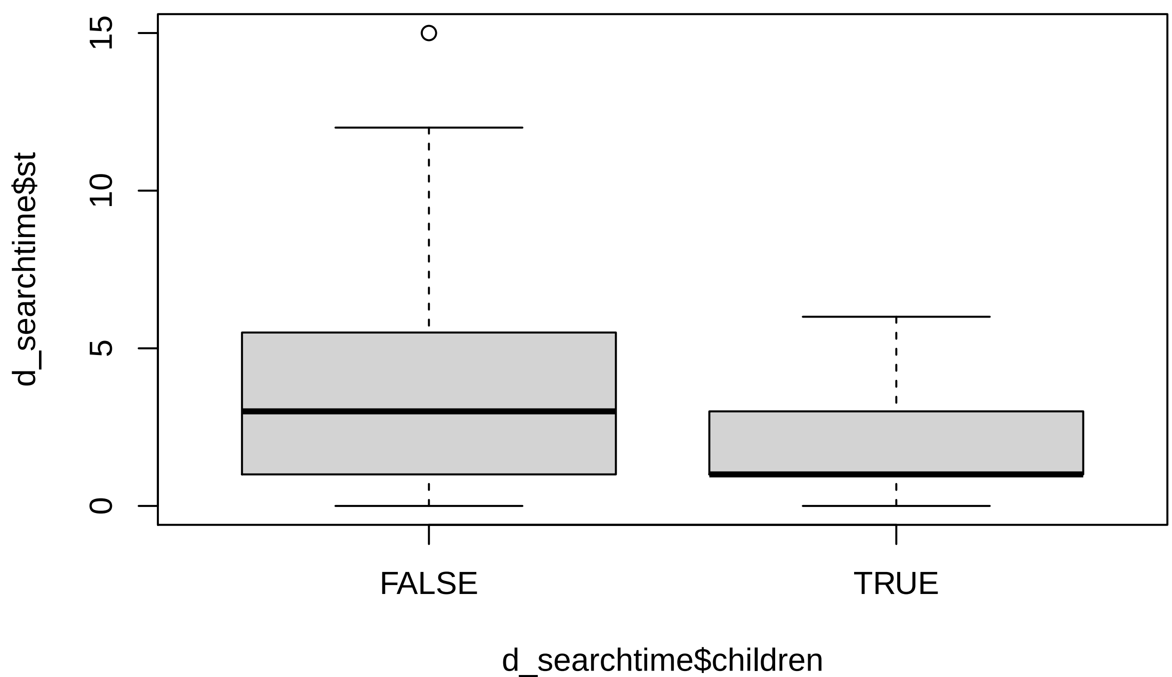
Let's check if this tendency might be due to an outlier:

```
anova(lm(st~children, data=d_searchtime))
```

```
## Analysis of Variance Table
```

```
##
## Response: st
##           Df Sum Sq Mean Sq F value  Pr(>F)
## children   1  42.73  42.727  3.3011 0.08078 .
## Residuals 26 336.52  12.943
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
boxplot(d_searchtime$st~d_searchtime$children)
```



The boxplot shows an outlier (15 months for the search time). Let's see if the tendency changes when we get rid of it:

```
d_searchtime<-d_searchtime[!(d_searchtime$st==15),]
d_searchtime
```

```
## # A tibble: 81 x 18
##    Timestamp  `Year of birth` `Were you ever ~ `Year when firs~ `Year when stop~
##    <chr>                <dbl> <chr>                       <dbl>            <dbl>
##  1 <NA>                    NA <NA>                           NA               NA
##  2 <NA>                    NA <NA>                           NA               NA
##  3 <NA>                    NA <NA>                           NA               NA
##  4 11/2/2020~            1986 No                             NA               NA
##  5 11/2/2020~            1993 No                             NA               NA
##  6 11/2/2020~            1992 Yes, and I'm cu~             2019               NA
##  7 11/2/2020~            1995 No                             NA               NA
```

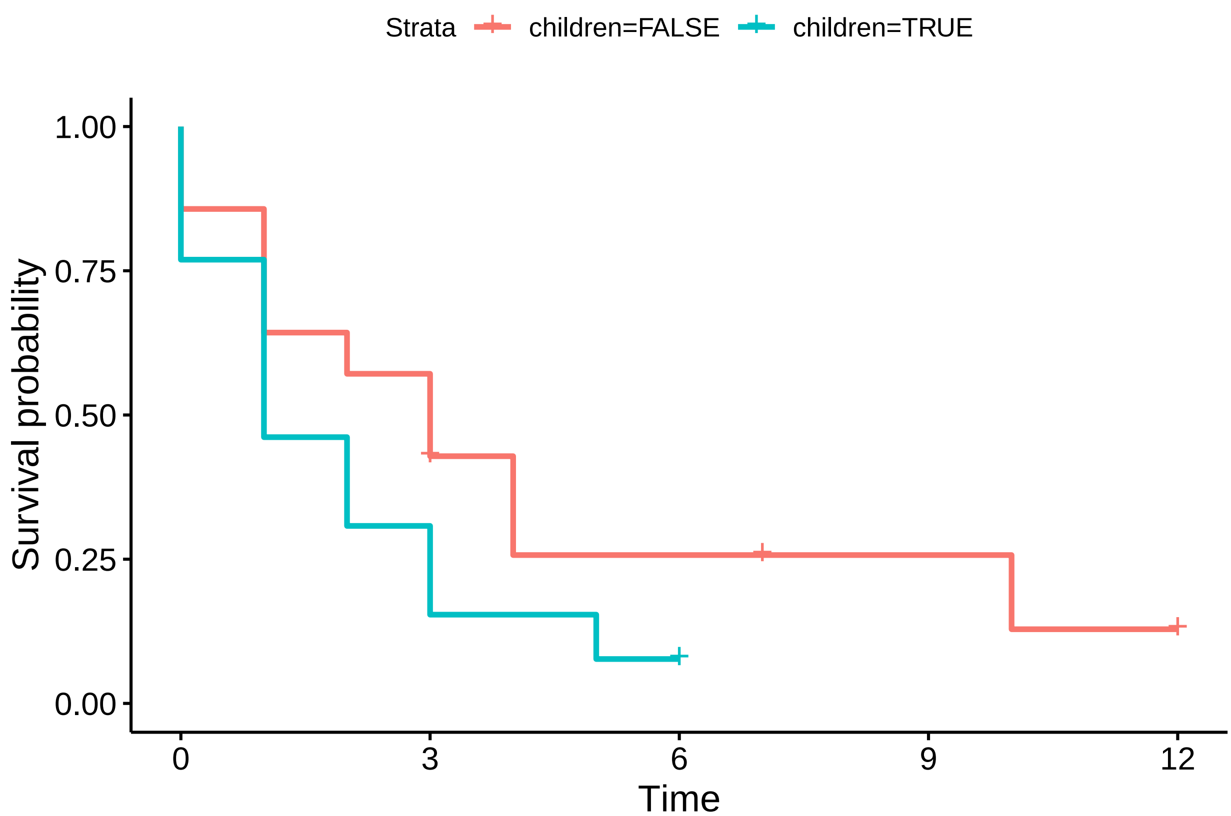
```
##  8 <NA>                    NA <NA>                             NA                NA
##  9 <NA>                    NA <NA>                             NA                NA
## 10 <NA>                    NA <NA>                             NA                NA
## # ... with 71 more rows, and 13 more variables:
## #   When did you start looking for an internship <chr>, Sex <chr>,
## #   When did you stopped looking for an internship <chr>,
## #   Have you found an internship? <chr>,
## #   Education: background (pick a main one you identify with) <chr>,
## #   Years of education <dbl>, Do you have children? <chr>, Cohort <chr>,
## #   sd <dttm>, ed <dttm>, st <dbl>, foundInt <lgl>, children <lgl>
```

```
survdiff <- survdiff(Surv(st, foundInt) ~ children, data = d_searchtime)
survdiff
```

```
## Call:
## survdiff(formula = Surv(st, foundInt) ~ children, data = d_searchtime)
##
## n=27, 54 observations deleted due to missingness.
##
##                  N Observed Expected (O-E)^2/E (O-E)^2/V
## children=FALSE 14       11    13.91     0.607      2.04
## children=TRUE  13       12     9.09     0.929      2.04
##
##  Chisq= 2  on 1 degrees of freedom, p= 0.2
```

The log-rank test shows that the results are even less significant than previously, meaning that the tendency observed might be due to this outlier.

```
fit.KM.children2 <- survfit(Surv(st, foundInt) ~ children, data = d_searchtime, type='kaplan-meier',con
ggsurvplot(fit.KM.children2)
```



Still, we can see this tendency graphically.

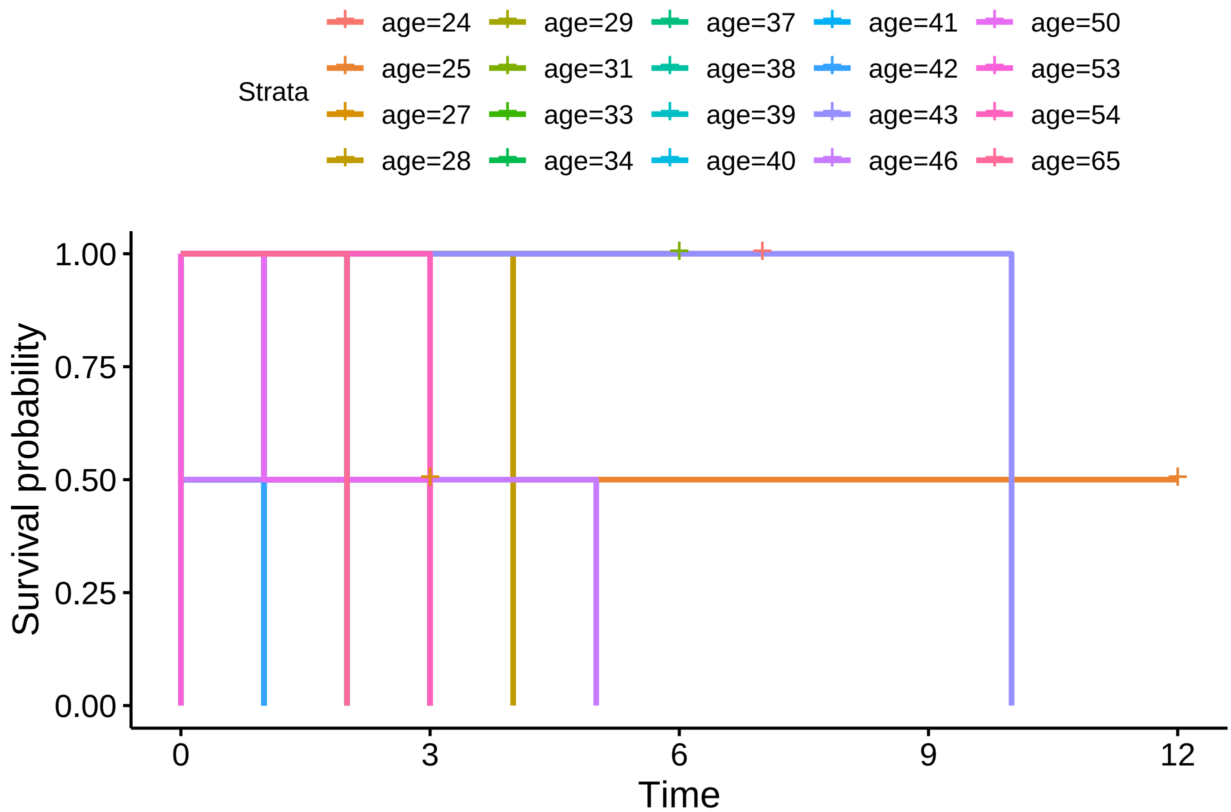
Impact of the age on the search Period

```
d_searchtime <-
  d_searchtime %>%
  mutate(Timestamp = as.POSIXct(Timestamp, format = "%d/%m/%Y %H:%M:%OS"),
         age = year(Timestamp) - `Year of birth`)
table(d_searchtime$age)
```

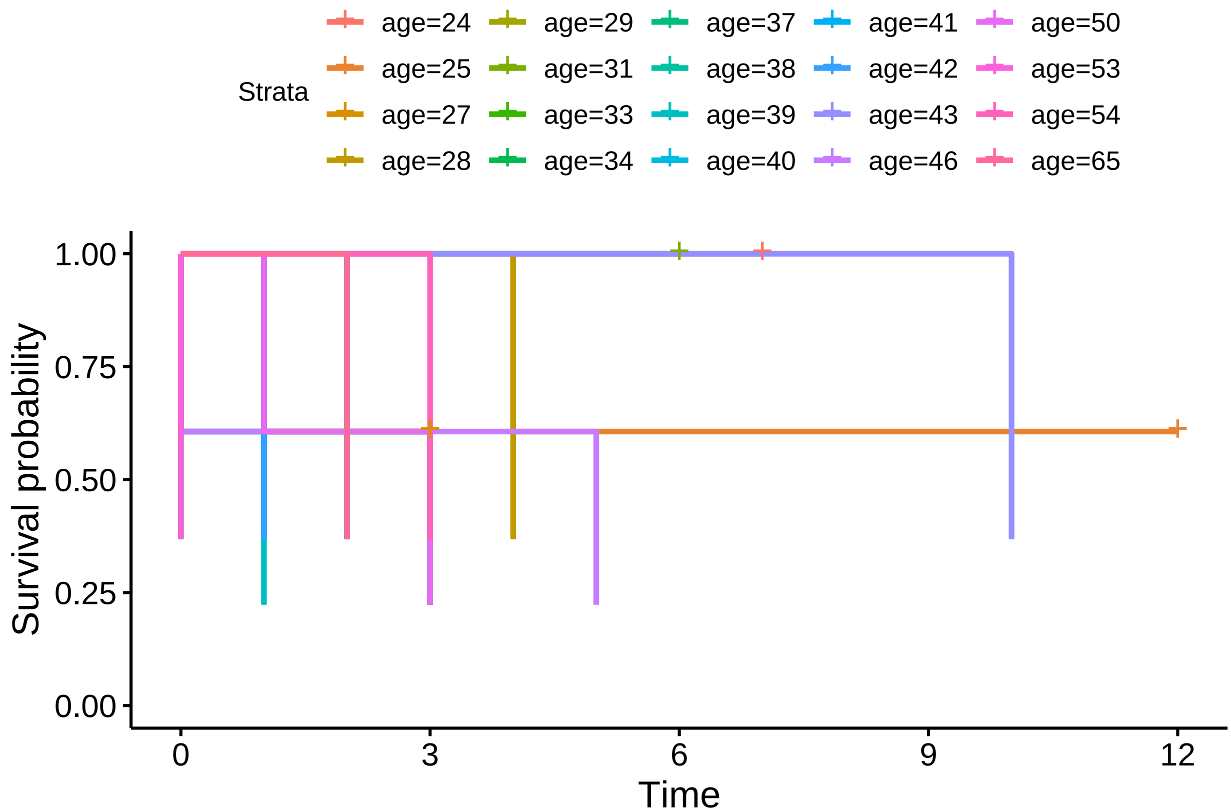
```
##
## 24 25 27 28 29 31 33 34 37 38 39 40 41 42 43 46 50 53 54 65
##  1  2  2  1  2  1  1  1  1  2  2  1  1  1  1  2  2  1  1  1
```

```
fit.KM.age <- survfit(Surv(st, foundInt) ~ age, data = d_searchtime, type='kaplan-meier',conf.type='log
fit.NA.age <- survfit(Surv(st, foundInt) ~ age, data = d_searchtime, type='fh',conf.type='log-log' )
```

```
par(mfrow = c(1, 2))
ggsurvplot(fit.KM.age)
```




```
ggsurvplot(fit.NA.age)
```




```
survdiff <- survdiff(Surv(st, foundInt) ~ age, data = d_searchtime)
survdiff
```

```
## Call:
## survdiff(formula = Surv(st, foundInt) ~ age, data = d_searchtime)
##
## n=27, 54 observations deleted due to missingness.
##
##         N Observed Expected (O-E)^2/E (O-E)^2/V
## age=24 1        0    1.522    1.5224    2.0803
## age=25 2        1    3.345    1.6438    2.5517
## age=27 2        1    1.540    0.1894    0.2687
## age=28 1        1    1.322    0.0786    0.1089
## age=29 2        2    1.540    0.1374    0.1949
## age=31 1        0    1.522    1.5224    2.0803
## age=33 1        1    0.703    0.1251    0.1667
## age=34 1        1    0.503    0.4900    0.6705
## age=37 1        1    0.185    3.5852    4.4000
## age=38 2        2    1.222    0.4955    0.6808
## age=39 2        2    0.689    2.4978    3.3915
## age=40 1        1    0.503    0.4900    0.6705
## age=41 1        1    0.703    0.1251    0.1667
## age=42 1        1    0.503    0.4900    0.6705
## age=43 1        1    2.022    0.5169    0.7663
## age=46 2        2    1.708    0.0501    0.0682
## age=50 2        2    1.540    0.1374    0.1949
```

```
## age=53 1         1    0.185    3.5852    4.4000
## age=54 1         1    1.037    0.0013    0.0018
## age=65 1         1    0.703    0.1251    0.1667
##
##  Chisq= 27.9  on 19 degrees of freedom, p= 0.08
```

The log rank test for difference in survival gives a p-value of p = 0.08, indicating the age doesn't seem to influence significantly the duration of the internship search.

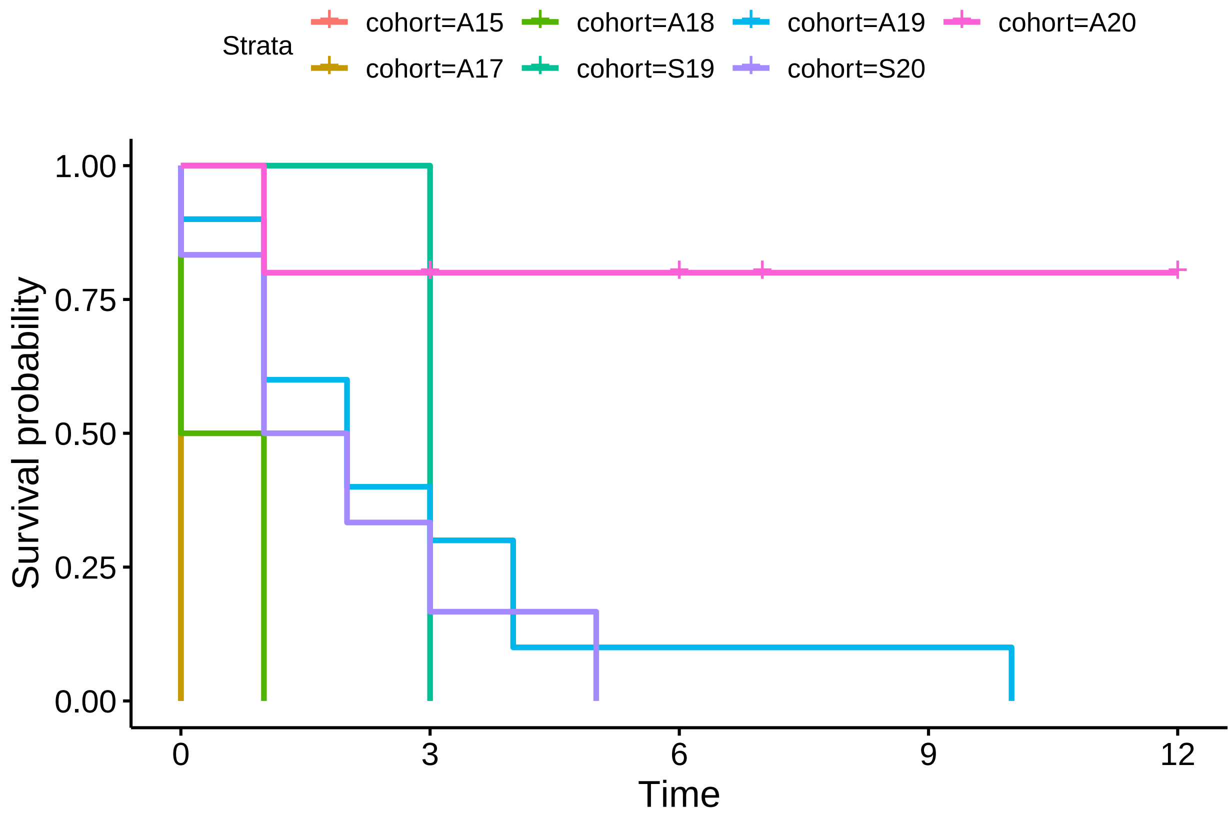
Cohort - Search Period

```
d_searchtime <- d_searchtime %>%
  mutate(
    cohort = factor(Cohort,
                    levels = paste0(c("S", "A"), rep(15:20, each = 2)))
  )
table(d_searchtime$cohort, useNA = "always")
```

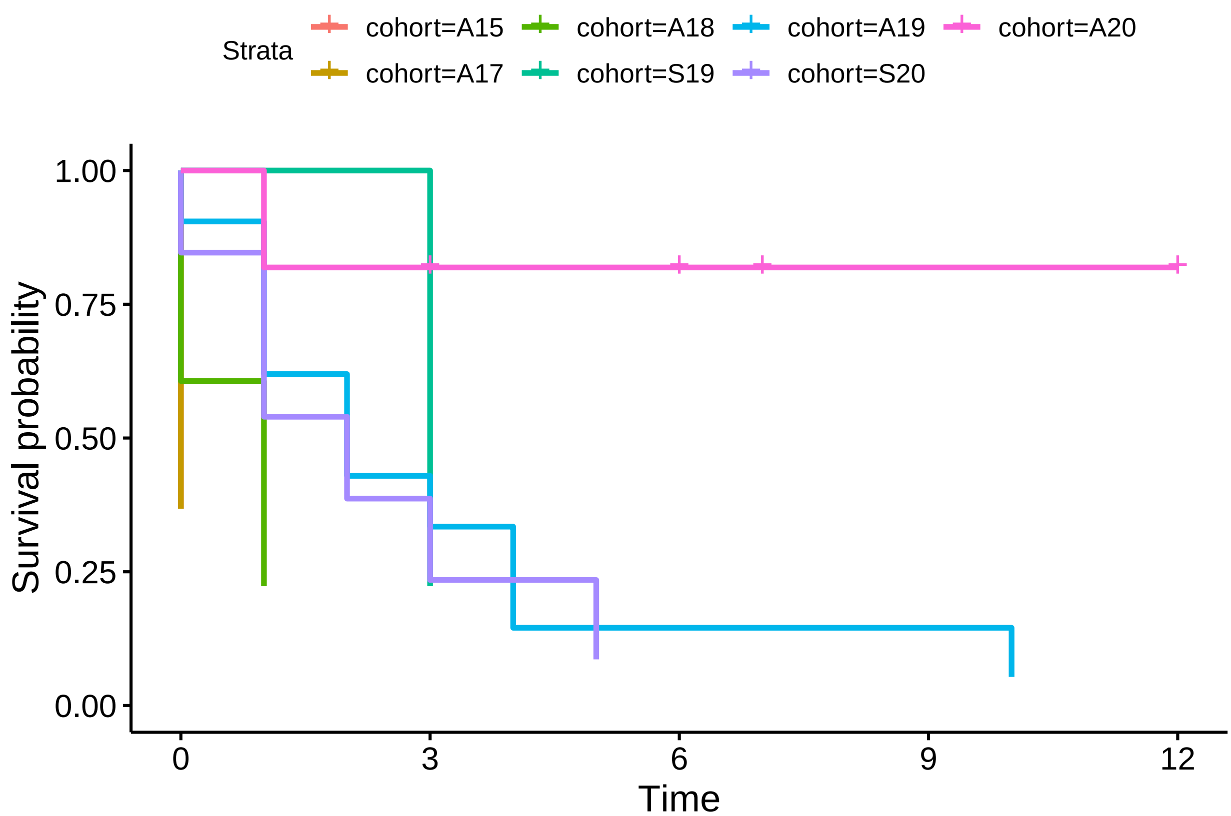
```
##
##  S15  A15  S16  A16  S17  A17  S18  A18  S19  A19  S20  A20 <NA>
##    0    1    0    0    0    1    0    2    2   10    6    5   54
```

```
fit.KM.cohort <- survfit(Surv(st, foundInt) ~ cohort, data = d_searchtime, type='kaplan-meier',conf.typ
fit.NA.cohort <- survfit(Surv(st, foundInt) ~ cohort, data = d_searchtime, type='fh',conf.type='log-log
```

```
ggsurvplot(fit.KM.cohort)
```




```
ggsurvplot(fit.NA.cohort)
```




```r
survdiff <- survdiff(Surv(st, foundInt) ~ cohort, data = d_searchtime)
survdiff
```

```
## Call:
## survdiff(formula = Surv(st, foundInt) ~ cohort, data = d_searchtime)
##
## n=27, 54 observations deleted due to missingness.
##
##               N Observed Expected (O-E)^2/E (O-E)^2/V
## cohort=A15  1        1    0.185    3.5852   4.40000
## cohort=A17  1        1    0.185    3.5852   4.40000
## cohort=A18  2        2    0.689    2.4978   3.39151
## cohort=S19  2        2    2.073    0.0026   0.00384
## cohort=A19 10       10    8.806    0.1619   0.33334
## cohort=S20  6        6    4.454    0.5363   0.85657
## cohort=A20  5        1    6.607    4.7587   8.82216
##
##  Chisq= 20.6  on 6 degrees of freedom, p= 0.002
```

The log rank test for difference in survival gives a p-value of p = 0.002, indicating that the cohort groups differ significantly in survival when it comes to finding an internship: indeed, it seems that cohort A15, A17, A18 and S20 found their interships faster than the other cohorts.

However, we can see that the number of people in cohorts A15,17 and 18 is quite low, it would maybe be relevant to drop those observation to see if we get the same results.

Impact of the educational background on the search Period

```
table(d_searchtime$`Education: background (pick a main one you identify with)`, useNA = "always")
```

```
##
##                                         Business, Management
##                                                            3
##                                             Finance, Economy
##                                                            1
##                               Literature, History, Philosophy
##                                                            1
## Mathematics, Physics, Chemistry, Computer Science, Statistics
##                                                           19
##                                             Medicine, Biology
##                                                            2
##                                                        Other
##                                                            1
##                                                         <NA>
##                                                           54
```

Let's first come up with shorter labels.

```
edu_labels <- tibble(
  `Education: background (pick a main one you identify with)` =
    c("Business, Management", "Finance, Economy",
      "Literature, History, Philosophy",
      "Mathematics, Physics, Chemistry, Computer Science, Statistics",
      "Medicine, Biology", "Other"),
  education = c("mgmt", "fin", "lit", "math", "bio", "oth")
  )

edu_labels
```

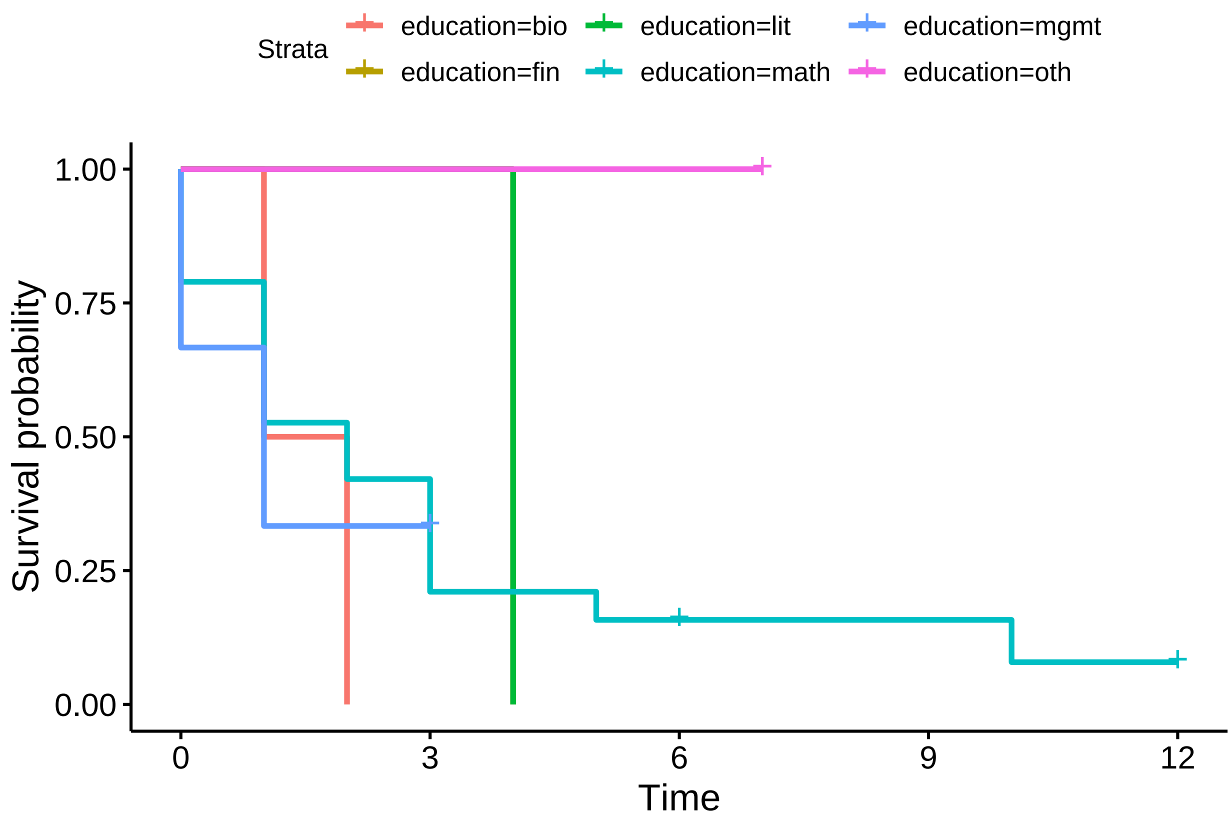
```
## # A tibble: 6 x 2
##   `Education: background (pick a main one you identify with)`   education
##   <chr>                                                         <chr>
## 1 Business, Management                                          mgmt
## 2 Finance, Economy                                              fin
## 3 Literature, History, Philosophy                               lit
## 4 Mathematics, Physics, Chemistry, Computer Science, Statistics math
## 5 Medicine, Biology                                             bio
## 6 Other                                                         oth
```

```
d_searchtime <- d_searchtime %>%
  inner_join(edu_labels, by = "Education: background (pick a main one you identify with)") %>%
  mutate(education = factor(education))
table(d_searchtime$education, useNA = "always")
```

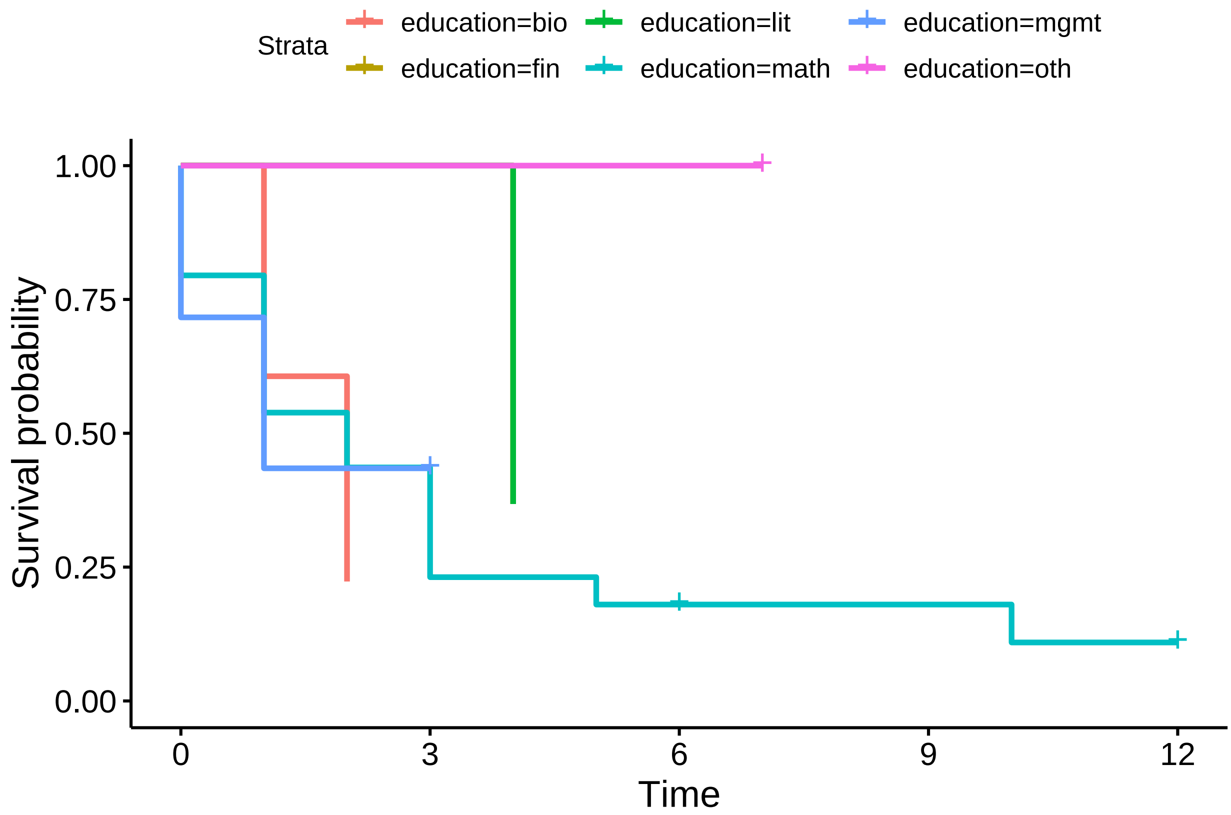
```
##
##  bio  fin  lit math mgmt  oth <NA>
##    2    1    1   19    3    1    0
```

```r
fit.KM.education <- survfit(Surv(st, foundInt) ~ education, data = d_searchtime, type='kaplan-meier',co
fit.NA.education <- survfit(Surv(st, foundInt) ~ education, data = d_searchtime, type='fh',conf.type='l
```

```r
ggsurvplot(fit.KM.education)
```




```r
ggsurvplot(fit.NA.education)
```




```
survdiff <- survdiff(Surv(st, foundInt) ~ education, data = d_searchtime)
survdiff
```

```
## Call:
## survdiff(formula = Surv(st, foundInt) ~ education, data = d_searchtime)
##
##                 N Observed Expected (O-E)^2/E (O-E)^2/V
## education=bio   2        2     1.21    0.5215    0.7288
## education=fin   1        1     1.32    0.0786    0.1089
## education=lit   1        1     1.32    0.0786    0.1089
## education=math 19       17    15.90    0.0760    0.3220
## education=mgmt  3        2     1.73    0.0438    0.0622
## education=oth   1        0     1.52    1.5224    2.0803
##
##  Chisq= 3.1  on 5 degrees of freedom, p= 0.7
```

The log rank test for difference in survival gives a p-value of $p = 0.7$, indicating the educational background doesn't seem to influence significantly the duration of the internship search.

Bonus question: can you build a predictive model to identify students at high risk of a long search? How well does your model perform?

Years of education

```
d_searchtime <- d_searchtime %>%
  mutate(edu_years = `Years of education`)
d_searchtime %>% mutate_if(is.factor, as.numeric)
```

```
## # A tibble: 27 x 22
##    Timestamp           `Year of birth` `Were you ever a sm~ `Year when first st~
##    <dttm>                        <dbl> <chr>                               <dbl>
##  1 2020-02-11 16:59:45            1986 No                                     NA
##  2 2020-02-11 17:00:00            1993 No                                     NA
##  3 2020-02-11 17:00:02            1992 Yes, and I'm curren~                 2019
##  4 2020-02-11 17:00:09            1995 No                                     NA
##  5 2020-02-11 17:01:06            1970 No                                     NA
##  6 2020-02-11 17:02:00            1993 No                                     NA
##  7 2020-02-11 17:02:02            1991 No                                     NA
##  8 2020-02-11 17:03:38            1980 No                                     NA
##  9 2020-02-11 17:05:55            1996 No                                     NA
## 10 2020-02-11 17:06:43            1981 No                                     NA
## # ... with 17 more rows, and 18 more variables:
## #   Year when stopped smoking <dbl>,
## #   When did you start looking for an internship <chr>, Sex <chr>,
## #   When did you stopped looking for an internship <chr>,
## #   Have you found an internship? <chr>,
## #   Education: background (pick a main one you identify with) <chr>,
## #   Years of education <dbl>, Do you have children? <chr>, Cohort <chr>,
## #   sd <dttm>, ed <dttm>, st <dbl>, foundInt <lgl>, children <lgl>, age <dbl>,
## #   cohort <dbl>, education <dbl>, edu_years <dbl>
```

```
table(d_searchtime$edu_years)
```

```
##
##  6 14 15 16 17 18 19 20 21 22 23 25
##  2  1  1  4  1  3  2  5  4  1  1  1
```

Sex

```
d_searchtime <- d_searchtime %>%
  mutate(sex = factor(Sex, levels = c("Female", "Male")))

table(d_searchtime$sex, useNA = "always")
```

```
##
## Female   Male   <NA>
##      6     21      0
```

Building different models:

```r
m1 <- coxph(Surv(st, foundInt) ~ 1, data = d_searchtime)
m2 <- coxph(Surv(st, foundInt) ~ cohort, data = d_searchtime)
m3 <- coxph(Surv(st, foundInt) ~ age, data = d_searchtime)
```

```r
m4 <- coxph(Surv(st, foundInt) ~ children, data = d_searchtime)
m5 <- coxph(Surv(st, foundInt) ~ education, data = d_searchtime)
```

```
## Warning in fitter(X, Y, istrat, offset, init, control, weights = weights, :
## Loglik converged before variable 5 ; coefficient may be infinite.
```

```r
m6 <- coxph(Surv(st, foundInt) ~ education + edu_years, data = d_searchtime)
```

```
## Warning in fitter(X, Y, istrat, offset, init, control, weights = weights, :
## Loglik converged before variable 5 ; coefficient may be infinite.
```

(Error: author of pkg:survival said the test that is being triggered to generate that warning is overly sensitive. Generally the warning is not correct.)

Comparing models according to their AIC:

```r
fits <- list(m2 = m2, m3 = m3,m4 = m4,m5 = m5,m6 = m6)
sapply(fits, AIC)
```

```
##       m2       m3       m4       m5       m6
## 112.6694 118.3826 120.0459 125.4637 123.0054
```

Best model seems to be m2.

```r
m_full <- coxph(Surv(st, foundInt) ~ cohort + children + sex +
                 education + edu_years + age, data = d_searchtime, control = coxph.control(iter.max = 2
```

```
## Warning in fitter(X, Y, istrat, offset, init, control, weights = weights, :
## Loglik converged before variable 18 ; coefficient may be infinite.
```

```r
mAIC <- step(m_full)
```

```
## Start:  AIC=123.37
## Surv(st, foundInt) ~ cohort + children + sex + education + edu_years +
##     age
```

```
## Warning in fitter(X, Y, istrat, offset, init, control, weights = weights, :
## Loglik converged before variable 7 ; coefficient may be infinite.
```

```
## Warning in fitter(X, Y, istrat, offset, init, control, weights = weights, :
## Loglik converged before variable 17 ; coefficient may be infinite.
```

```
## Warning in fitter(X, Y, istrat, offset, init, control, weights = weights, :
## Loglik converged before variable 17 ; coefficient may be infinite.
```

```
## Warning in fitter(X, Y, istrat, offset, init, control, weights = weights, :
## Loglik converged before variable 18 ; coefficient may be infinite.

## Warning in fitter(X, Y, istrat, offset, init, control, weights = weights, :
## Loglik converged before variable 18 ; coefficient may be infinite.

##             Df    AIC
## - education  5 116.03
## - age        1 121.69
## - sex        1 121.95
## <none>         123.37
## - edu_years  1 123.69
## - children   1 125.09
## - cohort     6 126.60
##
## Step:  AIC=116.03
## Surv(st, foundInt) ~ cohort + children + sex + edu_years + age
##
##             Df    AIC
## - age        1 114.11
## - sex        1 114.35
## - edu_years  1 115.66
## <none>         116.03
## - children   1 117.59
## - cohort     6 120.22
##
## Step:  AIC=114.11
## Surv(st, foundInt) ~ cohort + children + sex + edu_years
##
##             Df    AIC
## - sex        1 112.46
## - edu_years  1 113.76
## <none>         114.11
## - children   1 115.70
## - cohort     6 118.32
##
## Step:  AIC=112.46
## Surv(st, foundInt) ~ cohort + children + edu_years
##
##             Df    AIC
## - edu_years  1 111.85
## <none>         112.46
## - children   1 113.73
## - cohort     6 116.35
##
## Step:  AIC=111.85
## Surv(st, foundInt) ~ cohort + children
##
##            Df    AIC
## <none>        111.85
## - children  1 112.67
## - cohort    6 120.05
```

```r
summary(mAIC)
```

```
## Call:
## coxph(formula = Surv(st, foundInt) ~ cohort + children, data = d_searchtime,
##     control = coxph.control(iter.max = 21))
##
##   n= 27, number of events= 23
##
##                  coef exp(coef) se(coef)     z Pr(>|z|)
## cohortA15      5.7517  314.7337   1.6594 3.466 0.000528 ***
## cohortS16          NA        NA   0.0000    NA       NA
## cohortA16          NA        NA   0.0000    NA       NA
## cohortS17          NA        NA   0.0000    NA       NA
## cohortA17      4.6482  104.3962   1.5787 2.944 0.003236 **
## cohortS18          NA        NA   0.0000    NA       NA
## cohortA18      4.2423   69.5665   1.3278 3.195 0.001399 **
## cohortS19      1.9165    6.7974   1.2539 1.529 0.126386
## cohortA19      2.5155   12.3722   1.0797 2.330 0.019823 *
## cohortS20      1.8519    6.3716   1.1187 1.655 0.097861 .
## cohortA20          NA        NA   0.0000    NA       NA
## childrenTRUE   1.1035    3.0148   0.6293 1.754 0.079497 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##              exp(coef) exp(-coef) lower .95 upper .95
## cohortA15      314.734   0.003177   12.1759   8135.53
## cohortS16           NA         NA        NA        NA
## cohortA16           NA         NA        NA        NA
## cohortS17           NA         NA        NA        NA
## cohortA17      104.396   0.009579    4.7307   2303.81
## cohortS18           NA         NA        NA        NA
## cohortA18       69.566   0.014375    5.1540    938.98
## cohortS19        6.797   0.147114    0.5822     79.37
## cohortA19       12.372   0.080826    1.4906    102.69
## cohortS20        6.372   0.156946    0.7112     57.08
## cohortA20           NA         NA        NA        NA
## childrenTRUE     3.015   0.331697    0.8782     10.35
##
## Concordance= 0.765  (se = 0.065 )
## Likelihood ratio test= 22.14  on 7 df,   p=0.002
## Wald test            = 16.64  on 7 df,   p=0.02
## Score (logrank) test = 27.84  on 7 df,   p=2e-04
```

```r
m_final <- coxph(Surv(st, foundInt) ~ cohort + children,
               data = d_searchtime)
summary(m_final)
```

```
## Call:
## coxph(formula = Surv(st, foundInt) ~ cohort + children, data = d_searchtime)
##
##   n= 27, number of events= 23
##
```

```
##                  coef exp(coef) se(coef)     z Pr(>|z|)    
## cohortA15      5.7517  314.7337   1.6594 3.466 0.000528 ***
## cohortS16          NA        NA   0.0000    NA       NA    
## cohortA16          NA        NA   0.0000    NA       NA    
## cohortS17          NA        NA   0.0000    NA       NA    
## cohortA17      4.6482  104.3962   1.5787 2.944 0.003236 ** 
## cohortS18          NA        NA   0.0000    NA       NA    
## cohortA18      4.2423   69.5665   1.3278 3.195 0.001399 ** 
## cohortS19      1.9165    6.7974   1.2539 1.529 0.126386    
## cohortA19      2.5155   12.3722   1.0797 2.330 0.019823 *  
## cohortS20      1.8519    6.3716   1.1187 1.655 0.097861 .  
## cohortA20          NA        NA   0.0000    NA       NA    
## childrenTRUE   1.1035    3.0148   0.6293 1.754 0.079497 .  
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
##              exp(coef) exp(-coef) lower .95 upper .95
## cohortA15      314.734   0.003177   12.1759   8135.53
## cohortS16           NA         NA        NA        NA
## cohortA16           NA         NA        NA        NA
## cohortS17           NA         NA        NA        NA
## cohortA17      104.396   0.009579    4.7307   2303.81
## cohortS18           NA         NA        NA        NA
## cohortA18       69.566   0.014375    5.1540    938.98
## cohortS19        6.797   0.147114    0.5822     79.37
## cohortA19       12.372   0.080826    1.4906    102.69
## cohortS20        6.372   0.156946    0.7112     57.08
## cohortA20           NA         NA        NA        NA
## childrenTRUE     3.015   0.331697    0.8782     10.35
## 
## Concordance= 0.765  (se = 0.065 )
## Likelihood ratio test= 22.14  on 7 df,   p=0.002
## Wald test            = 16.64  on 7 df,   p=0.02
## Score (logrank) test = 27.84  on 7 df,   p=2e-04
```

```
AIC(m1)
```

```
## [1] 119.9864
```

```
AIC(m2)
```

```
## [1] 112.6694
```

```
AIC(m3)
```

```
## [1] 118.3826
```

```
AIC(m4)
```

```
## [1] 120.0459
```

```r
AIC(m5)
```

```
## [1] 125.4637
```

```r
AIC(m6)
```

```
## [1] 123.0054
```

```r
AIC(mAIC)
```

```
## [1] 111.8501
```

```r
AIC(m_final)
```

```
## [1] 111.8501
```

Let's now make model based predictions

```r
i.training <- sample.int(nrow(d_searchtime), size = ceiling(nrow(d_searchtime)/2), replace = FALSE)
i.testing <- setdiff(seq_len(nrow(d_searchtime)), i.training)
d_training <- d_searchtime[i.training, ]
d_testing <- d_searchtime[i.testing, ]
```

Let's test the two models that gave the best AIC.

First we train our models:

```r
MA <- coxph(Surv(st, foundInt) ~ cohort, data = d_training, control = coxph.control(iter.max = 22))
```

```
## Warning in fitter(X, Y, istrat, offset, init, control, weights = weights, :
## Loglik converged before variable 5,7,8,9,10 ; coefficient may be infinite.
```

```r
MB <- coxph(Surv(st, foundInt) ~ cohort + children, data = d_training, control = coxph.control(iter.max
```

```
## Warning in fitter(X, Y, istrat, offset, init, control, weights = weights, :
## Loglik converged before variable 5,7,8,9,10 ; coefficient may be infinite.
```

```r
d_testing$lp_A <- predict(MA, newdata = d_testing, type = "lp")
d_testing$lp_B <- predict(MB, newdata = d_testing, type = "lp")
```

Let's now assess our models:

```r
res.coxA<- coxph(Surv(st, foundInt) ~ lp_A, data = d_testing)


test.phA <- cox.zph(res.coxA)
test.phA
```

```
##        chisq df   p
## lp_A    2.65  1 0.1
## GLOBAL  2.65  1 0.1
```

```r
res.coxB<- coxph(Surv(st, foundInt) ~ lp_B, data = d_testing)
test.phB <- cox.zph(res.coxB)
test.phB
```

```
##         chisq df    p
## lp_B     2.58  1 0.11
## GLOBAL   2.58  1 0.11
```

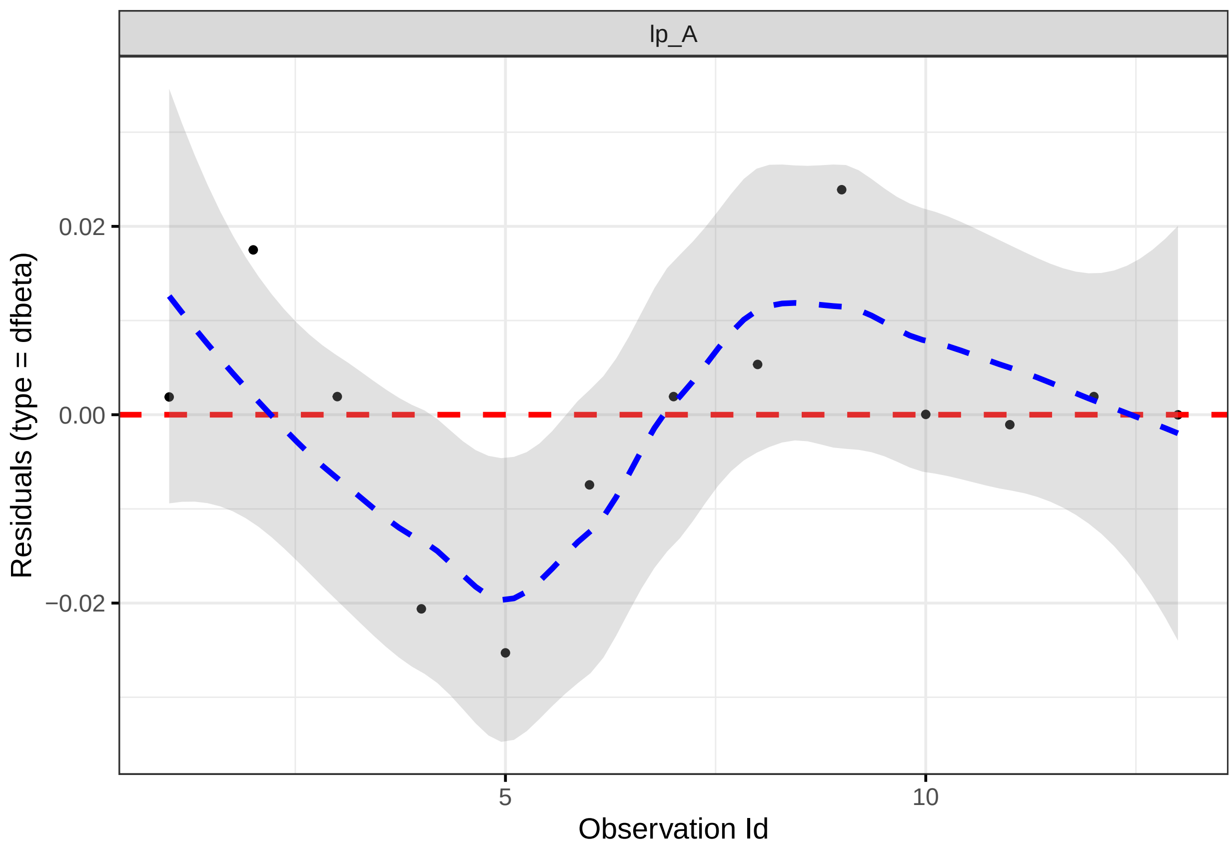
From the output above, the tests are not statistically significant for each of the covariates, and the global tests are also not statistically significant. Therefore, we can assume the proportional hazards.

To test influential observations or outliers, we can visualize either:

- the dfbeta values
- the deviance residuals

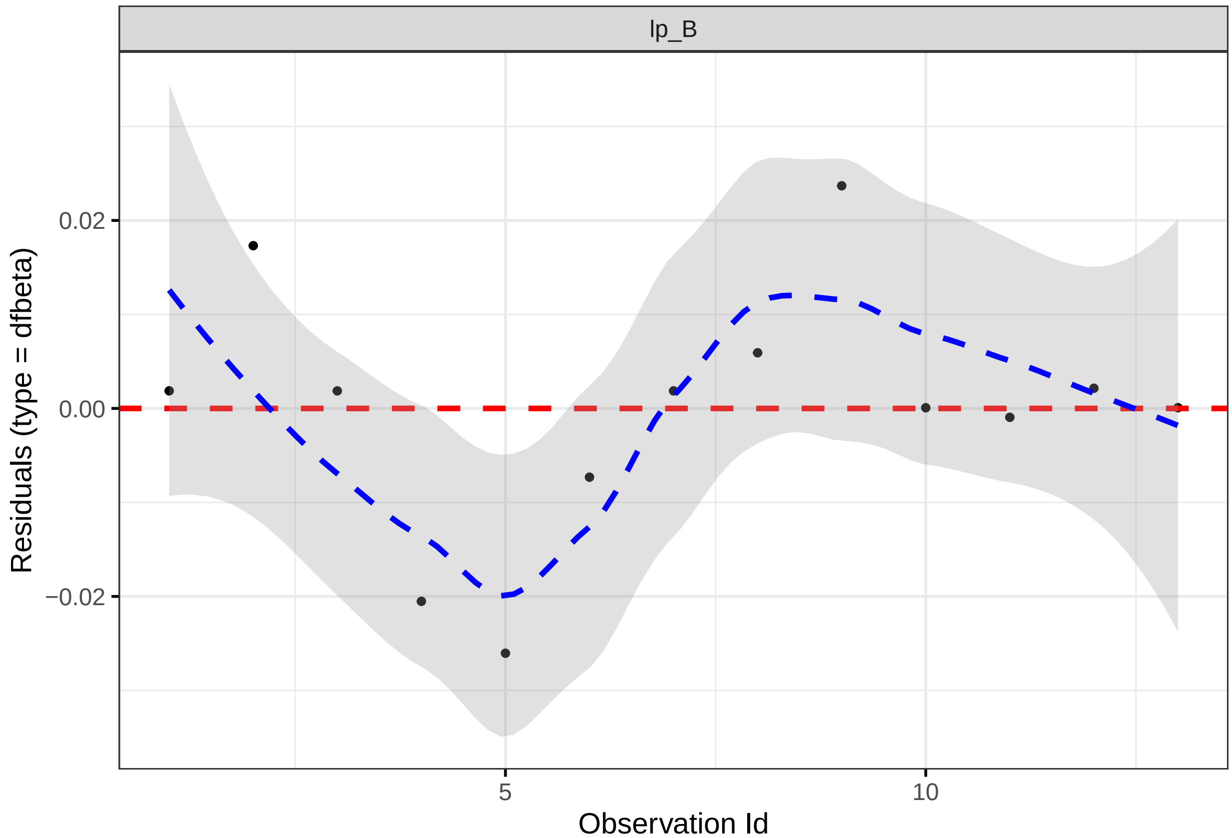
```r
ggcoxdiagnostics(res.coxA, type = "dfbeta", linear.predictions = FALSE, ggtheme = theme_bw())
```

```
## `geom_smooth()` using formula 'y ~ x'
```




```r
ggcoxdiagnostics(res.coxB, type = "dfbeta", linear.predictions = FALSE, ggtheme = theme_bw())
```

```
## `geom_smooth()` using formula 'y ~ x'
```



The above index plots show that comparing the magnitudes of the largest dfbeta values to the regression coefficients suggests that none of the observations is terribly influential individually.

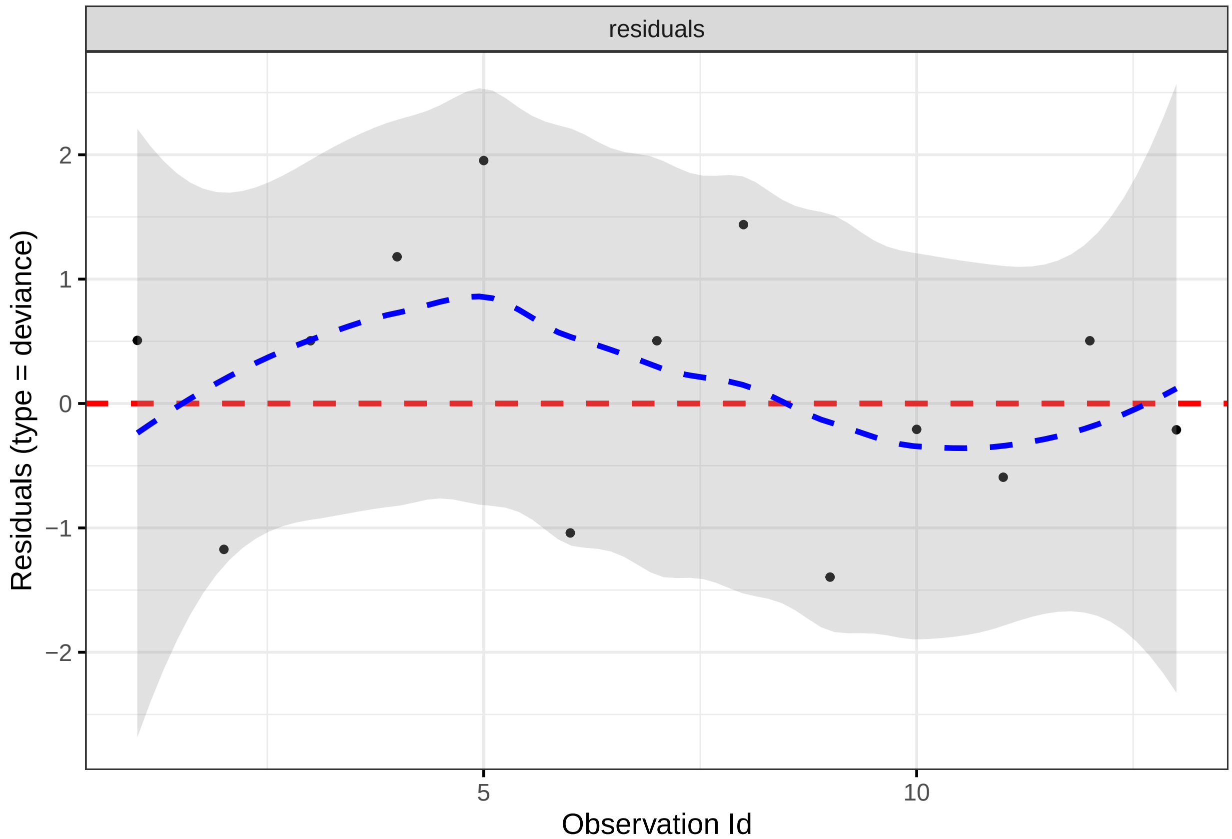
It's also possible to check outliers by visualizing the deviance residuals. The deviance residual is a normalized transform of the martingale residual. These residuals should be roughly symmetrically distributed about zero with a standard deviation of 1.

Positive values correspond to individuals that "found an internship too soon" compared to expected survival times. Negative values correspond to individual that "took to long to find an intership".

Very large or small values are outliers, which are poorly predicted by the model.

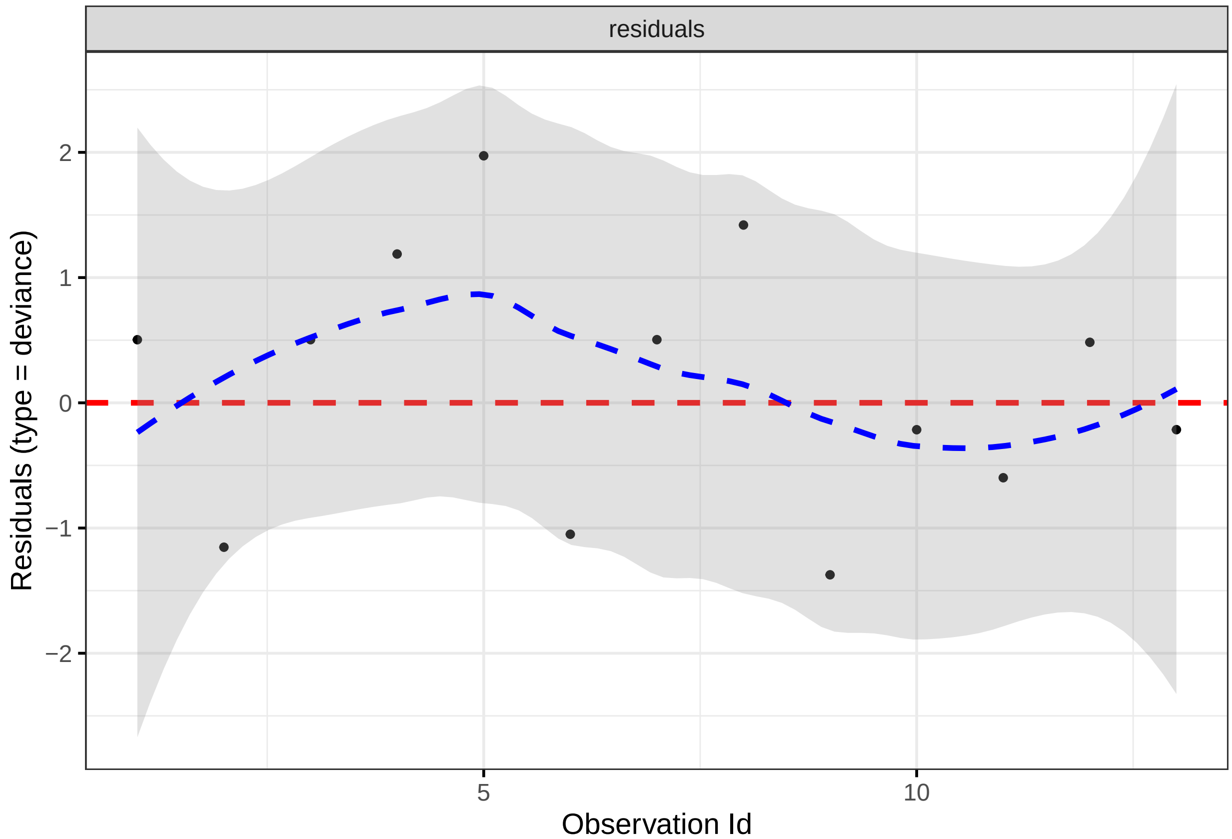
```
ggcoxdiagnostics(res.coxA, type = "deviance",
                 linear.predictions = FALSE, ggtheme = theme_bw())
```

```
## 'geom_smooth()' using formula 'y ~ x'
```




```r
ggcoxdiagnostics(res.coxB, type = "deviance",
                 linear.predictions = FALSE, ggtheme = theme_bw())
```

```
## `geom_smooth()` using formula 'y ~ x'
```

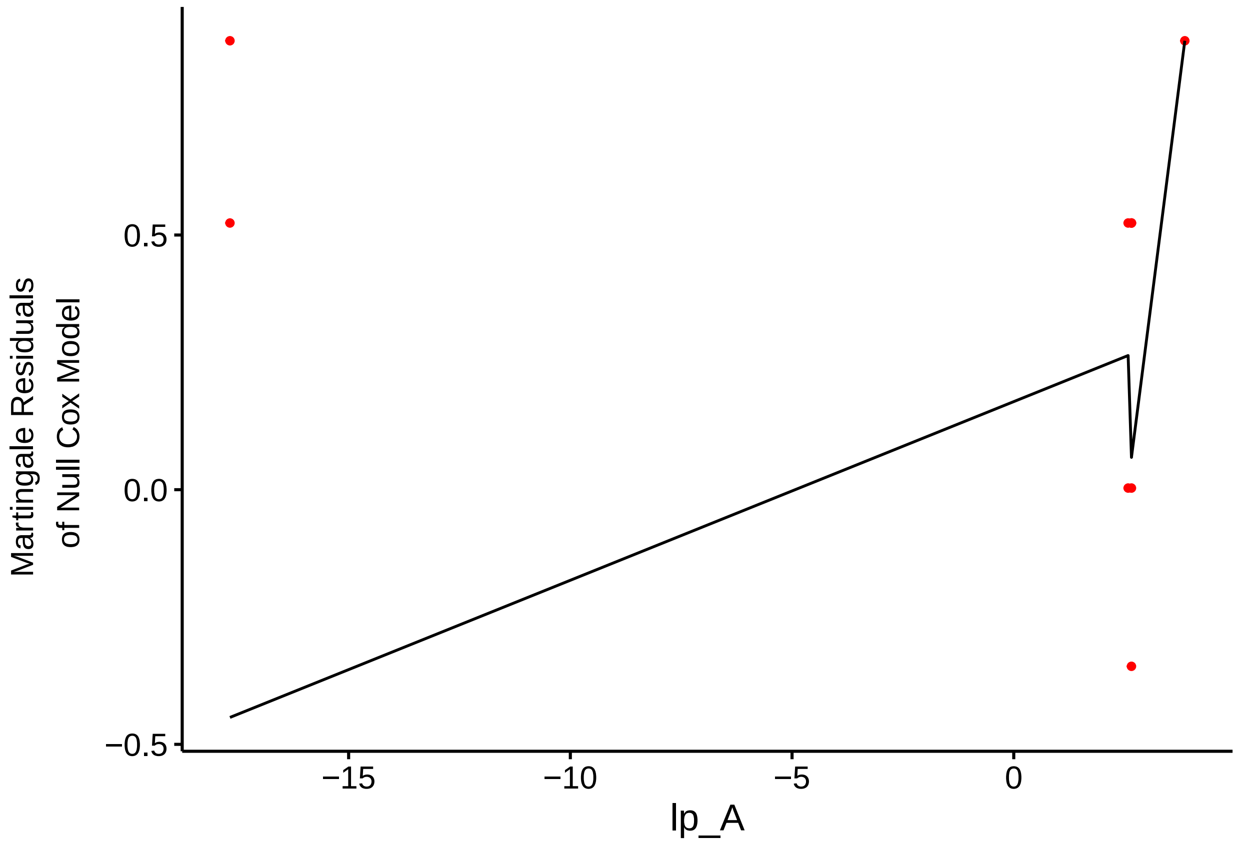


Both models look fairly symmetric around 0.

Testing non-linearity

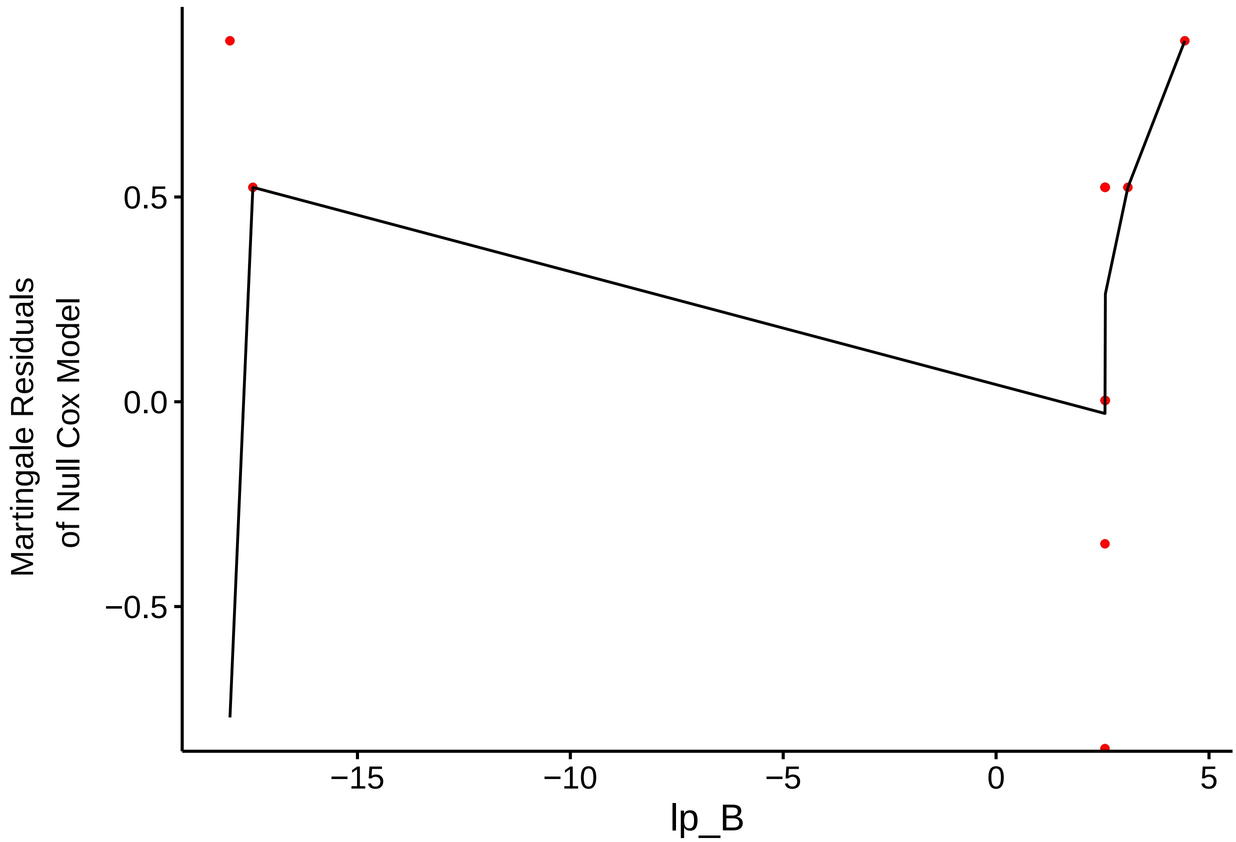
```
ggcoxfunctional(Surv(st, foundInt) ~ lp_A, data = d_testing)
```

```
## Warning: arguments formula is deprecated; will be removed in the next version;
## please use fit instead.
```




```
ggcoxfunctional(Surv(st, foundInt) ~ lp_B, data = d_testing)
```

```
## Warning: arguments formula is deprecated; will be removed in the next version;
## please use fit instead.
```



It appears that, nonlinearity is here for both models.

Both models seem to be valid regarding the Cox model assumptions.

```
save.image("myWorkSpace.RData")
```