# Survival Data Analysis for Cancer Data

*Corrado Lanera, Danila Azzolina, Daniele Bottigliengo*[1]

*02 - 06 October, 2017*

[1]Unit of Biostatistics, Epidemiology and Public Health of the Dep. of Cardiac, Thoracic and Vascular Sciences — Univ. of Padova

# Contents

# Introduction

This book is designed to collect notes and exercises from the Ph.D. course on **Survival Data Analysis for Cancer Data** by prof. Sylvie Chevret and prof. Matthieu Resche-Rigon from ECSTRRA Team, Inserm, University of Paris Diderot, promoted by the Dep. of Mathematical Sciences "G. L. Lagrange" of the Politecnico of Torino (Italy).

## Contributions

Any contribution is welcome! From the download button on the top of each (HTML) page you can download both the `epub` and the `PDF` versions of the present book.

If you find any mistake/typo or want to share ideas, you can help improve the book in the following way:

- Providing a solution proposal by opening a pull request to the related git repository (https://github.com/CorradoLanera/SuDACDa/pulls)
- Asking for a fix by opening an issue to the project (https://github.com/CorradoLanera/SuDACDa/issues)

## Settings

Here, there are the libraries loaded during the course, w/ the relative options, plus some packages and options useful to write code more understandable by humans obtaining nicer output.

```
# Packages for the analyses
library(survival)                                          # Survival Analysis
library(survminer)                      # Drawing Survival Curves using 'ggplot2'
library(cmprsk)                                              # Competing risk
library(rms)              # Regression Modeling Strategy (include Hmisc package)
  options(datadist = 'dd')                   # Distribution Summaries used by rms

# Package(s) for data management
library(tidyverse)                       # Imports the principal tidyverse packages

# Document output options
knitr::opts_chunk$set(
    echo        = TRUE,                                    # Render all the code
    message     = FALSE,                                 # Do net render messages
    warning     = FALSE,                                 # Do not render warnings
    fig.height  = 4.4,   # Right figure height to permit two figures in a PDF page
    cache.extra = knitr::rand_seed   # cache random seed to assure reproducibility
)
```

The following code create the packages.bib files which is the BibTeX lists of all the packages references we have loaded.

```
# Automatically create a bib database for the loaded packages
knitr::write_bib(c(.packages(), 'bookdown', 'knitr', 'rmarkdown'),
  file = 'packages.bib'
)
```

# Chapter 1

# *Monday*: Introduction to Survival Analyses and simulation of data

## 1.1 Key (operative) concepts

1. Time has asymmetric density and can be censored:

- not possible to summarize it by the means
- cannot be normal distributed
- use exponential family

2. Plot the log-plot to check the distribution assumptions

3. Censoring can be:

- Right: event not (yet) occurred at f-up
  - Fixed (identical f-up for anyone)
  - Sequential ($min(T_i, C_i)$)
  - Random
- Left: the event has occurred before the observed period (all population but not all information, e.g. menarche date)
- Interval: the event can be occurred between two times (but don't know when)
- Left truncated: starting point is after the beginning (different from Left, all the information but not complete population)

4. Models:

- statistical: non-informative censoring (Kaplan-Meier, Cox model, …)
- probabilistic: independent censoring (life tables)
- parametric (`survival::survreg()`, need to define the distribution) VS non-parametric (`survival::survfit()` or `rms::npsurv()`, no need to define distribution)

## 1.2 Simulated Data

1. Simulate a sample of $n = 100$ or 1000 exponential survival times, w/ mean $\theta = 5$.

- Non censored

```
set.seed(171002)
n              <- c(thousand = 1000)                                  # samples
t              <- rexp(n, rate = 5)                        # random exponential times
status_no_cens <- rep(1, times = n)          # no censored data --> all are cases
```

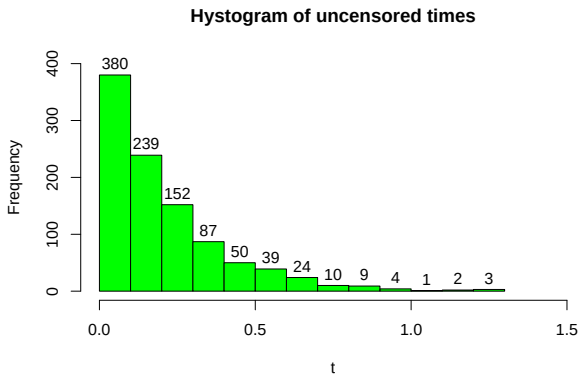
- Uniform censoring over $[0, a]$, w/ $a = 1, a = 0.5$ or $a = 2$

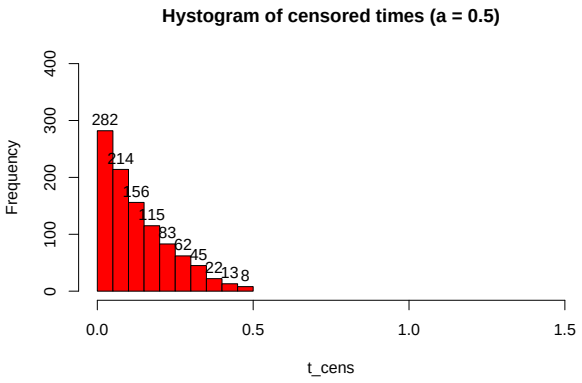
```
a              <- c(cens_05 = 0.5)    # upper bound of the uniform censoring dist
cens           <- runif(n, min = 0, max = a)                    # censored times
t_cens         <- pmin(t, cens)     # censored times are earlier than event times
status_cens    <- status_no_cens - (t_cens == cens)       # remove censored cases
```

2. Plot the observed survival times

- Non censored and censored

```
# NOTE: for the plots to be comparable, xlim and ylim have to be the same range
#       for both the plots. Moreover to drow well adjusted plots, they were set
#       a posteriori.
hist(t,
  main = 'Hystogram of uncensored times',
  col  = 'green',
  xlim = c(0, 1.5),
  ylim = c(0, 400),
  labels = TRUE                          # add the labels over the top of the bars
)
hist(t_cens,
  main = 'Hystogram of censored times (a = 0.5)',
  col  = 'red',
  xlim = c(0, 1.5),
  ylim = c(0, 400),
  labels = TRUE
)
```




3. Parametric estimation of survival function

- Uncensored

```
# `?survreg` := "Regression for a Parametric Survival Model"
#
# R formula: y ~ x <--> math formula: y = f(x)
#
# Here we want to model the response (labelled time) as they are, w/out any
```

```
# furter investigation on the effect on them from some other variable
survreg(Surv(t, status_no_cens) ~ 1,
  dist = 'exponential'
) %>%
  summary     # here `summary()` add some more statistics to the standard output
```

```
##
## Call:
## survreg(formula = Surv(t, status_no_cens) ~ 1, dist = "exponential")
##             Value Std. Error     z p
## (Intercept) -1.58     0.0316 -50.1 0
##
## Scale fixed at 1
##
## Exponential distribution
## Loglik(model)= 584.3   Loglik(intercept only)= 584.3
## Number of Newton-Raphson Iterations: 4
## n= 1000
```

- Censored

```
survreg(Surv(t_cens, status_cens) ~ 1,
  dist = 'exponential'
) %>%
  summary
```

```
##
## Call:
## survreg(formula = Surv(t_cens, status_cens) ~ 1, dist = "exponential")
##             Value Std. Error     z p
## (Intercept) -1.57     0.0401 -39.2 0
##
## Scale fixed at 1
##
## Exponential distribution
## Loglik(model)= 355.9   Loglik(intercept only)= 355.9
## Number of Newton-Raphson Iterations: 4
## n= 1000
```

4. Non parametric estimation of survival and the distribution functions

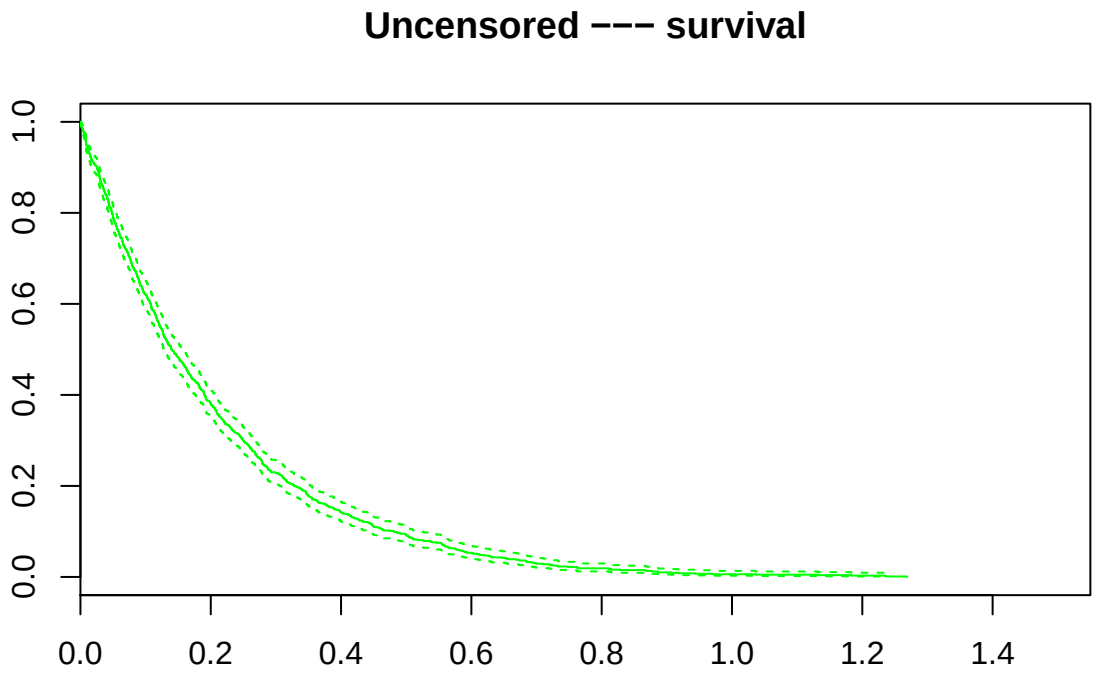
- Uncensored

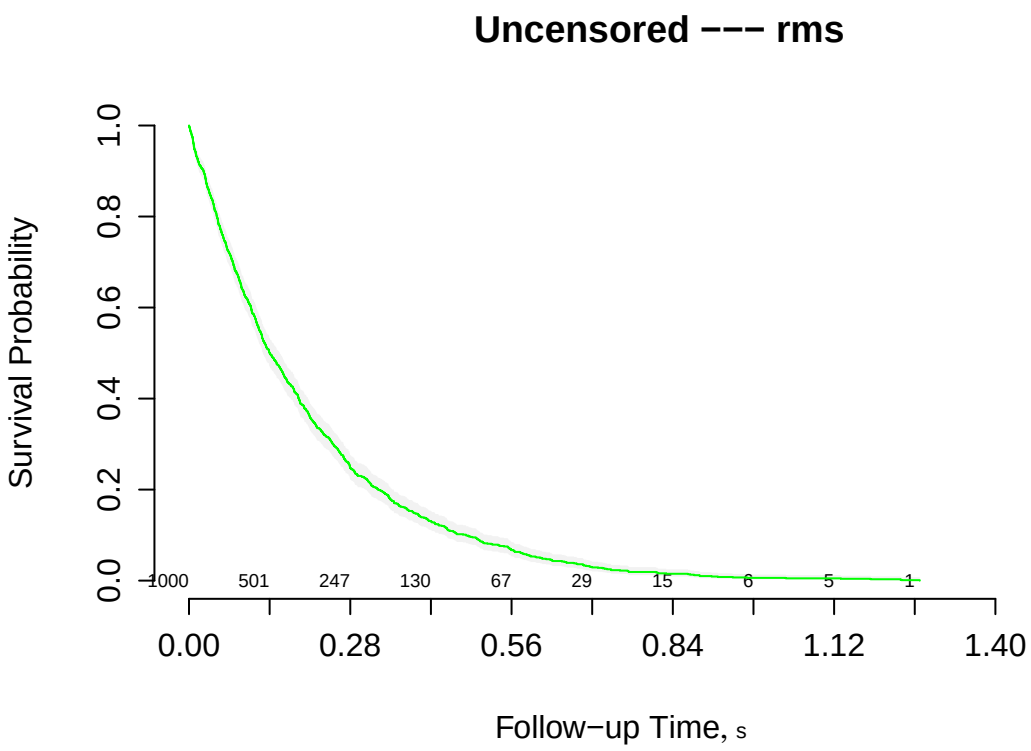
```
# `?survfit` := "Create survival curves"
survfit(Surv(t, status_no_cens) ~ 1)
```

```
## Call: survfit(formula = Surv(t, status_no_cens) ~ 1)
##
##        n   events   median  0.95LCL  0.95UCL
## 1000.000 1000.000    0.140    0.128    0.158
```

```
# Here we would like to compare to approach to survival plots:
# 1. Using the packege _survival_, so the standard one
# 2. Uisng the package _rms_, a comprehensive package for regression analyses

# Using survival `plot` provided by the _survival_ package
# (`?survival:::plot.survfit`), we can continue to
```

```r
# use the `survfit()` function for nonparametric survival estimation from the
# same _survival_ package
survfit(Surv(t, status_no_cens) ~ 1) %>%
  plot(
    xlim     = c(0, 1.55),
    conf.int  = TRUE,
    mark.time = TRUE,
    col       = 'green',
    main      = 'Uncensored --- survival'
)


# Using the survplot from the _rms_ package (`survplot`), we have to switch to
# the `npsurv()` function for nonparametric survival estimation from the _rms_
# package
npsurv(Surv(t, status_no_cens) ~ 1) %>%
  survplot(
    xlim     = c(0, 1.5),
    conf.int = TRUE,
    n.risk   = TRUE,
    col      = 'green'
)
title(main = 'Uncensored --- rms') # unfortunally survplot do not have an
                                   # integrated option for the title...
```



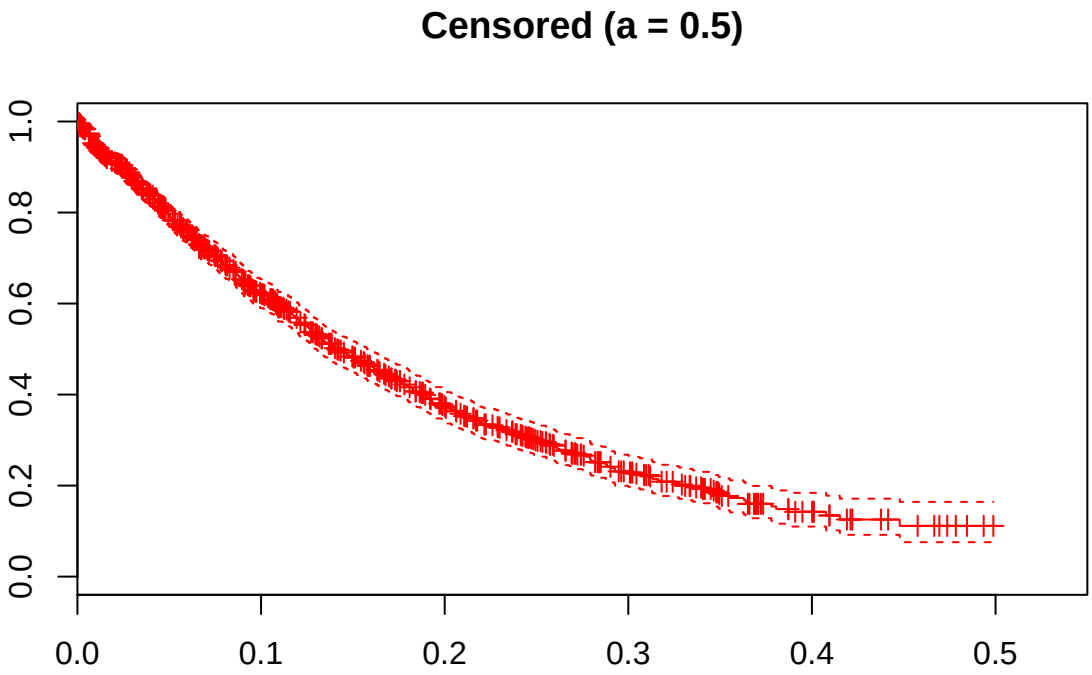


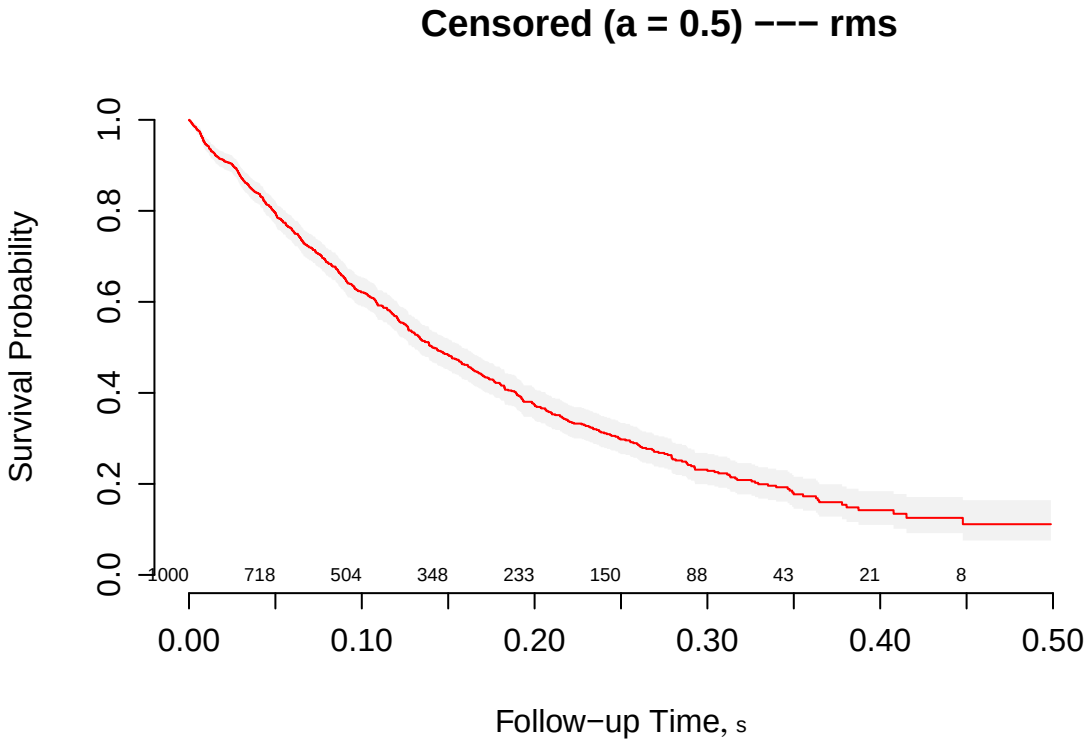

- censored

```
survfit(Surv(t_cens, status_cens) ~ 1)
```

```
## Call: survfit(formula = Surv(t_cens, status_cens) ~ 1)
##
##        n   events   median  0.95LCL  0.95UCL
## 1000.000  623.000    0.141    0.130    0.158
```

```
survfit(Surv(t_cens, status_cens) ~ 1) %>%
  plot(
    xlim     = c(0, 0.55),
    conf.int  = TRUE,
    mark.time = TRUE,
    col      = 'red',
    main      = 'Censored (a = 0.5)'
)

npsurv(Surv(t_cens, status_cens) ~ 1) %>%
  survplot(
    xlim     = c(0, 0.5),
    conf.int = TRUE,
    n.risk   = TRUE,
    col      = 'red'
)
title(main = 'Censored (a = 0.5) --- rms')
```


Censored (a = 0.5)
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.1
0.2
0.3
0.4
0.5


Censored (a = 0.5) −−− rms
Survival Probability
0.0
0.2
0.4
0.6
0.8
1.0
1000
718
504
348
233
150
88
43
21
8
0.00
0.10
0.20
0.30
0.40
0.50
Follow−up Time, s

## 1.3 mgus data from survival package

1. Load and explore data

```
data(mgus)                                                    # load
head(mgus)                                                  # first 10 rows
```

```
##   id age    sex dxyr pcdx pctime futime death alb creat  hgb mspike
## 1  1  78 female   68 <NA>     NA    748     1 2.8   1.2 11.5    2.0
## 2  2  73 female   66   LP   1310   6751     1  NA    NA   NA    1.3
## 3  3  87   male   68 <NA>     NA    277     1 2.2   1.1 11.2    1.3
## 4  4  86   male   69 <NA>     NA   1815     1 2.8   1.3 15.3    1.8
## 5  5  74 female   68 <NA>     NA   2587     1 3.0   0.8  9.8    1.4
## 6  6  81   male   68 <NA>     NA    563     1 2.9   0.9 11.5    1.8
```

```
dim(mgus)                                                # number of rows and cols
```

```
## [1] 241  12
```

```
names(mgus)                                                 # name of the columns
```

```
##  [1] "id"     "age"    "sex"    "dxyr"   "pcdx"   "pctime" "futime"
##  [8] "death"  "alb"    "creat"  "hgb"    "mspike"
```

```
str(mgus)                                       # R internal structure of the object
```

```
## 'data.frame':    241 obs. of  12 variables:
##  $ id    : num  1 2 3 4 5 6 7 8 9 10 ...
##  $ age   : atomic  78 73 87 86 74 81 72 79 85 58 ...
##   ..- attr(*, "label")= chr "AGE AT date_on"
##  $ sex   : Factor w/ 2 levels "female","male": 1 1 2 2 1 2 1 1 1 2 ...
##   ..- attr(*, "label")= chr "Sex"
##  $ dxyr  : num  68 66 68 69 68 68 68 69 70 65 ...
##  $ pcdx  : Factor w/ 4 levels "AM","LP","MA",..: NA 2 NA NA NA NA NA NA NA NA ...
##  $ pctime: atomic  NA 1310 NA NA NA NA NA NA NA NA ...
##   ..- attr(*, "label")= chr "Progression to Group 4 (days)"
##  $ futime: atomic  748 6751 277 1815 2587 ...
##   ..- attr(*, "label")= chr "Follow-Up Time"
##  $ death : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ alb   : atomic  2.8 NA 2.2 2.8 3 2.9 3 3.1 3.2 3.5 ...
##   ..- attr(*, "label")= chr "Serum Albumin"
##  $ creat : atomic  1.2 NA 1.1 1.3 0.8 0.9 0.8 0.8 1 1 ...
##   ..- attr(*, "label")= chr "Serum Creatinine"
##  $ hgb   : atomic  11.5 NA 11.2 15.3 9.8 11.5 13.5 15.5 12.4 14.8 ...
##   ..- attr(*, "label")= chr "Hemoglobin"
##  $ mspike: atomic  2 1.3 1.3 1.8 1.4 1.8 1.3 1.4 1.5 2.2 ...
##   ..- attr(*, "label")= chr "Serum M-Spike"
##  - attr(*, "formats")=List of 1
##   ..$ death:List of 2
##   .. ..$ values: num  0 1
##   .. ..$ labels: chr  "Alive" "Dead"
```

```
summary(mgus)                                              # summary from base R
```

```
##        id              age            sex           dxyr        pcdx
##  Min.   :  1   Min.   :34.00   female:104   Min.   :56.0   AM  :  8
##  1st Qu.: 61   1st Qu.:55.00   male  :137   1st Qu.:66.0   LP  :  5
```

```
##  Median :121   Median :63.00                Median :68.0   MA  :  7
##  Mean   :121   Mean   :62.87                Mean   :67.4   MM  : 44
##  3rd Qu.:181   3rd Qu.:72.00                3rd Qu.:70.0   NA's:177
##  Max.   :241   Max.   :90.00                Max.   :73.0
##
##      pctime          futime          death             alb
##  Min.   :  365   Min.   :    6   Min.   :0.0000   Min.   :1.800
##  1st Qu.: 2469   1st Qu.: 2422   1st Qu.:1.0000   1st Qu.:2.900
##  Median : 3778   Median : 5022   Median :1.0000   Median :3.200
##  Mean   : 4342   Mean   : 5425   Mean   :0.9336   Mean   :3.204
##  3rd Qu.: 5750   3rd Qu.: 8264   3rd Qu.:1.0000   3rd Qu.:3.500
##  Max.   :11685   Max.   :14325   Max.   :1.0000   Max.   :5.100
##  NA's   :177                                      NA's   :31
##      creat            hgb            mspike
##  Min.   :0.600   Min.   : 7.40   Min.   :0.300
##  1st Qu.:0.900   1st Qu.:12.20   1st Qu.:1.500
##  Median :1.000   Median :13.20   Median :1.700
##  Mean   :1.095   Mean   :13.15   Mean   :1.764
##  3rd Qu.:1.100   3rd Qu.:14.50   3rd Qu.:2.000
##  Max.   :6.400   Max.   :16.60   Max.   :3.200
##  NA's   :43      NA's   :1
```

```
describe(mgus)    # more comprehensive description from _Hisc_ package, loaded by
```

```
## mgus
##
##  12  Variables      241  Observations
## --------------------------------------------------------------------------
## id
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##      241        0      241        1      121    80.67       13       25
##      .25      .50      .75      .90      .95
##       61      121      181      217      229
##
## lowest :   1   2   3   4   5, highest: 237 238 239 240 241
## --------------------------------------------------------------------------
## age : AGE AT date_on
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##      241        0       53    0.999    62.87    13.42       44       48
##      .25      .50      .75      .90      .95
##       55       63       72       78       81
##
## lowest : 34 35 36 37 38, highest: 84 85 86 87 90
## --------------------------------------------------------------------------
## sex : Sex
##        n  missing distinct
##      241        0        2
##
## Value      female   male
## Frequency     104    137
## Proportion  0.432  0.568
## --------------------------------------------------------------------------
## dxyr
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##      241        0       17     0.97     67.4    3.073       61       63
```

```
##      .25      .50      .75      .90      .95
##       66       68       70       70       70
##
## Value         56    58    59    60    61    62    63    64    65    66
## Frequency      1     1     5     5     2     7     7    10    10    18
## Proportion 0.004 0.004 0.021 0.021 0.008 0.029 0.029 0.041 0.041 0.075
##
## Value         67    68    69    70    71    72    73
## Frequency     24    40    45    62     2     1     1
## Proportion 0.100 0.166 0.187 0.257 0.008 0.004 0.004
## ---------------------------------------------------------------------------
## pcdx
##        n  missing distinct
##       64      177        4
##
## Value         AM    LP    MA    MM
## Frequency      8     5     7    44
## Proportion 0.125 0.078 0.109 0.688
## ---------------------------------------------------------------------------
## pctime : Progression to Group 4 (days)
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##       64      177       63        1     4342     3030     1223     1409
##      .25      .50      .75      .90      .95
##     2469     3778     5750     8946    10051
##
## lowest :   365   700   954  1218  1249, highest:  9723 10109 10359 11354 11685
## ---------------------------------------------------------------------------
## futime : Follow-Up Time
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##      241        0      237        1     5425     4222      283      779
##      .25      .50      .75      .90      .95
##     2422     5022     8264    11425    12140
##
## lowest :     6     7    31    32    39, highest: 12931 13019 13152 14111 14325
## ---------------------------------------------------------------------------
## death
##        n  missing distinct     Info      Sum     Mean      Gmd
##      241        0        2    0.186      225   0.9336   0.1245
##
## ---------------------------------------------------------------------------
## alb : Serum Albumin
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##      210       31       26    0.995    3.204   0.5293      2.3      2.6
##      .25      .50      .75      .90      .95
##      2.9      3.2      3.5      3.8      3.9
##
## lowest : 1.8 1.9 2.1 2.2 2.3, highest: 4.0 4.1 4.3 4.5 5.1
## ---------------------------------------------------------------------------
## creat : Serum Creatinine
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##      198       43       19    0.978    1.095     0.39    0.700    0.800
##      .25      .50      .75      .90      .95
##    0.900    1.000    1.100    1.300    1.615
##
```

```
## Value        0.6   0.7   0.8   0.9   1.0   1.1   1.2   1.3   1.4   1.5
## Frequency      4    13    26    42    35    29    18    12     4     4
## Proportion 0.020 0.066 0.131 0.212 0.177 0.146 0.091 0.061 0.020 0.020
##
## Value        1.6   1.7   2.0   2.5   2.6   3.5   3.6   3.7   6.4
## Frequency      1     3     1     1     1     1     1     1     1
## Proportion 0.005 0.015 0.005 0.005 0.005 0.005 0.005 0.005 0.005
## ---------------------------------------------------------------------------
## hgb : Hemoglobin
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##      240        1       66    0.999    13.15    1.865    10.20    11.09
##      .25      .50      .75      .90      .95
##    12.20    13.20    14.50    15.11    15.51
##
## lowest :  7.4  7.7  8.4  9.5  9.6, highest: 15.9 16.1 16.2 16.5 16.6
## ---------------------------------------------------------------------------
## mspike : Serum M-Spike
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##      241        0       23    0.993    1.764   0.4687      1.1      1.3
##      .25      .50      .75      .90      .95
##      1.5      1.7      2.0      2.3      2.5
##
## lowest : 0.3 0.8 0.9 1.0 1.1, highest: 2.5 2.6 2.7 2.9 3.2
## ---------------------------------------------------------------------------
```

```
                # the _rms_ one

mgus_df <- as_tibble(mgus)          # tidy data frame (important info printed all
                                    # together, and visualization auto-adjusted
                                    # to the consol width)
mgus_df
```

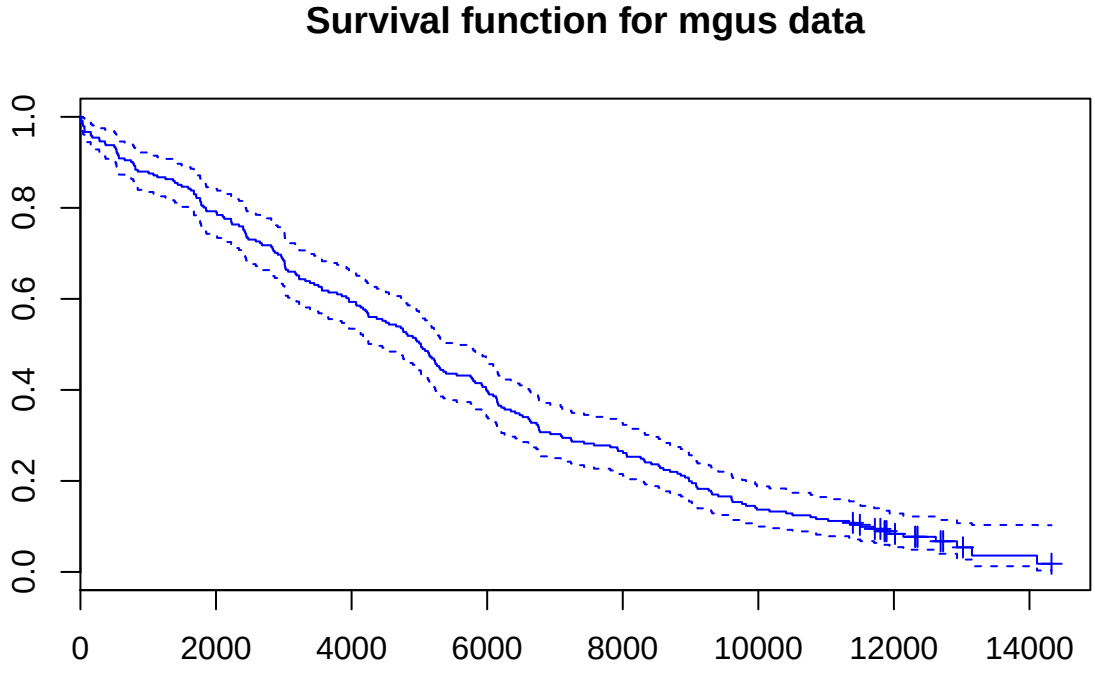
```
## # A tibble: 241 x 12
##       id   age    sex  dxyr   pcdx pctime futime death   alb creat   hgb
##  * <dbl> <dbl> <fctr> <dbl> <fctr>  <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl>
##  1     1    78 female    68   <NA>     NA    748     1   2.8   1.2  11.5
##  2     2    73 female    66     LP   1310   6751     1    NA    NA    NA
##  3     3    87   male    68   <NA>     NA    277     1   2.2   1.1  11.2
##  4     4    86   male    69   <NA>     NA   1815     1   2.8   1.3  15.3
##  5     5    74 female    68   <NA>     NA   2587     1   3.0   0.8   9.8
##  6     6    81   male    68   <NA>     NA    563     1   2.9   0.9  11.5
##  7     7    72 female    68   <NA>     NA   1135     1   3.0   0.8  13.5
##  8     8    79 female    69   <NA>     NA   2016     1   3.1   0.8  15.5
##  9     9    85 female    70   <NA>     NA   2422     1   3.2   1.0  12.4
## 10    10    58   male    65   <NA>     NA   6155     1   3.5   1.0  14.8
## # ... with 231 more rows, and 1 more variables: mspike <dbl>
```

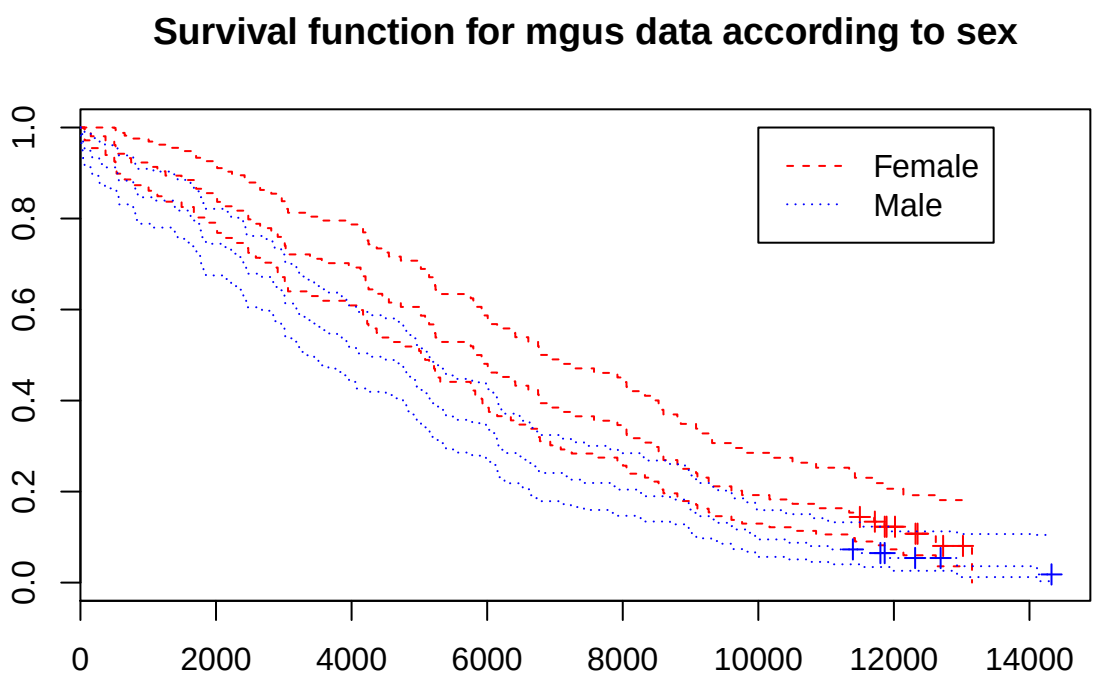
2. Non parametric Kaplan-Meyer estimation of the survival function

- Estimate the survival function from randomization overall and according to sex.

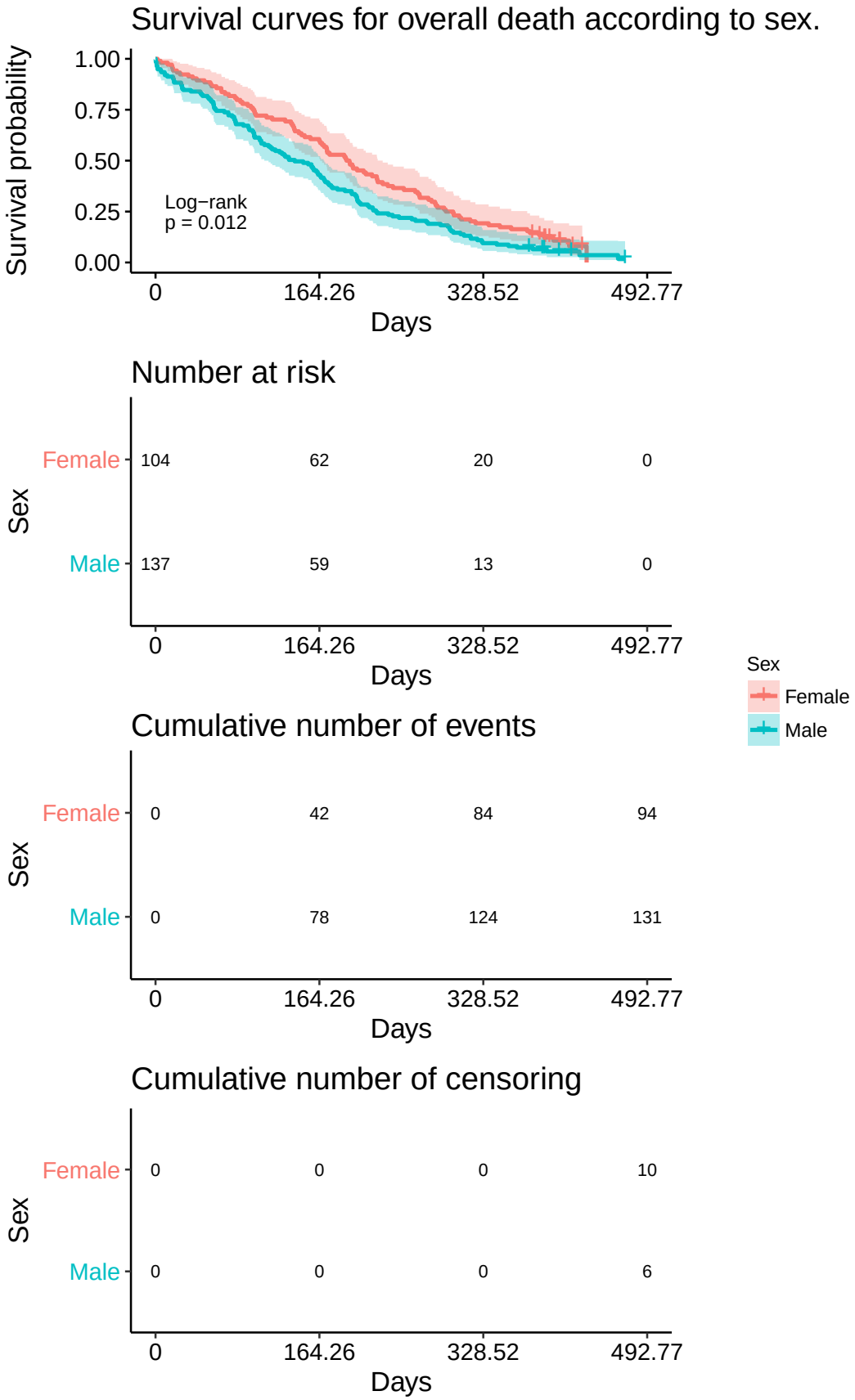
```
survfit(Surv(futime, death) ~ 1,
  data = mgus_df
) %>%
  plot(
    conf.int  = TRUE,
```

```
    mark.time = TRUE,
    col       = 'blue',
    main      = 'Survival function for mgus data'
)


survfit(Surv(futime, death) ~ sex,
  data = mgus_df
) %>%
  plot(
    conf.int  = TRUE,
    mark.time = TRUE,
    main      = 'Survival function for mgus data according to sex',
    col       = c('red', 'blue'),
    lty       = c(2, 3)
)
legend(
  x = 10000, y = 1,
  legend = c("Female", "Male"),
  col    = c('red', 'blue'),
  lty    = c(2, 3)
)
```






```
# For survival object the package _survminer_ provide ggplot2 plots
# (`?ggsurvplot`) which could be very interesting and quite comprehensive.

survfit(Surv(futime, death) ~ sex,
  data = mgus_df
) %>%
  ggsurvplot(
    conf.int            = TRUE,                      # draw confidence intervals
    pval                = TRUE,                                  # show pvalue
    pval.method         = TRUE,                          # print the test name
    title               = 'Survival curves for overall death according to sex.',
    xlab                = 'Days',
    legend              = 'right',                               # legend position
    legend.title        = 'Sex',
    legend.labs         = c('Female', 'Male'),
    risk.table          = TRUE,     # admits interesting options other than TRUE
    cumcensor           = TRUE,
    cumevents           = TRUE,
    pval.size           = 3.5,   # from here these are options passed to `ggpar`
    risk.table.fontsize = 3,     # for a better visualization
    fontsize            = 3,     # (auto-explicatives)
    xscale              = 30.44
  )
```


Survival curves for overall death according to sex.
Survival probability
1.00
0.75
0.50
0.25
0.00
Log−rank
p = 0.012
0
164.26
328.52
492.77
Days
Number at risk
Sex
Female
Male
104
62
20
0
137
59
13
0
Cumulative number of events
0
42
84
94
0
78
124
131
Cumulative number of censoring
0
0
0
10
0
0
0
6
Sex
Female
Male

Note: No female reaches the end of the f-up!

- Test the effect of sex

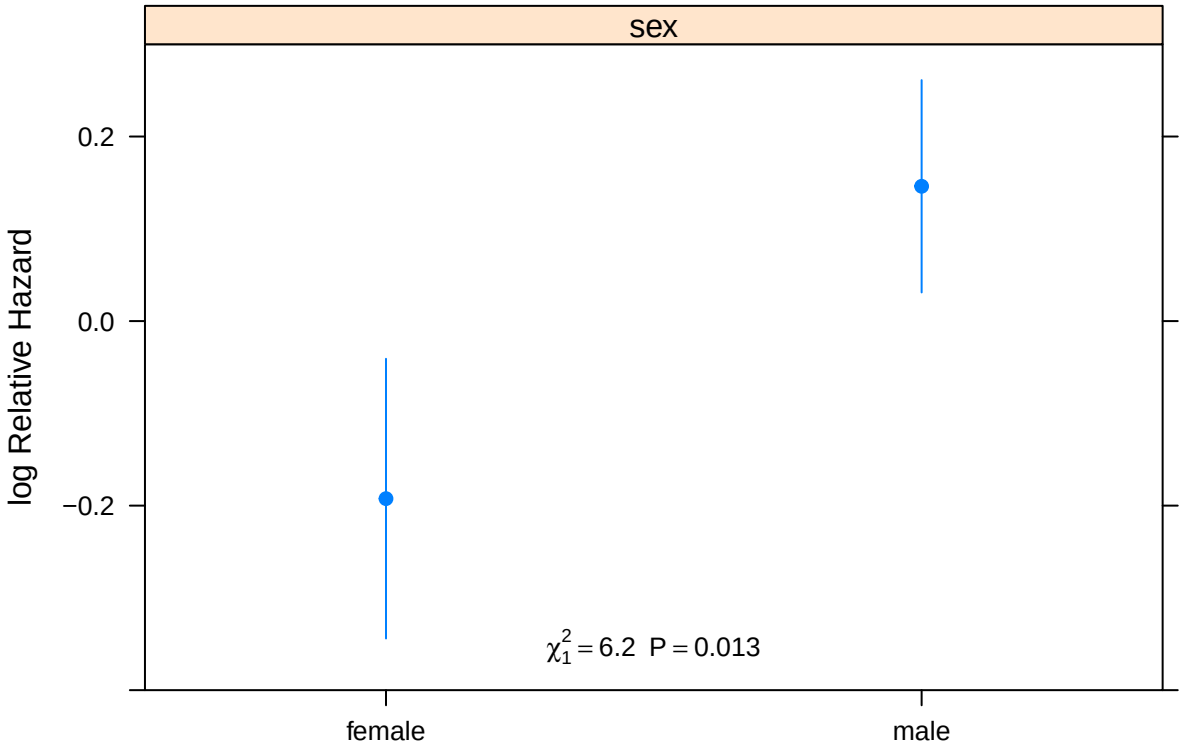
```
# Using __survival__ (no plot method is provided for this solution)
survdiff(Surv(futime, death) ~ sex,
  data = mgus_df
)
```

```
## Call:
## survdiff(formula = Surv(futime, death) ~ sex, data = mgus_df)
##
##              N Observed Expected (O-E)^2/E (O-E)^2/V
## sex=female 104       94      113      3.08      6.25
## sex=male   137      131      112      3.08      6.25
##
##  Chisq= 6.2  on 1 degrees of freedom, p= 0.0124
```

```
# using __rms__
dd <- datadist(mgus_df)  # To evaluate cph, _rms_ needs this object which simply
                         # store statistics about the data.
                         #
                         # Note: the name of the object (i.e. "dd") has to be
                         #       exactly the same as the one specified into the
                         #       option set just after the `library(rms)` call.
                         #       (See: Chapter settings)
cox_model <- cph(Surv(futime, death) ~ sex,
  data  = mgus_df
)

summary(cox_model)                              # return effect size and HR w/ CI
```

```
##              Effects              Response : Surv(futime, death)
##
##  Factor            Low High Diff. Effect   S.E.    Lower 0.95 Upper 0.95
##  sex - female:male 2   1    NA    -0.33853 0.13603 -0.60514   -0.071916
##   Hazard Ratio     2   1    NA     0.71282      NA  0.54600    0.930610
```

```
Predict(cox_model) %>%          # Compute predicted values and confidence limits
                                #
                                # Note: pay attention to Title-case "P"redict
  plot(
    groups = 'sex',
    anova  = anova(cox_model),       # Compute and print the $\chi^2$ statistics
    pval   = TRUE                    # print the pvalue
)
```




## 1.4 Non parametric Kaplan-Meier estimation of the survival function

1. Let consider a sample of $n = 500$

```
n <- 500
```

2. Simulate the dates of entry in the cohort, from January, 2010 to January, 2017

```
n_days <- 365.25 * 7              # Seven years, taking into account bissextiles
time_start <- runif(n = n,
  min = 0,
  max = n_days
) %>%
  as.Date(origin = '2010-01-01')
```

3. Simulate the data-set of death, assuming exponential death times of mean 2 years

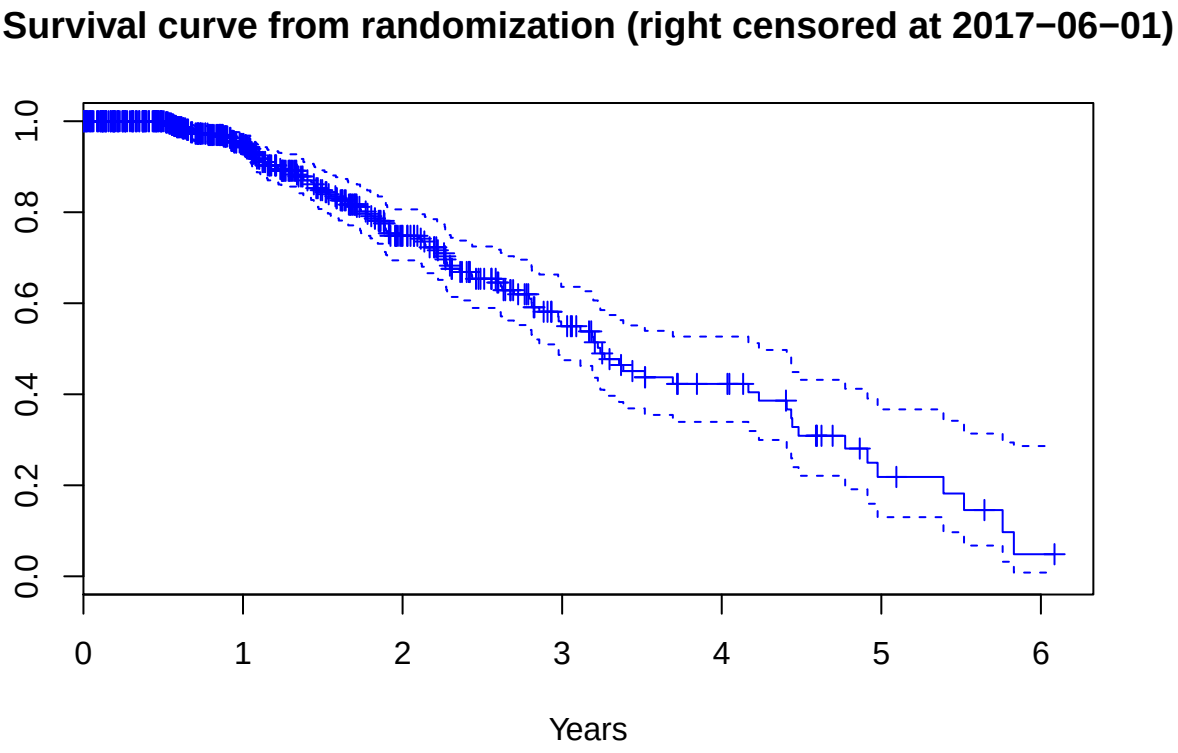
```
mean_death_time <- 365.25 * 2
death_t         <- rexp(n, rate = 1 / mean_death_time)
status_no_cens  <- rep(1, n)
```

4. Let fix the reference date of the analyses of June, 2017

```
end_date     <- as.Date('2017-06-01')           # Fixed date for the end of f-up
death_r_cens <- pmin(death_t, end_date - time_start)
status_cens  <- status_no_cens - (death_t == death_r_cens)
```

5. Estimate the survival function from randomization

```
survfit(Surv(death_r_cens, status_cens) ~ 1) %>%
  plot(
    conf.int  = TRUE,
    mark.time = TRUE,
    main      = 'Survival curve from randomization (right censored at 2017-06-01)',
    col       = 'blue',
    xlab      = 'Years',
    xscale    = 365.25
)
```



# Chapter 2

# *Tuesday*: Cox models

## 2.1 Key (operative) concepts

1. Non-informative censoring assumption!

   We cannot test for it, but we can be convinced of it

2. Test any covariates for proportional hazard. If fail:

- If $H_0$ is valid, it is not a problem
- Is it due to outliers?
- Does this variable really need?
- … do you really think that proportional of hazard should hold? What about shift to a different model?

3. Test continuous variable for log-linearity. If fail:

- try a transformation of the variable (i.e., log, spline, …)
- if it is not possible (e.g. *U-shape*) perform a categorization

  When performing categorization do not base it on the p-value: you have to explain why this choiche is clinically relevant and not statistically significant!

4. The biggest problem w/ databases w/ more observations for each patients is not the model but to produce a table w/ the right information in the right position. In particular we need the following columns

- id
- start
- end
- event
- covariates…

5. Get results easy to explain to / understand by a clinician!

## 2.2 Basic tests and funtions

For this part we will use the data `pbc` (`?pbc`) from the package **survival**.

> Note: `data(pbc)` load the `pbc` data-set and the `pbcseq` one, so on one side we do not need to call `data(pbcseq)` to load the letter, on the other side `data(pbcseq)` will throw an error because to load it we have to call `data(pbc)`. (We will use both data-sets.)

```
set.seed(171003)
data(pbc)                                                    # load the data-set
# ?pbc
pbc_df <- as_tibble(pbc)                              # create the tibble version of it
dd <- datadist(pbc_df)     # store in the dd variable its `datadist()` for _rms_

pbc_df                                                       # give a look at it
```

```
## # A tibble: 418 x 20
##       id  time status   trt      age    sex ascites hepato spiders edema
##    <int> <int>  <int> <int>    <dbl> <fctr>   <int>  <int>   <int> <dbl>
##  1     1   400      2     1 58.76523      f       1      1       1   1.0
##  2     2  4500      0     1 56.44627      f       0      1       1   0.0
##  3     3  1012      2     1 70.07255      m       0      0       0   0.5
##  4     4  1925      2     1 54.74059      f       0      1       1   0.5
##  5     5  1504      1     2 38.10541      f       0      1       1   0.0
##  6     6  2503      2     2 66.25873      f       0      1       0   0.0
##  7     7  1832      0     2 55.53457      f       0      1       0   0.0
##  8     8  2466      2     2 53.05681      f       0      0       0   0.0
##  9     9  2400      2     1 42.50787      f       0      0       1   0.0
## 10    10    51      2     2 70.55989      f       1      0       1   1.0
## # ... with 408 more rows, and 10 more variables: bili <dbl>, chol <int>,
## #   albumin <dbl>, copper <int>, alk.phos <dbl>, ast <dbl>, trig <int>,
## #   platelet <int>, protime <dbl>, stage <int>
```

```
describe(pbc_df)                                      # and whatch at some statistics
```

```
## pbc_df
##
##  20  Variables      418  Observations
## --------------------------------------------------------------------------------
## id
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##      418        0      418        1    209.5    139.7    21.85    42.70
##      .25      .50      .75      .90      .95
##   105.25   209.50   313.75   376.30   397.15
##
## lowest :   1   2   3   4   5, highest: 414 415 416 417 418
## --------------------------------------------------------------------------------
## time
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##      418        0      399        1     1918     1253    245.1    606.8
##      .25      .50      .75      .90      .95
##   1092.8   1730.0   2613.5   3524.2   4040.6
##
## lowest :   41   43   51   71   77, highest: 4500 4509 4523 4556 4795
## --------------------------------------------------------------------------------
## status
##        n  missing distinct     Info     Mean      Gmd
##      418        0        3    0.772   0.8301   0.9699
##
## Value          0     1     2
## Frequency    232    25   161
## Proportion 0.555 0.060 0.385
```

```
## ---------------------------------------------------------------------------
## trt
##        n  missing distinct     Info     Mean      Gmd
##      312      106        2     0.75    1.494   0.5015
##
## Value          1     2
## Frequency    158   154
## Proportion 0.506 0.494
## ---------------------------------------------------------------------------
## age
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##      418        0      344        1    50.74    11.96    33.84    36.37
##      .25      .50      .75      .90      .95
##    42.83    51.00    58.24    64.30    67.92
##
## lowest : 26.27789 28.88433 29.55510 30.27515 30.57358
## highest: 74.52430 75.00068 75.01164 76.70910 78.43943
## ---------------------------------------------------------------------------
## sex
##        n  missing distinct
##      418        0        2
##
## Value          m     f
## Frequency     44   374
## Proportion 0.105 0.895
## ---------------------------------------------------------------------------
## ascites
##        n  missing distinct     Info      Sum     Mean      Gmd
##      312      106        2    0.213       24  0.07692   0.1425
##
## ---------------------------------------------------------------------------
## hepato
##        n  missing distinct     Info      Sum     Mean      Gmd
##      312      106        2     0.75      160   0.5128   0.5013
##
## ---------------------------------------------------------------------------
## spiders
##        n  missing distinct     Info      Sum     Mean      Gmd
##      312      106        2    0.616       90   0.2885   0.4118
##
## ---------------------------------------------------------------------------
## edema
##        n  missing distinct     Info     Mean      Gmd
##      418        0        3    0.391   0.1005   0.1756
##
## Value        0.0   0.5   1.0
## Frequency    354    44    20
## Proportion 0.847 0.105 0.048
## ---------------------------------------------------------------------------
## bili
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##      418        0       98    0.998    3.221    3.742     0.50     0.60
##      .25      .50      .75      .90      .95
##     0.80     1.40     3.40     8.03    14.00
```

```
##
## lowest :  0.3  0.4  0.5  0.6  0.7, highest: 21.6 22.5 24.5 25.5 28.0
## ---------------------------------------------------------------------------
## chol
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##      284      134      201        1    369.5    194.5    188.4    213.6
##      .25      .50      .75      .90      .95
##    249.5    309.5    400.0    560.8    674.0
##
## lowest :  120  127  132  149  151, highest: 1336 1480 1600 1712 1775
## ---------------------------------------------------------------------------
## albumin
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##      418        0      154        1    3.497    0.473    2.750    2.967
##      .25      .50      .75      .90      .95
##    3.243    3.530    3.770    4.010    4.141
##
## lowest : 1.96 2.10 2.23 2.27 2.31, highest: 4.30 4.38 4.40 4.52 4.64
## ---------------------------------------------------------------------------
## copper
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##      310      108      158        1    97.65    83.16    17.45    24.00
##      .25      .50      .75      .90      .95
##    41.25    73.00   123.00   208.10   249.20
##
## lowest :   4   9  10  11  12, highest: 412 444 464 558 588
## ---------------------------------------------------------------------------
## alk.phos
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##      312      106      295        1     1983     1760    599.6    663.0
##      .25      .50      .75      .90      .95
##    871.5   1259.0   1980.0   3826.4   6669.9
##
## lowest :   289.0   310.0   369.0   377.0   414.0
## highest: 11046.6 11320.2 11552.0 12258.8 13862.4
## ---------------------------------------------------------------------------
## ast
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##      312      106      179        1    122.6    60.45    54.25    60.45
##      .25      .50      .75      .90      .95
##    80.60   114.70   151.90   196.47   219.25
##
## lowest :  26.35  28.38  41.85  43.40  45.00, highest: 288.00 299.15 328.60 338.00 457.25
## ---------------------------------------------------------------------------
## trig
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##      282      136      146        1    124.7    64.07    56.00    63.10
##      .25      .50      .75      .90      .95
##    84.25   108.00   151.00   195.00   230.95
##
## lowest :  33  44  46  49  50, highest: 319 322 382 432 598
## ---------------------------------------------------------------------------
## platelet
##        n  missing distinct     Info     Mean      Gmd      .05      .10
```

```
##      407       11      243        1      257    109.7    114.9    138.2
##      .25      .50      .75      .90      .95
##    188.5    251.0    318.0    386.2    430.0
##
## lowest :  62  70  71  76  79, highest: 517 518 539 563 721
## ---------------------------------------------------------------------------
## protime
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##      416        2       48    0.998    10.73    1.029     9.60     9.80
##      .25      .50      .75      .90      .95
##    10.00    10.60    11.10    12.00    12.45
##
## lowest :  9.0  9.1  9.2  9.3  9.4, highest: 13.8 14.1 15.2 17.1 18.0
## ---------------------------------------------------------------------------
## stage
##        n  missing distinct     Info     Mean      Gmd
##      412        6        4    0.893    3.024   0.9519
##
## Value          1     2     3     4
## Frequency     21    92   155   144
## Proportion 0.051 0.223 0.376 0.350
## ---------------------------------------------------------------------------
```

### 2.2.1 Impact of sex on death {sex2}

First of all we have to ask to our self, and to the clinicians, some questions:

1. There are non-informative censoring? Yes, because there is a final data-independent date (i.e. July, 1986). This is completely non-informative w/ regards to the patients. So we can suspect a non-informative censoring and start investigations using Cox model.
2. What we have to do w/ the transplant? I.e., `event` has three levels: censored, transplant, dead; how we have to consider transplanted patients? In this case, clinicians answered that the transplant status is completely random! So, we can believe that it is a non-informative censoring.[1]

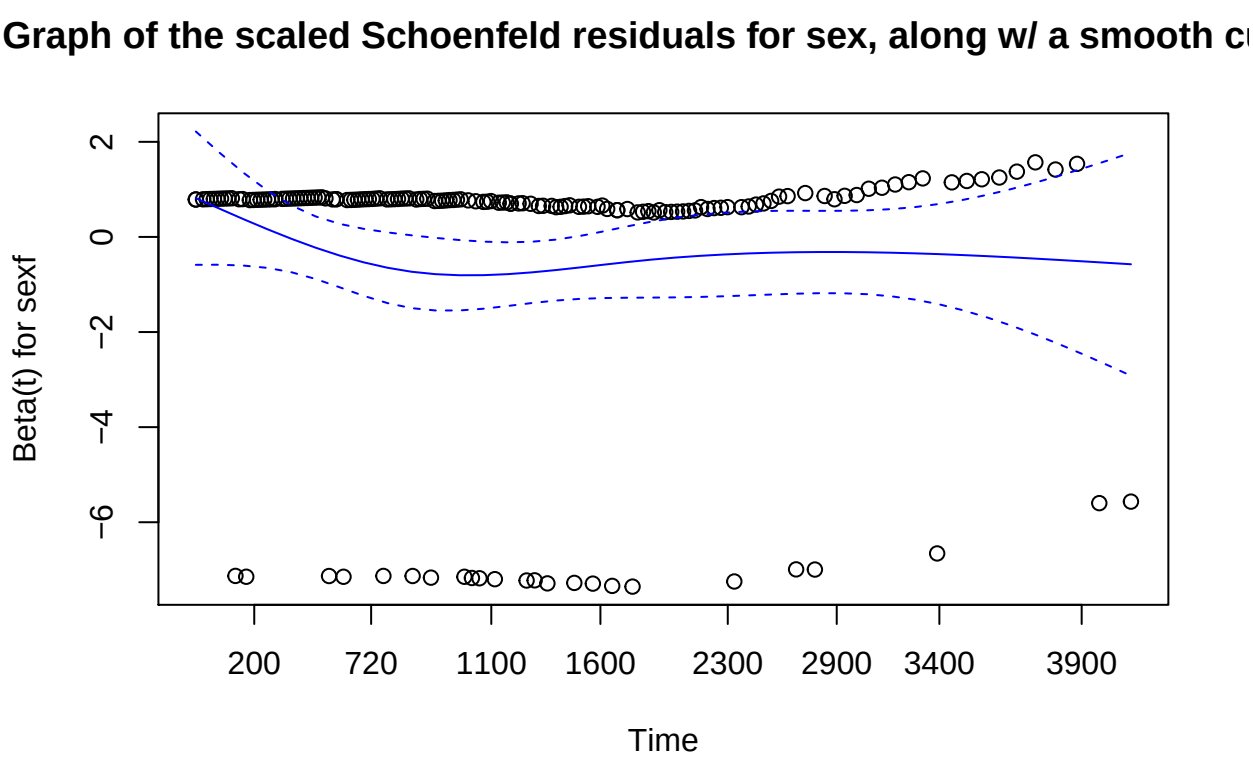
Moreover we have to consider, before to start, that `sex` is a categorical variable, so we have to check (only) the proportionality of the hazards.

```
# Using _survival_ (Cox model for proportional hazard against sex)
cox_sex <- coxph(Surv(time, status == 2) ~ sex,
  data = pbc_df
)

cox.zph(cox_sex)                          # test for proportionality of hazards
```

```
##          rho chisq     p
## sexf -0.0563 0.502 0.479
```

```
cox.zph(cox_sex) %>%
  plot(
    main = 'Graph of the scaled Schoenfeld residuals for sex, along w/ a smooth curve',
    col = 'blue'
)
```

[1] In reality this is not really true because who stay better is on the top of the list!




The proportional hazard assumption is not invalidate so we can continue w/ the analyses.

```
summary(cox_sex)
```

```
## Call:
## coxph(formula = Surv(time, status == 2) ~ sex, data = pbc_df)
##
##   n= 418, number of events= 161
##
##         coef exp(coef) se(coef)      z Pr(>|z|)
## sexf -0.3809    0.6833   0.2221 -1.714   0.0864 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##      exp(coef) exp(-coef) lower .95 upper .95
## sexf    0.6833      1.464    0.4421     1.056
##
## Concordance= 0.518  (se = 0.013 )
## Rsquare= 0.006   (max possible= 0.985 )
## Likelihood ratio test= 2.69  on 1 df,   p=0.101
## Wald test            = 2.94  on 1 df,   p=0.08645
## Score (logrank) test = 2.97  on 1 df,   p=0.08459
```

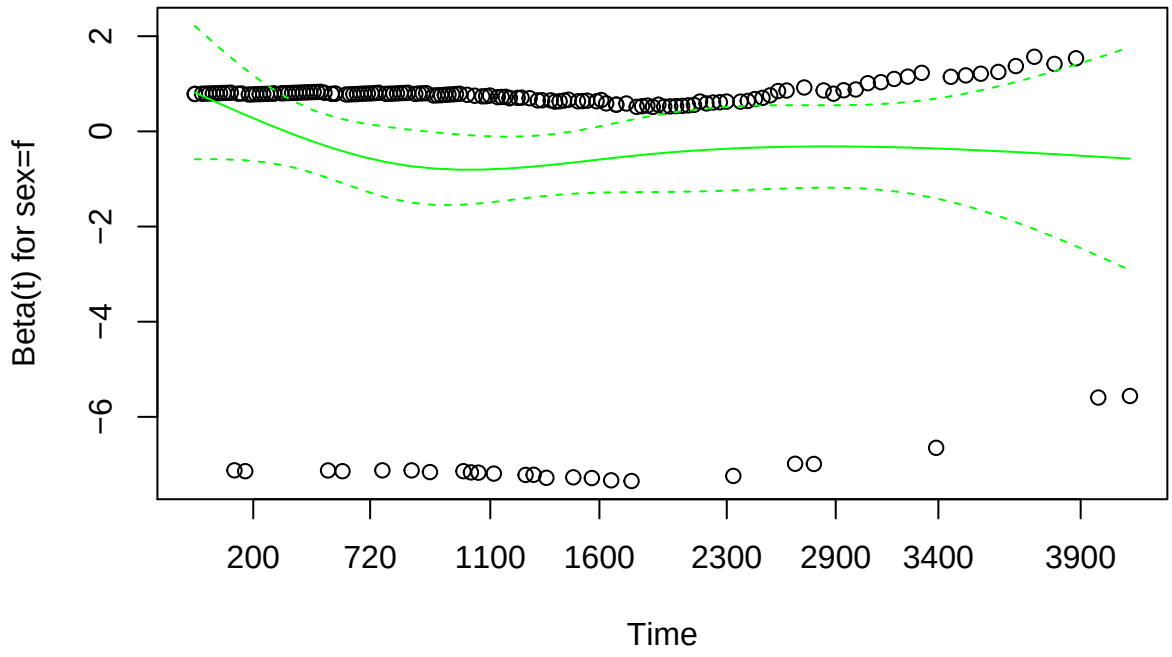
The effect of sex, viewed as hazard ration, say that if you are a female it seems that you have a lower risk to die, but it is not significant (i.e., $p$-value $> 0.05$ and CI include 1).

> Anyone have the same risk, 1, to die... What the hazard ration say is that if at the begin of a day you are alive, if you a re a woman you have 32% less probability to die before the end of the day respect a men.

```
# Using rms
rms_sex <- cph(Surv(time, status == 2) ~ sex,
  data = pbc_df,
  x    = TRUE,                    # to compute cox.zph, we need to store x and y
  y    = TRUE
)

cox.zph(rms_sex)
```

```
##           rho chisq     p
## sex=f -0.0563 0.501 0.479
```

```
cox.zph(rms_sex) %>%
  plot(col = 'green')                          # exactly the same results as before!
```




```
summary(rms_sex)       # a cleaner and more informative output, note Low and High
```

```
##              Effects              Response : Surv(time, status == 2)
##
##  Factor        Low High Diff. Effect  S.E.    Lower 0.95 Upper 0.95
##  sex - m:f     2   1    NA    0.38206 0.22205 -0.053149  0.81727
##   Hazard Ratio 2   1    NA    1.46530      NA  0.948240  2.26430
```

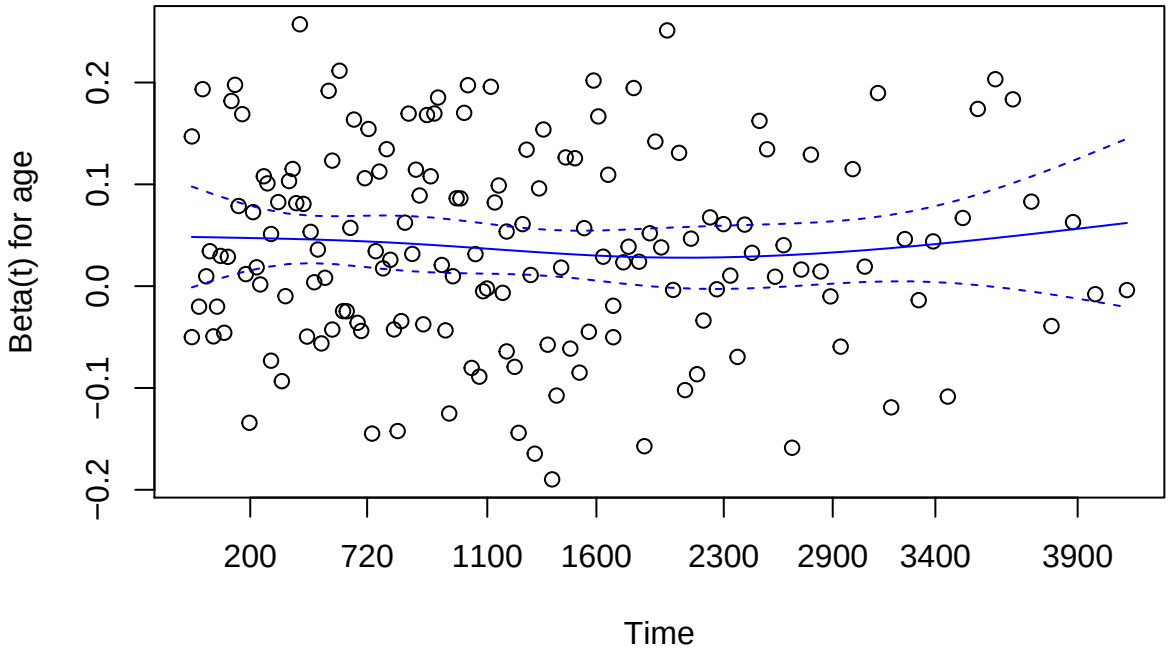
### 2.2.2 Impact of age on death

- we have to check for the proportional HR
- It is continuous variable, we have to check the the log-linearity too

```
# Using survival
cox_age <- coxph(Surv(time, status == 2) ~ age,
  data = pbc_df
)

cox.zph(cox_age)
```

```
##         rho chisq    p
## age -0.0304 0.139 0.71
```

```
cox.zph(cox_age) %>%
  plot(col = 'blue')
```



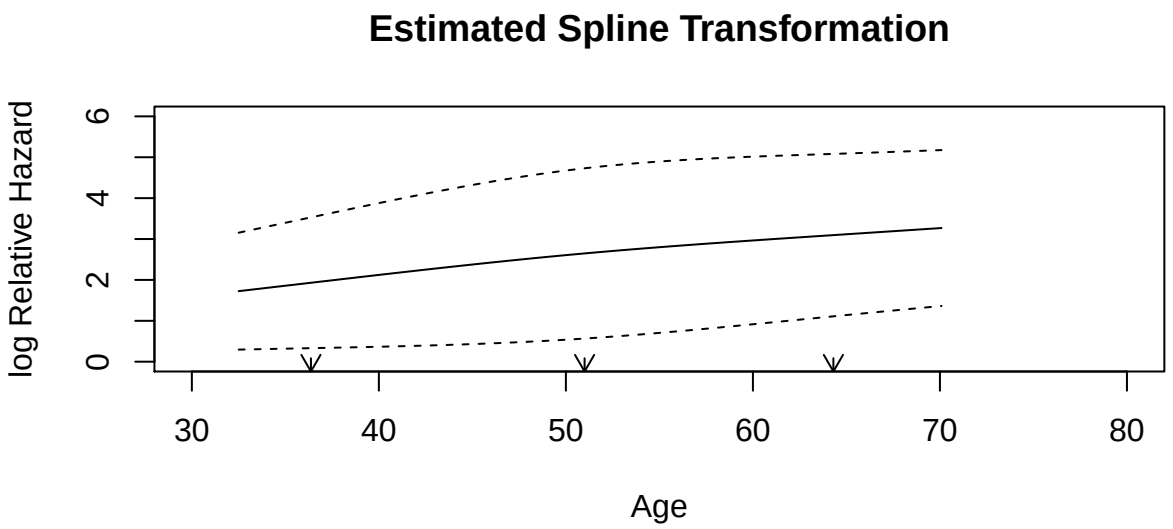

The proportional hazard hypothesis is not invalidated

> The outputs of rcspline.plot are a plot and a very long matrix w/ the values of xe, lower, upper. The latters are not of our interest, but there are no options to not get them. So we include the command into `invisible(capture.output(.))` box.[2]

```
invisible(capture.output(rcspline.plot(
  x       = pbc_df$age,
  y       = pbc_df$time,
  event   = pbc_df$status == 2,
  nk      = 3,                    # default are 5 knots, too mutch for this model
#  model   = 'cox',          # If event is present, model is assumed to be "cox"
  xlab    = 'Age',
  statloc = 'll'
```

[2] As suggested by *couthcommander* in https://github.com/CorradoLanera/SuDACDa/issues/2.

```
)))
```




The log-linearity is not invalidated

> Note: sometimes you *know* the answer for log-linearity (for any reason), in those cases do not test for it!! (It is not very powerful so for small sample sizes it never reject it)

```
summary(cox_age)
```

```
## Call:
## coxph(formula = Surv(time, status == 2) ~ age, data = pbc_df)
##
##   n= 418, number of events= 161
##
##         coef exp(coef) se(coef)     z Pr(>|z|)
## age 0.039185  1.039963 0.007847 4.994 5.92e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##     exp(coef) exp(-coef) lower .95 upper .95
## age      1.04     0.9616     1.024     1.056
##
## Concordance= 0.616  (se = 0.025 )
## Rsquare= 0.058   (max possible= 0.985 )
## Likelihood ratio test= 25.19  on 1 df,   p=5.205e-07
## Wald test            = 24.94  on 1 df,   p=5.922e-07
## Score (logrank) test = 25.3  on 1 df,   p=4.918e-07
```

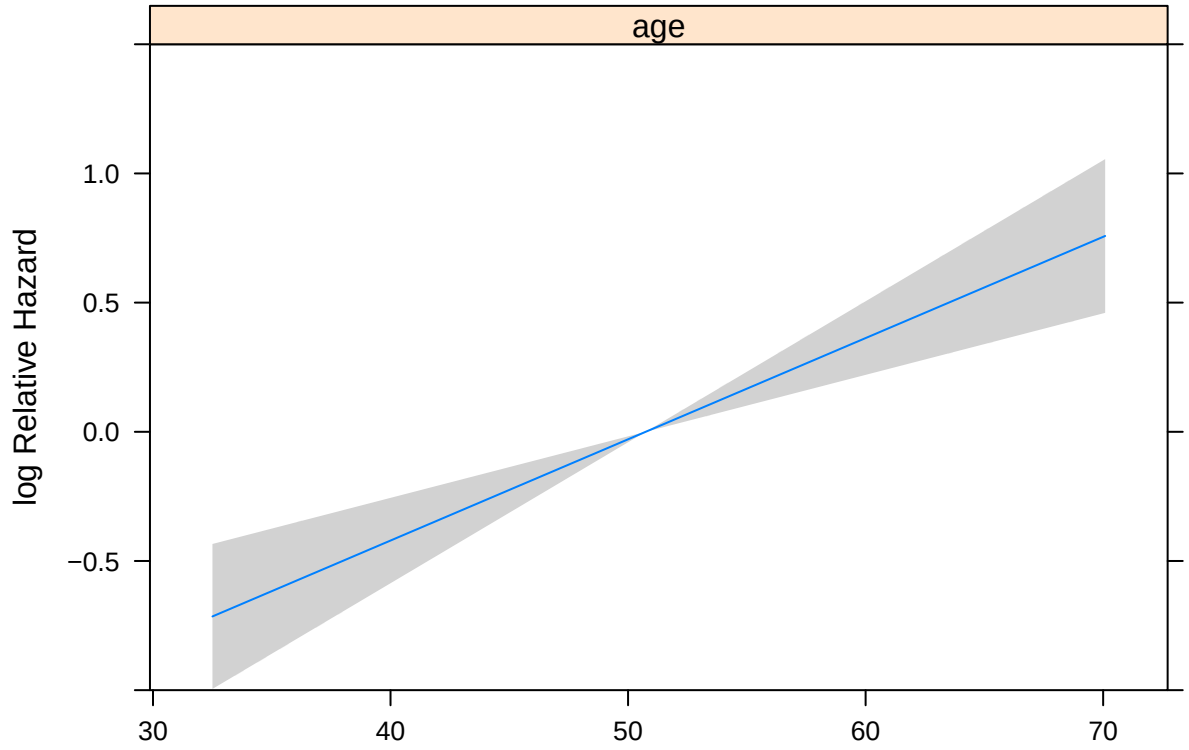
The effect is significant but too low to understand, so we can change the "measure of time" to expand it.

```
coxph(Surv(time, status == 2) ~ I(age / 10),          # consider 10 years as one
  data = pbc_df
) %>%
  summary
```

```
## Call:
## coxph(formula = Surv(time, status == 2) ~ I(age/10), data = pbc_df)
##
##   n= 418, number of events= 161
##
##              coef exp(coef) se(coef)     z Pr(>|z|)
## I(age/10) 0.39185   1.47972  0.07847 4.994 5.92e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##           exp(coef) exp(-coef) lower .95 upper .95
## I(age/10)      1.48     0.6758     1.269     1.726
##
## Concordance= 0.616  (se = 0.025 )
## Rsquare= 0.058   (max possible= 0.985 )
## Likelihood ratio test= 25.19  on 1 df,   p=5.205e-07
## Wald test            = 24.94  on 1 df,   p=5.922e-07
## Score (logrank) test = 25.3  on 1 df,   p=4.918e-07
```

Here, the effect is increased, but we have to pay attention the an increment of "one", here, corresponds to an increment of ten years!

```
# Using rms
rms_age <- cph(Surv(time, status == 2) ~ age,
  data = pbc_df,
  x    = TRUE,                    # to compute cox.zph, we need to store x and y
  y    = TRUE
)

Predict(rms_age) %>%
  plot
```




```
summary(rms_age)  # _rms_ show effects from the Lower to the Higher limit of IQR
```

```
##              Effects              Response : Surv(time, status == 2)
##
##  Factor        Low    High   Diff.  Effect  S.E.    Lower 0.95 Upper 0.95
##  age           42.832 58.241 15.409 0.60379 0.12091 0.36681    0.84076
##   Hazard Ratio 42.832 58.241 15.409 1.82900      NA 1.44310    2.31810
```

```
                  # and report the different between them as well as the HR, so
                  # we do not need to perform triky transformation which asks
                  # for an alterate interpretation of the result
```

In particular, the effect quite doubled in fifteen years.[3].

### 2.2.3 Impact of aspartate aminotransferase (ast) on death

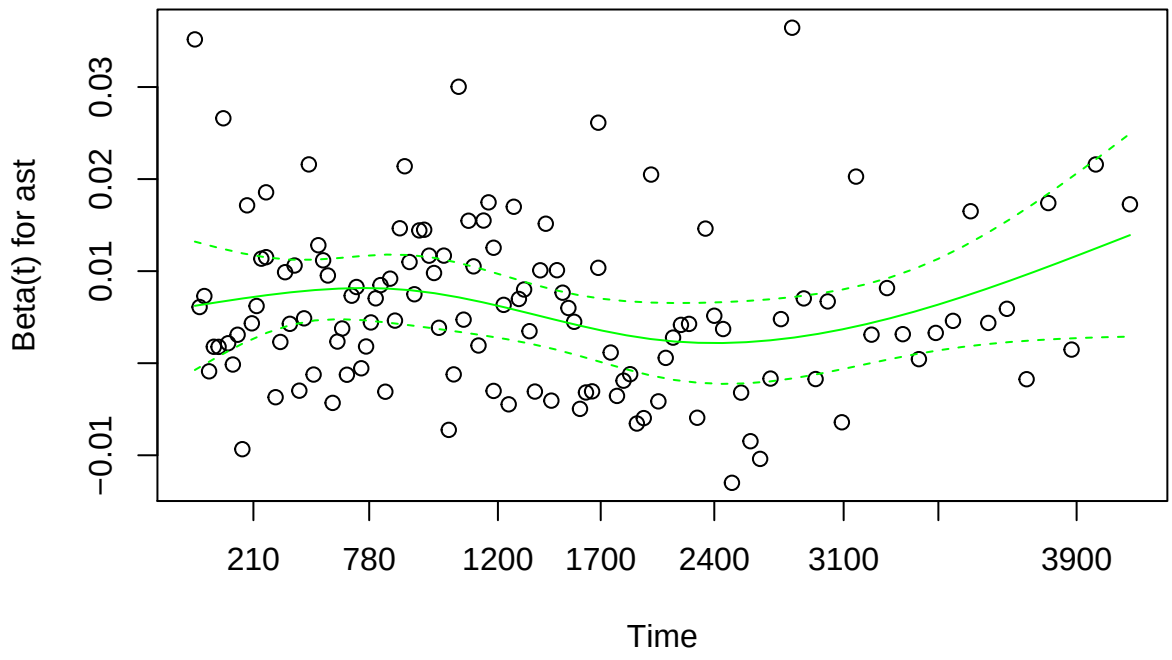
- same of age

```
rms_ast <- cph(Surv(time, status == 2) ~ ast,
  data = pbc_df,
  x    = TRUE,
  y    = TRUE
)

cox.zph(rms_ast)
```

[3]Good example in which only the clinicians know if it is an effect clinically relevant (deciding it **before** the analyses) or not

```
##         rho chisq     p
## ast -0.0641 0.274 0.601
```

```
cox.zph(rms_ast) %>%
  plot(col = 'green')
```

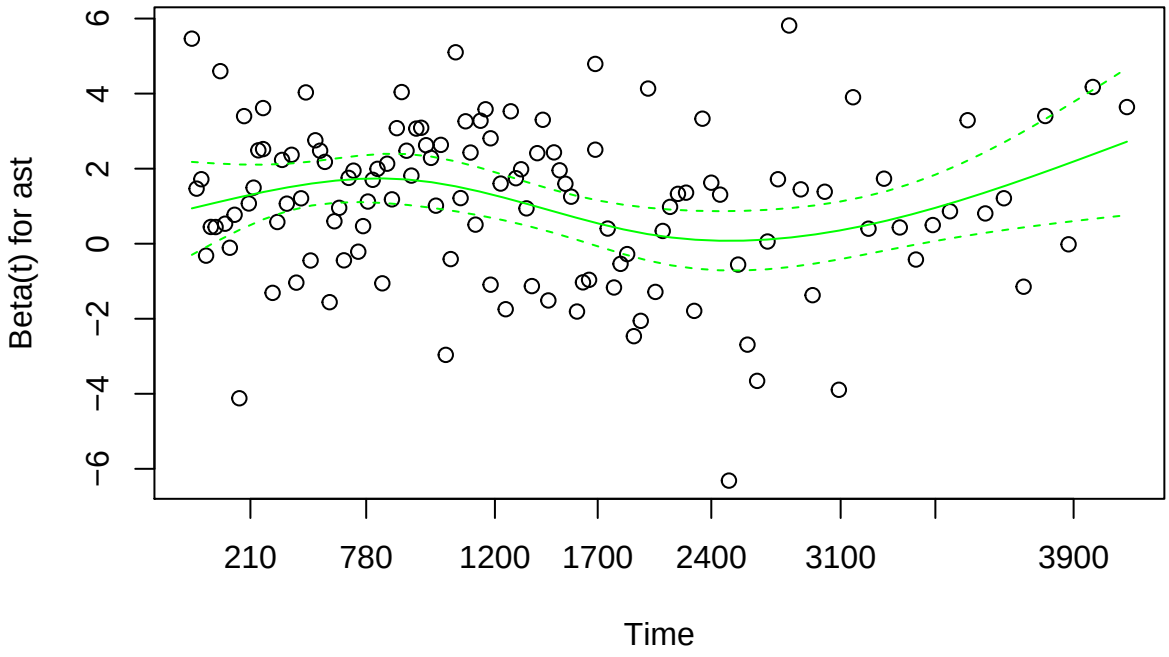



The proportional hazard assumption is not violated, but by the graph it seams not that linear. Try to transform it using the *log*() transformation.

```
log_ast <- cph(Surv(time, status == 2) ~ log(ast),
  data = pbc_df,
  x    = TRUE,
  y    = TRUE
)

cox.zph(log_ast)
```

```
##      rho chisq     p
## ast -0.1  1.13 0.289
```

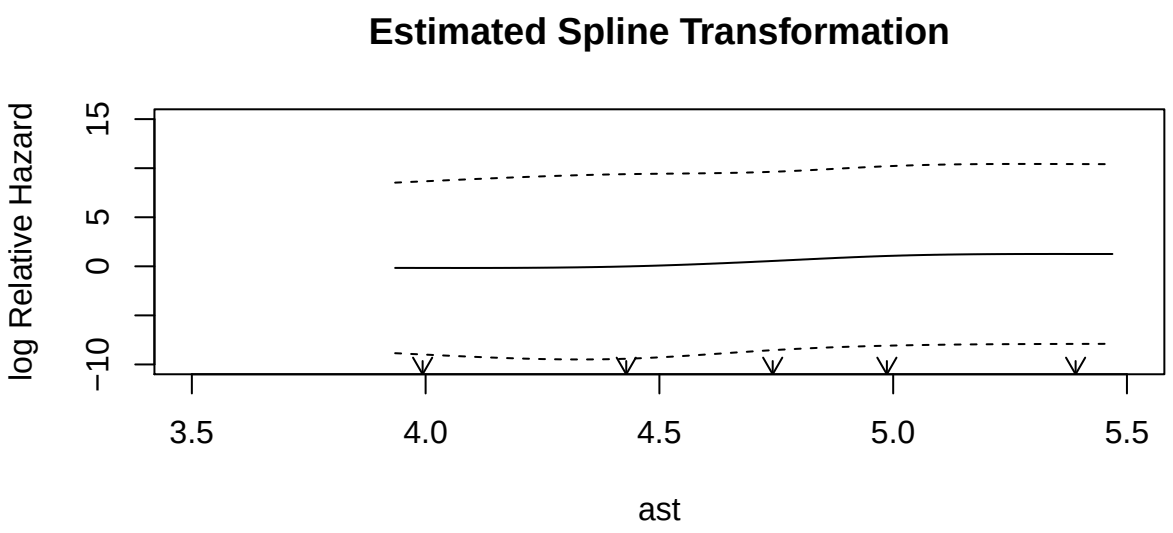
```
cox.zph(log_ast) %>%
  plot(col = 'green')
```




The situation is not much better...but we can say that there exists a line living in the middle of the band...so we are not very happy but we accept it.

Let's test for log-linearity

```
invisible(capture.output(rcspline.plot(
  x       = log(pbc_df$ast),
  y       = pbc_df$time,
  event   = pbc_df$status == 2,
  xlab    = 'ast',
  statloc = 'll'
)))
```




The log-linear assumption is not violated.

Finally, look at the effect of the log of `ast`

```
summary(log_ast)
```

```
##              Effects              Response : Surv(time, status == 2)
##
##  Factor        Low  High  Diff. Effect  S.E.    Lower 0.95 Upper 0.95
##  ast           80.6 151.9 71.3  0.69872 0.12499 0.45374    0.9437
##   Hazard Ratio 80.6 151.9 71.3  2.01120      NA 1.57420    2.5695
```

It is significantly protective, w/ doubling the effect between the borders of the IQR.

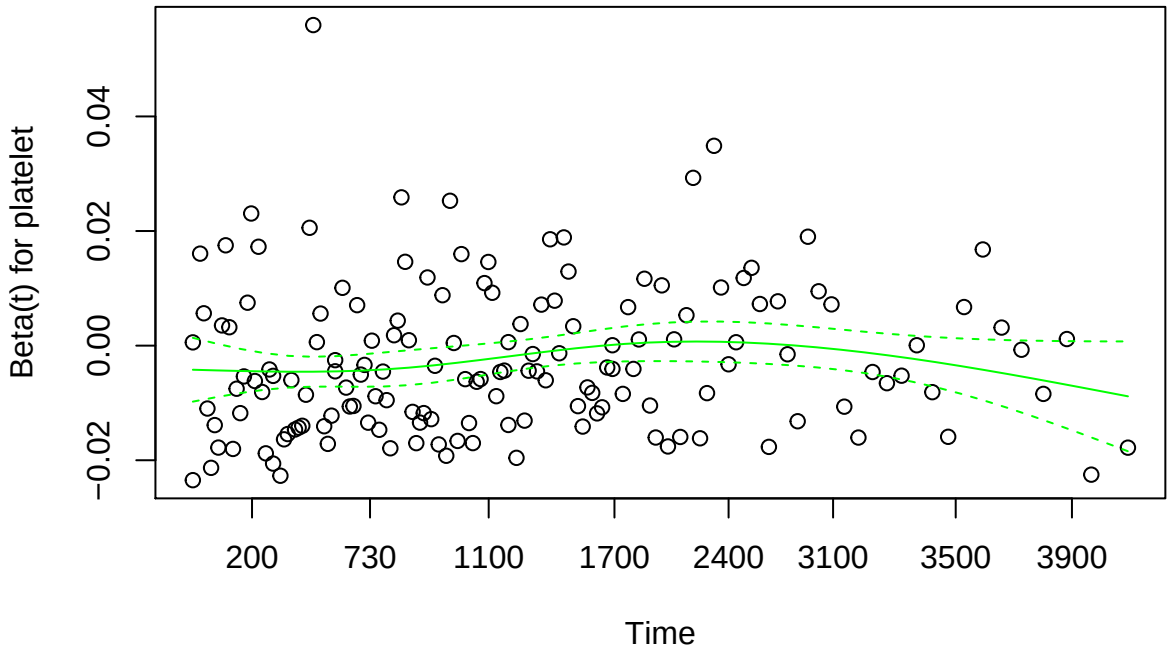
### 2.2.4 Impact of platelet on death

- same of age

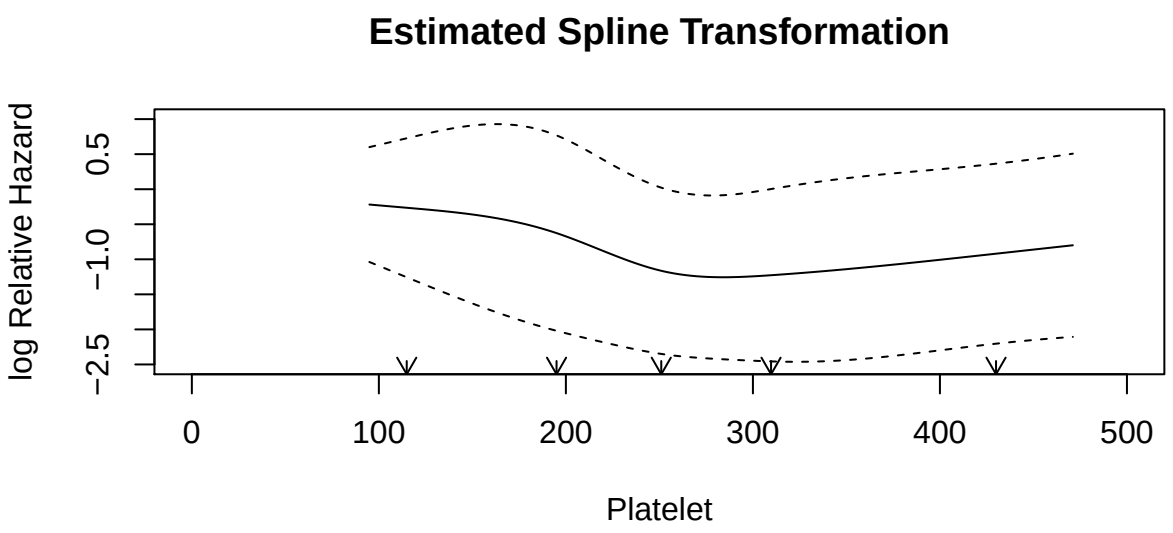
```
rms_platelet <- cph(Surv(time, status == 2) ~ platelet,
  data = pbc_df,
  x    = TRUE,
  y    = TRUE
)

cox.zph(rms_platelet)
```

```
##             rho chisq     p
## platelet 0.0688     1 0.316
```

```
cox.zph(rms_platelet) %>%
  plot(col = 'green')
```




```
invisible(capture.output(rcspline.plot(
  x       = pbc_df$platelet %>% as.numeric,
  y       = pbc_df$time,
  event   = pbc_df$status == 2,
  xlab    = 'Platelet',
  statloc = 'll'
)))
```

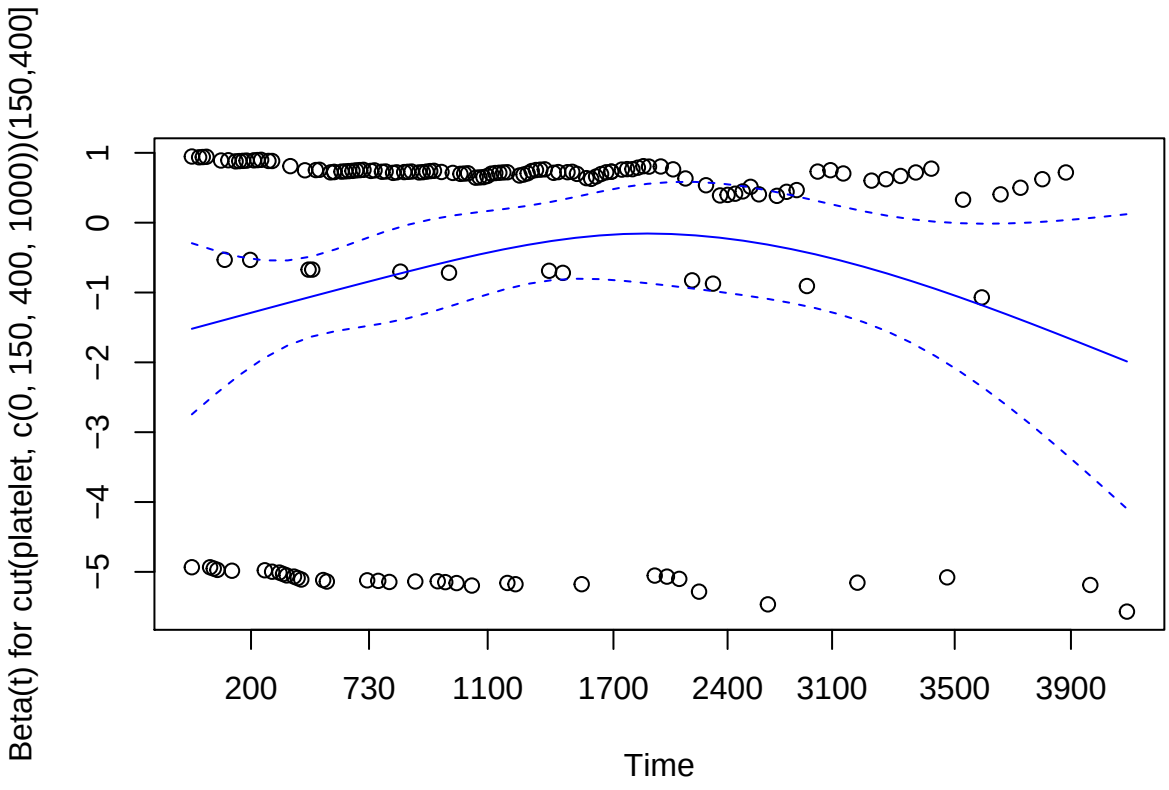


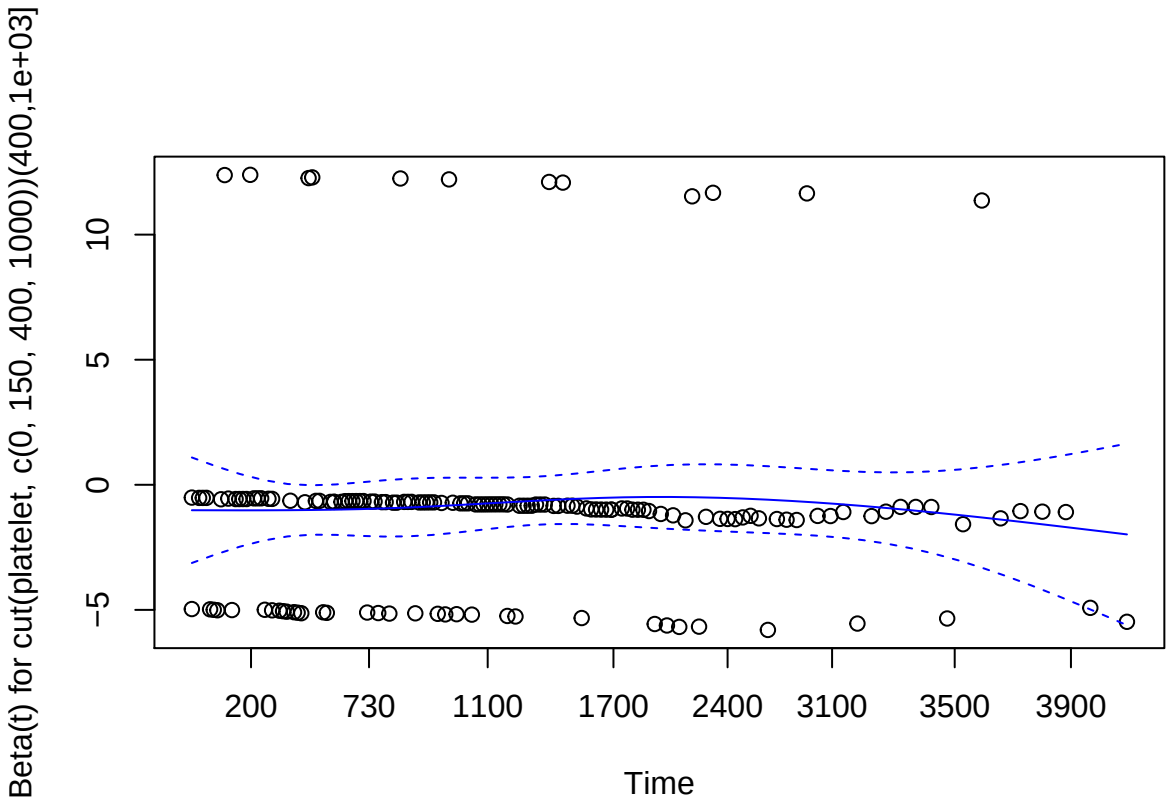

The log-linear plot has a U-shape so, standard transformation are not good. We can try to perform a categorization. Two strategy: 1. look at the log-linear plot and try to find a good cut-points, but we have to explain how we have defined them (and "use the p-value" is not a good strategy) 2. Use standard non related cutoff, such as median or quartiles

```
cox_cut_platelet <- coxph(
  Surv(time, status == 2) ~ cut(platelet, c(0, 150, 400, 1000)),
  data = pbc_df
)

cox.zph(cox_cut_platelet)
```

```
##                                                rho   chisq     p
## cut(platelet, c(0, 150, 400, 1000))(150,400]   0.06413 0.62938 0.428
## cut(platelet, c(0, 150, 400, 1000))(400,1e+03] 0.00345 0.00181 0.966
## GLOBAL                                            NA 0.74607 0.689
```

```
cox.zph(cox_cut_platelet) %>%
  plot(col = 'blue')
```


Beta(t) for cut(platelet, c(0, 150, 400, 1000))(150,400]
1
0
−1
−2
−3
−4
−5
200
730
1100
1700
2400
3100
3500
3900
Time


Beta(t) for cut(platelet, c(0, 150, 400, 1000))(400,1e+03]
10
5
0
−5
200
730
1100
1700
2400
3100
3500
3900
Time

```
summary(cox_cut_platelet)
```

```
## Call:
## coxph(formula = Surv(time, status == 2) ~ cut(platelet, c(0,
##     150, 400, 1000)), data = pbc_df)
##
##   n= 407, number of events= 155
##    (11 observations deleted due to missingness)
##
##                                                    coef exp(coef) se(coef)
## cut(platelet, c(0, 150, 400, 1000))(150,400]   -0.7164    0.4885   0.1948
## cut(platelet, c(0, 150, 400, 1000))(400,1e+03] -0.8445    0.4298   0.3352
##                                                      z Pr(>|z|)
## cut(platelet, c(0, 150, 400, 1000))(150,400]   -3.678 0.000235 ***
## cut(platelet, c(0, 150, 400, 1000))(400,1e+03] -2.519 0.011755 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##                                                exp(coef) exp(-coef)
## cut(platelet, c(0, 150, 400, 1000))(150,400]      0.4885      2.047
## cut(platelet, c(0, 150, 400, 1000))(400,1e+03]    0.4298      2.327
##                                                lower .95 upper .95
## cut(platelet, c(0, 150, 400, 1000))(150,400]      0.3335    0.7156
## cut(platelet, c(0, 150, 400, 1000))(400,1e+03]    0.2228    0.8290
##
## Concordance= 0.561  (se = 0.018 )
## Rsquare= 0.031   (max possible= 0.984 )
## Likelihood ratio test= 12.68  on 2 df,   p=0.001766
## Wald test            = 14.53  on 2 df,   p=0.0006987
## Score (logrank) test = 15.19  on 2 df,   p=0.0005032
```

But here the reference level, i.e. the contrast, is the lower level but the interested is what happen if we lie above or over the standard values, so we have to relevel the category to make the medium level as the reference one, i.e. the first.

```
pbc_df <- pbc_df %>%
  mutate(
    platelet_ref = cut(pbc_df$platelet, c(0, 150, 400, 1000)) %>%
                      relevel(ref = "(150,400]")
)

cox_relev_platelet <- coxph(Surv(time, status == 2) ~ platelet_ref,
  data = pbc_df
)

summary(cox_relev_platelet)
```

```
## Call:
## coxph(formula = Surv(time, status == 2) ~ platelet_ref, data = pbc_df)
##
##   n= 407, number of events= 155
##    (11 observations deleted due to missingness)
##
##                            coef exp(coef) se(coef)      z Pr(>|z|)
## platelet_ref(0,150]      0.7164    2.0471   0.1948  3.678 0.000235 ***
```

```
## platelet_ref(400,1e+03] -0.1281    0.8798   0.3046 -0.420 0.674235
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##                          exp(coef) exp(-coef) lower .95 upper .95
## platelet_ref(0,150]         2.0471     0.4885    1.3975     2.999
## platelet_ref(400,1e+03]     0.8798     1.1366    0.4843     1.598
##
## Concordance= 0.561  (se = 0.018 )
## Rsquare= 0.031   (max possible= 0.984 )
## Likelihood ratio test= 12.68  on 2 df,   p=0.001766
## Wald test            = 14.53  on 2 df,   p=0.0006987
## Score (logrank) test = 15.19  on 2 df,   p=0.0005032
```

## 2.3 Investigation on adjusted variables and interactions

Clinician: what is the effect of treatment (`trt`) on death?

```
cph(Surv(time, status == 2) ~ trt,
  data = pbc_df
) %>%
  summary
```

```
##              Effects              Response : Surv(time, status == 2)
##
##  Factor        Low High Diff. Effect    S.E.    Lower 0.95 Upper 0.95
##  trt           1   2    1     -0.057189 0.17916 -0.40835   0.29397
##   Hazard Ratio 1   2    1      0.944420      NA  0.66475   1.34170
```

No significant effect for treatment.

Clinician: an adjusted w/ edema?

```
cph(Surv(time, status == 2) ~ trt + edema,
  data = pbc_df
) %>%
  summary
```

```
##              Effects              Response : Surv(time, status == 2)
##
##  Factor        Low High Diff. Effect    S.E.    Lower 0.95 Upper 0.95
##  trt           1   2    1     -0.065946 0.17953 -0.41781    0.28592
##   Hazard Ratio 1   2    1      0.936180      NA  0.65849    1.33100
##  edema         0   1    1      2.280700 0.25761  1.77580    2.78560
##   Hazard Ratio 0   1    1      9.783600      NA  5.90510   16.21000
```

No effect for treatment nor edema

Clinicians: and what about their interaction?[4]

```
cph(Surv(time, status == 2) ~ trt * edema,
  data = pbc_df
) %>%
  summary
```

[4]The answer here should be "if there are no marginal significant effect is has no sense to look at the interaction terms!".

```
##              Effects              Response : Surv(time, status == 2)
##
##  Factor        Low High Diff. Effect   S.E.    Lower 0.95 Upper 0.95
##  trt           1   2    1     -0.24014 0.22959 -0.69012    0.20984
##   Hazard Ratio 1   2    1      0.78652      NA  0.50151    1.23350
##  edema         0   1    1      2.62340 0.37161  1.89500    3.35170
##   Hazard Ratio 0   1    1     13.78200      NA  6.65280   28.55200
##
## Adjusted to: trt=1 edema=0.5
```

No effect.

Clinicians: and what about adjusted w/ stage?

```
adj_pbc <- pbc_df %>%
  mutate(stage_fct = factor(stage))

dd <- datadist(adj_pbc)

cph(Surv(time, status == 2) ~ trt + stage_fct,
  data = adj_pbc
) %>%
  summary
```

```
##              Effects              Response : Surv(time, status == 2)
##
##  Factor          Low High Diff. Effect   S.E.    Lower 0.95 Upper 0.95
##  trt             1   2     1    -0.14713 0.17989 -0.49971    0.20545
##   Hazard Ratio   1   2     1     0.86318      NA  0.60671    1.22810
##  stage_fct - 1:3 3   1    NA    -2.17290 1.01080 -4.15410   -0.19176
##   Hazard Ratio   3   1    NA     0.11384      NA  0.01570    0.82550
##  stage_fct - 2:3 3   2    NA    -0.54826 0.29344 -1.12340    0.02687
##   Hazard Ratio   3   2    NA     0.57795      NA  0.32517    1.02720
##  stage_fct - 4:3 3   4    NA     0.91613 0.19771  0.52862    1.30360
##   Hazard Ratio   3   4    NA     2.49960      NA  1.69660    3.68270
```

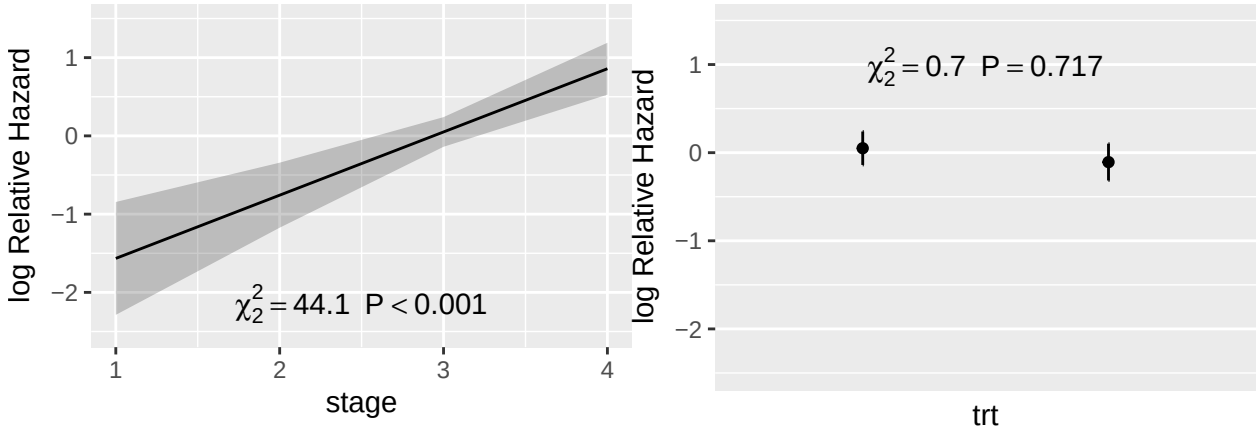
Treatment still w/ no significant effect. `stage` has some effects, i.e. from the 3 to 1 or to 4,

Clinicians: oh, so let's look at the interactions!

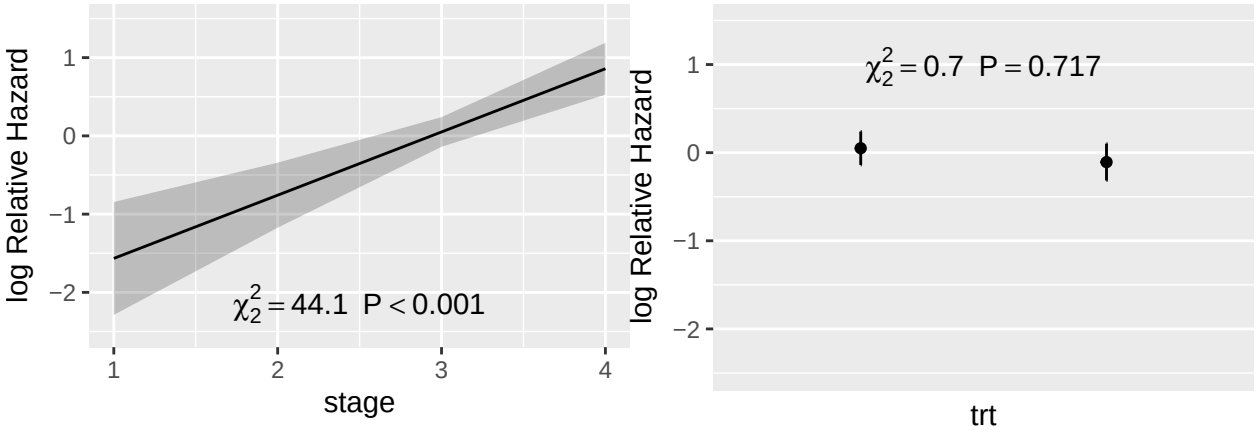
```
rms_trt_stage <- cph(Surv(time, status == 2) ~ trt * stage,
  data = adj_pbc
)

summary(rms_trt_stage)
```

```
##              Effects              Response : Surv(time, status == 2)
##
##  Factor        Low High Diff. Effect   S.E.    Lower 0.95 Upper 0.95
##  trt           1   2    1     -0.15717 0.20284 -0.55473    0.24039
##   Hazard Ratio 1   2    1      0.85456      NA  0.57423    1.27170
##  stage         2   4    2      1.61630 0.33135  0.96682    2.26570
##   Hazard Ratio 2   4    2      5.03420      NA  2.62960    9.63790
##
## Adjusted to: trt=1 stage=3
```

```
Predict(rms_trt_stage) %>%
  ggplot(anova = anova(rms_trt_stage), pval = TRUE)
```




```
Predict(rms_trt_stage) %>%
  ggplot(anova = anova(rms_trt_stage), pval = TRUE)
```




Treatment continue to have no effect

## 2.4 Longitudinal suvival data analayses

Load a data-set, update the `datadist()` for the **rms** package, and take a look at the data

```
pbcseq_df <- as_tibble(pbcseq)
dd <- datadist(pbcseq_df)

pbcseq_df
```

```
## # A tibble: 1,945 x 19
##       id futime status   trt      age    sex   day ascites hepato spiders
##    <int>  <int>  <int> <int>    <dbl> <fctr> <int>   <int>  <int>   <int>
##  1     1    400      2     1 58.76523      f     0       1      1       1
##  2     1    400      2     1 58.76523      f   192       1      1       1
##  3     2   5169      0     1 56.44627      f     0       0      1       1
##  4     2   5169      0     1 56.44627      f   182       0      1       1
##  5     2   5169      0     1 56.44627      f   365       0      1       1
##  6     2   5169      0     1 56.44627      f   768       0      1       1
##  7     2   5169      0     1 56.44627      f  1790       1      1       1
##  8     2   5169      0     1 56.44627      f  2151       1      1       1
##  9     2   5169      0     1 56.44627      f  2515       1      1       1
## 10     2   5169      0     1 56.44627      f  2882       1      1       1
## # ... with 1,935 more rows, and 9 more variables: edema <dbl>, bili <dbl>,
## #   chol <int>, albumin <dbl>, alk.phos <int>, ast <dbl>, platelet <int>,
```

```
## #   protime <dbl>, stage <int>
```

- The only tricky task is to correctly manage and prepare the data. Our proposal take advantage of the `dplyr` functionality

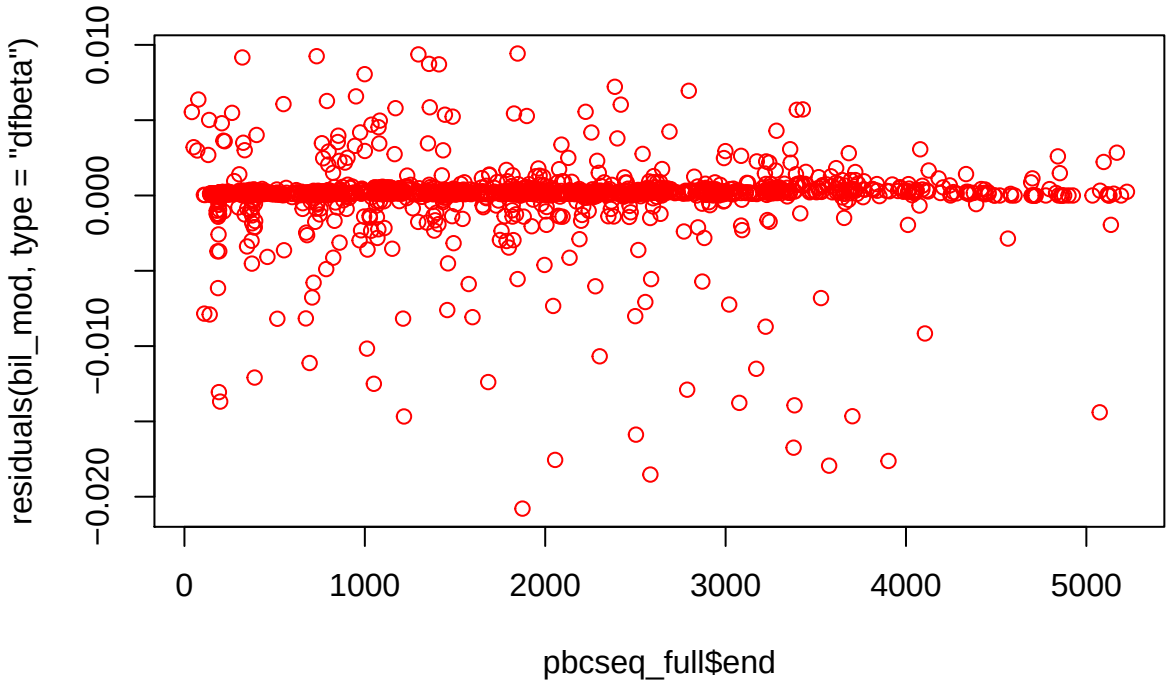
```
pbcseq_full <- pbcseq_df %>%
  group_by(id) %>%                    # perform all the next according to the id
  mutate(
    start  = day,                                  # just to have consistent names
    end    = lead(day),           # the end is "the next start" (last will be NA)
    status = if_else(is.na(end), status, 0L),
    end    = if_else(is.na(end), futime, end) # fill the NA-ends (i.e. the lasts)
                                              # w/ the real end
)
```

### 2.4.1 Impact of bilurubine of death

```
bil_mod <- cph(
  Surv(time = start, time2 = end, event = status == 2L) ~ log(bili),
  data = pbcseq_full,
  x    = TRUE,
  y    = TRUE
)

summary(bil_mod)
```

```
##              Effects              Response : Surv(time = start, time2 = end, event = status == 2)
##
##  Factor        Low High Diff. Effect S.E.    Lower 0.95 Upper 0.95
##  bili          0.8 3.9  3.1   2.0410 0.13391 1.7786      2.3035
##   Hazard Ratio 0.8 3.9  3.1   7.6984      NA 5.9213     10.0090
```

```
plot(
  x   = pbcseq_full$end,
  y   = residuals(bil_mod, type = 'dfbeta'),
  col = 'red'
)
```

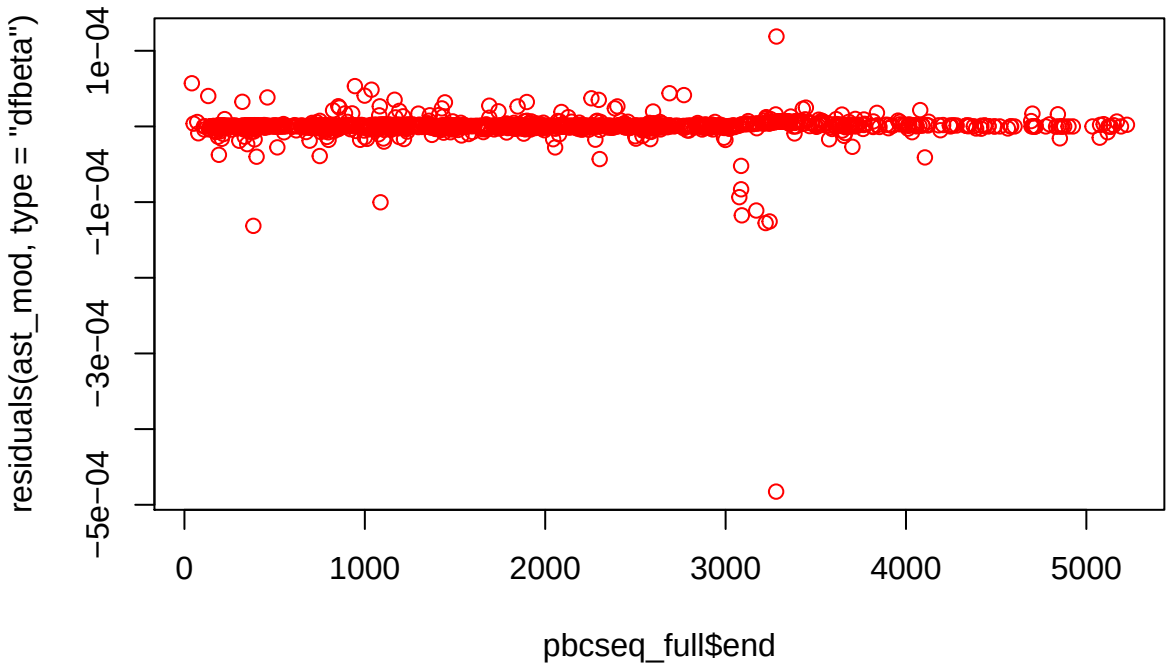



### 2.4.2 Impact of ast

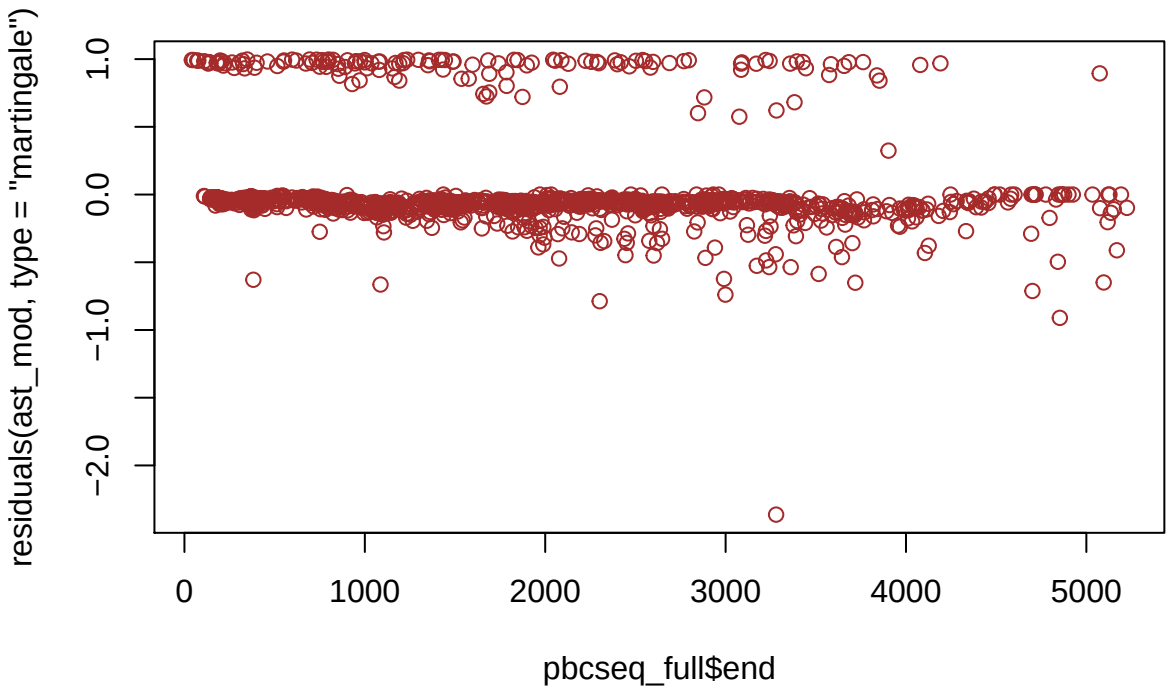
```
ast_mod <- cph(
  Surv(time = start, time2 = end, event = status == 2L) ~ ast,
  data = pbcseq_full,
  x    = TRUE,
  y    = TRUE
)

summary(ast_mod)
```

```
##              Effects              Response : Surv(time = start, time2 = end, event = status == 2)
##
##  Factor        Low High Diff. Effect S.E.     Lower 0.95 Upper 0.95
##  ast           72  155  83    0.2875 0.044222 0.20083    0.37417
##   Hazard Ratio 72  155  83    1.3331       NA 1.22240    1.45380
```

```
plot(
  x   = pbcseq_full$end,
  y   = residuals(ast_mod, type = 'dfbeta'),
  col = 'red'
)
```




```
plot(
  x   = pbcseq_full$end,
  y   = residuals(ast_mod, type = 'martingale'),
  col = 'brown'
)
```




What happen w/ the strange observations? We try to find which is that outlier.

```
# look at the residual caracteristics
residuals(ast_mod, type = 'martingale') %>%
  describe
```

```
## .
##          n    missing   distinct       Info       Mean        Gmd
##       1945          0       1855          1 -2.395e-17     0.1868
##        .05        .10        .25        .50        .75        .90
##   -0.16941   -0.11177   -0.07449   -0.05021   -0.03303   -0.01672
##        .95
##    0.95170
##
## lowest : -2.3632251 -0.9104219 -0.7874086 -0.7390614 -0.7120605
## highest:  0.9960346  0.9963091  0.9968442  0.9970148  0.9970895
```

```
# take the id of the lowest
strange_id <- residuals(ast_mod, type = 'martingale') %>%
  which.min

# take a look to the ast
pbcseq_full$ast %>% describe
```

```
## .
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##     1945        0      418        1    122.7    74.42     41.9     51.2
##      .25      .50      .75      .90      .95
##     72.0    107.0    155.0    209.3    250.7
```

```
##
## lowest :    6.2   21.0   21.7   22.0   23.3, highest:  473.0  655.7  685.1  918.0 1205.0
```

```
# check the id
pbcseq_full$ast[[strange_id]]
```

```
## [1] 1205
```

Here is another example in which the opinion of a clinician is mandatory, i.e. we cannot decide if ignore outliers, which ones, etc

## 2.5 Prognostic model

### 2.5.1 prognostic model w/ àscites, edema, sex, bili, ast, platelet, stage

```
# prepare an ad hoc data frame
pbc_updated <- pbc_df %>%
  mutate(
    bili_log     = log(bili),
    ast_log      = log(ast),
    platelet_ref = platelet_ref, # we have already defined it
    stage_fct    = as.factor(stage)
  )

dd <- datadist(pbc_updated)

# take a look at them
pbc_updated %>%
  dplyr::select(
    ascites, edema, sex, bili_log, ast_log, platelet_ref, stage_fct
  ) %>%
  describe
```

```
## .
##
##  7  Variables      418  Observations
## ---------------------------------------------------------------------------
## ascites
##        n  missing distinct     Info      Sum     Mean      Gmd
##      312      106        2    0.213       24  0.07692   0.1425
##
## ---------------------------------------------------------------------------
## edema
##        n  missing distinct     Info     Mean      Gmd
##      418        0        3    0.391   0.1005   0.1756
##
## Value        0.0   0.5   1.0
## Frequency    354    44    20
## Proportion 0.847 0.105 0.048
## ---------------------------------------------------------------------------
## sex
##        n  missing distinct
##      418        0        2
```

```
##
## Value          m     f
## Frequency     44   374
## Proportion 0.105 0.895
## ---------------------------------------------------------------------------
## bili_log
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##      418        0       98    0.998   0.5715    1.149  -0.6931  -0.5108
##      .25      .50      .75      .90      .95
##  -0.2231   0.3365   1.2238   2.0832   2.6391
##
## lowest : -1.2039728 -0.9162907 -0.6931472 -0.5108256 -0.3566749
## highest:  3.0726933  3.1135153  3.1986731  3.2386785  3.3322045
## ---------------------------------------------------------------------------
## ast_log
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##      312      106      179        1     4.71   0.5075    3.994    4.102
##      .25      .50      .75      .90      .95
##    4.389    4.742    5.023    5.280    5.390
##
## lowest : 3.271468 3.345685 3.734092 3.770459 3.806662
## highest: 5.662960 5.700945 5.794841 5.823046 6.125230
## ---------------------------------------------------------------------------
## platelet_ref
##        n  missing distinct
##      407       11        3
##
## Value        (150,400]     (0,150] (400,1e+03]
## Frequency          311          61          35
## Proportion       0.764       0.150       0.086
## ---------------------------------------------------------------------------
## stage_fct
##        n  missing distinct
##      412        6        4
##
## Value          1     2     3     4
## Frequency     21    92   155   144
## Proportion 0.051 0.223 0.376 0.350
## ---------------------------------------------------------------------------
```

There are 11 basic df (one each continuous variable and one-minus-n-level for the categorical one), so to use all of them we need at least 110 obs. Data has 418, this allow us to use a more complex model, w/ some interaction, splines, etc (more or less other $15 - -30$ df).

We decide (following suggestions from Harrell Jr (2015)) to consider splines for any continuous variable (w/ 3 knots) and consider sex interaction w/ them and the other numerical variables, leading to near 20 df.

```
data_used <- pbc_updated %>%
  dplyr::select(status, time,
    sex, ascites, edema, bili_log, ast_log, platelet_ref, stage_fct
)

dd <- datadist(data_used)
```

```r
# all the data-set
cph(
  Surv(time, status == 2) ~
    sex * (ascites + edema + rcs(bili_log, 3) + rcs(ast_log, 3)) +
    platelet_ref + stage_fct,
  data = data_used
) %>%
  summary
```

```
##              Effects              Response : Surv(time, status == 2)
##
##  Factor                                Low      High   Diff.   Effect
##  ascites                                0.00000 1.0000 1.00000  1.1613000
##   Hazard Ratio                          0.00000 1.0000 1.00000  3.1941000
##  edema                                  0.00000 1.0000 1.00000  0.6370800
##   Hazard Ratio                          0.00000 1.0000 1.00000  1.8910000
##  bili_log                              -0.22314 1.2238 1.44690  1.2273000
##   Hazard Ratio                         -0.22314 1.2238 1.44690  3.4120000
##  ast_log                                4.38950 5.0232 0.63372  0.3791100
##   Hazard Ratio                          4.38950 5.0232 0.63372  1.4610000
##  sex - m:f                              2.00000 1.0000      NA  2.0211000
##   Hazard Ratio                          2.00000 1.0000      NA  7.5463000
##  platelet_ref - (0,150]:(150,400]       1.00000 2.0000      NA  0.0907330
##   Hazard Ratio                          1.00000 2.0000      NA  1.0950000
##  platelet_ref - (400,1e+03]:(150,400]   1.00000 3.0000      NA  0.0059127
##   Hazard Ratio                          1.00000 3.0000      NA  1.0059000
##  stage_fct - 1:3                        3.00000 1.0000      NA -1.2551000
##   Hazard Ratio                          3.00000 1.0000      NA  0.2850400
##  stage_fct - 2:3                        3.00000 2.0000      NA -0.3858200
##   Hazard Ratio                          3.00000 2.0000      NA  0.6798900
##  stage_fct - 4:3                        3.00000 4.0000      NA  0.6523900
##   Hazard Ratio                          3.00000 4.0000      NA  1.9201000
##  S.E.    Lower 0.95 Upper 0.95
##  0.31762  0.538770   1.78380
##       NA  1.713900   5.95260
##  0.35904 -0.066626   1.34080
##       NA  0.935550   3.82200
##  0.28103  0.676490   1.77810
##       NA  1.967000   5.91870
##  0.18497  0.016587   0.74164
##       NA  1.016700   2.09940
##  0.54588  0.951150   3.09100
##       NA  2.588700  21.99800
##  0.25377 -0.406650   0.58812
##       NA  0.665870   1.80060
##  0.35526 -0.690380   0.70221
##       NA  0.501380   2.01820
##  1.03160 -3.277100   0.76682
##       NA  0.037739   2.15290
##  0.30581 -0.985210   0.21356
##       NA  0.373360   1.23810
##  0.23174  0.198190   1.10660
##       NA  1.219200   3.02400
##
```

```
## Adjusted to: sex=f ascites=0 edema=0.5 bili_log=0.3364722 ast_log=4.74232
```

```
# W/out missing data, and w/ beckward stepwise variable selection
cph(
  Surv(time, status == 2) ~
    sex + ascites + edema + bili_log + ast_log + platelet_ref + stage_fct,
  data = pbc_updated %>%
    filter(complete.cases(.))
) %>%
  step(trace = 0) %>%
  summary
```

```
##              Effects              Response : Surv(time, status == 2)
##
##  Factor          Low      High   Diff.  Effect   S.E.    Lower 0.95
##  ascites          0.00000 1.0000 1.0000  0.62140 0.33765 -0.040376
##   Hazard Ratio    0.00000 1.0000 1.0000  1.86150      NA  0.960430
##  edema            0.00000 1.0000 1.0000  1.18730 0.33082  0.538900
##   Hazard Ratio    0.00000 1.0000 1.0000  3.27820      NA  1.714100
##  bili_log        -0.22314 1.2238 1.4469  1.24480 0.15866  0.933880
##   Hazard Ratio   -0.22314 1.2238 1.4469  3.47240      NA  2.544400
##  sex - m:f        2.00000 1.0000     NA  0.52890 0.25407  0.030938
##   Hazard Ratio    2.00000 1.0000     NA  1.69710      NA  1.031400
##  stage_fct - 1:3  3.00000 1.0000     NA -1.48070 1.01500 -3.470100
##   Hazard Ratio    3.00000 1.0000     NA  0.22748      NA  0.031115
##  stage_fct - 2:3  3.00000 2.0000     NA -0.29414 0.31748 -0.916400
##   Hazard Ratio    3.00000 2.0000     NA  0.74517      NA  0.399960
##  stage_fct - 4:3  3.00000 4.0000     NA  0.53395 0.22987  0.083422
##   Hazard Ratio    3.00000 4.0000     NA  1.70570      NA  1.087000
##  Upper 0.95
##  1.28320
##  3.60810
##  1.83570
##  6.26950
##  1.55580
##  4.73890
##  1.02690
##  2.79230
##  0.50872
##  1.66320
##  0.32812
##  1.38840
##  0.98449
##  2.67640
```

# Chapter 3

# *Wednesday*: Competing risk

## 3.1 Key (operative) concepts

1. Patient are exposed simultaneously to $k(\geq 2)$ causes
2. Effect Free Survival (EFS) is univariate, i.e. only the First Observed Event (FOE) is considered and of interest
3. The interest is not in the survival model

   "At $\infty$ all individuals will not die in the ICU"
4. Type of observed time

- Cansored (conventionally coded w/ 0)
- Failure w/ a FOE different from the last absorbing one (coded w/ $1 - -k - 1$)
- Failure w/ the FOE as the last absorbing event (coded w/ $k$)
- $T_k = min\Big(\tilde{T_k}^{D_k} | D_k \in \{\text{causes of failure for } k\}\Big)$

5. Cumulative Incidence Function (CIF) do not require independence between causes
6. In competing risk, K-M is biased, i.e. overestimates the CIF (because it the independence assumption is violated)
7. Tests

- w/out competing risk: log-rank
- w/ competing risk: modified $\chi^2$ (Gray, 1988)

8. Regression strategies for competing risk

- Case Specific Hazard Ratio (CS-HR) — Cox, useful for clinical interests (present it for each competing risk taken singularly)
- Subdistribution Hazard Ratio (SHR) — Fine-Gray, useful for administrative] interests (present it for the global risk considering the contribution of each competing one)

  Test the proportional hazard assumption for SHR

  There are formulas for the sample size calculation when considering competing risk

## 3.2 Data manipulation

```
set.seed(171004)
data(mgus, package = 'survival')
# ?mgus

mgus_df <- as_tibble(mgus)
dd <- datadist(mgus_df)

mgus_df
```

```
## # A tibble: 241 x 12
##       id   age    sex  dxyr   pcdx pctime futime death   alb creat   hgb
##  * <dbl> <dbl> <fctr> <dbl> <fctr>  <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl>
##  1     1    78 female    68   <NA>     NA    748     1   2.8   1.2  11.5
##  2     2    73 female    66     LP   1310   6751     1    NA    NA    NA
##  3     3    87   male    68   <NA>     NA    277     1   2.2   1.1  11.2
##  4     4    86   male    69   <NA>     NA   1815     1   2.8   1.3  15.3
##  5     5    74 female    68   <NA>     NA   2587     1   3.0   0.8   9.8
##  6     6    81   male    68   <NA>     NA    563     1   2.9   0.9  11.5
##  7     7    72 female    68   <NA>     NA   1135     1   3.0   0.8  13.5
##  8     8    79 female    69   <NA>     NA   2016     1   3.1   0.8  15.5
##  9     9    85 female    70   <NA>     NA   2422     1   3.2   1.0  12.4
## 10    10    58   male    65   <NA>     NA   6155     1   3.5   1.0  14.8
## # ... with 231 more rows, and 1 more variables: mspike <dbl>
```

1. Find number of patient w/ malignancy (AKA transition), death (w/out malignancy) and Free of Events.

```
mgus_df <- mgus_df %>%
  mutate(
    malignancy = !is.na(pcdx)
  )

mgus_df %>%
  group_by(malignancy, death) %>%
  summarise(n = n())
```

```
## # A tibble: 4 x 3
## # Groups:   malignancy [?]
##   malignancy death     n
##        <lgl> <dbl> <int>
## 1      FALSE     0    14
## 2      FALSE     1   163
## 3       TRUE     0     2
## 4       TRUE     1    62
```

Patients w/ malignancy as a FOE are 64; patients which experiment death as FOE are 163, while the ones FoE are 14. 163.

2. Find the indicator for censored, malignancy and death (`indicator`)
3. Find the time-to-event to use in the models (`time_t`)

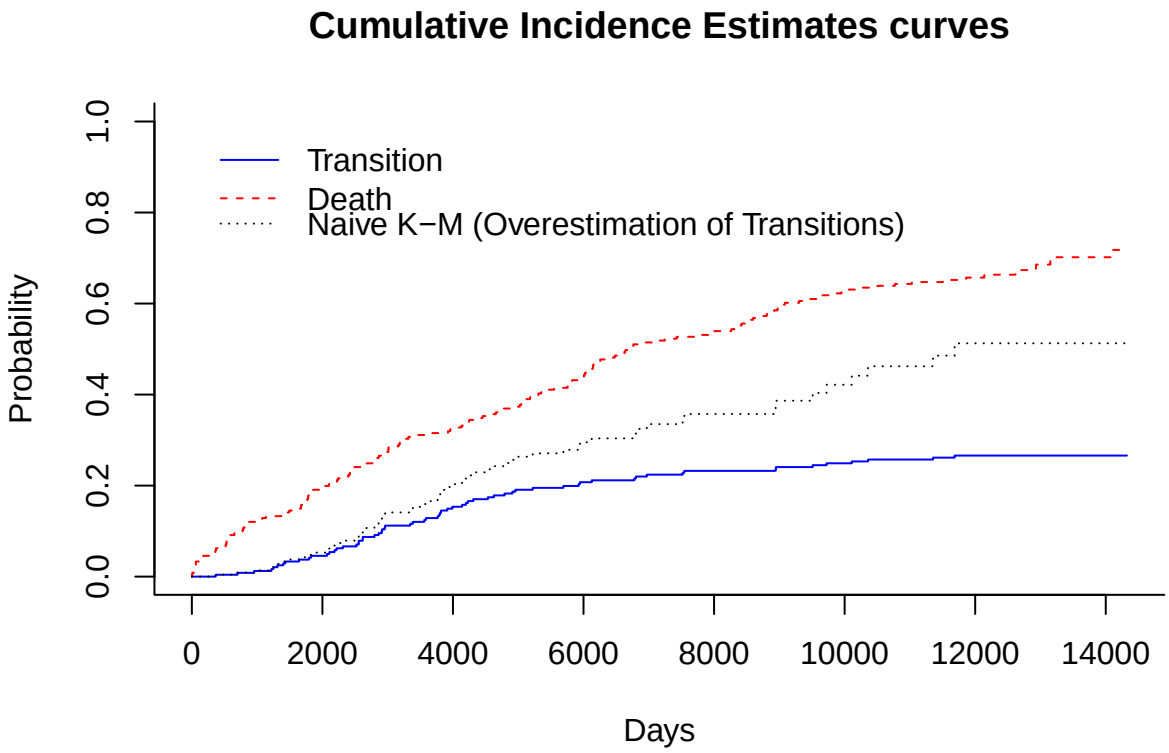
```
mgus_df <- mgus_df %>%
  mutate(
    indicator = if_else(malignancy, 1, 2 * death),
```

```
    time_t    = pmin(futime, pctime, na.rm = TRUE)
)

mgus_df
```

```
## # A tibble: 241 x 15
##       id   age    sex  dxyr   pcdx pctime futime death   alb creat   hgb
##    <dbl> <dbl> <fctr> <dbl> <fctr>  <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl>
##  1     1    78 female    68   <NA>     NA    748     1   2.8   1.2  11.5
##  2     2    73 female    66     LP   1310   6751     1    NA    NA    NA
##  3     3    87   male    68   <NA>     NA    277     1   2.2   1.1  11.2
##  4     4    86   male    69   <NA>     NA   1815     1   2.8   1.3  15.3
##  5     5    74 female    68   <NA>     NA   2587     1   3.0   0.8   9.8
##  6     6    81   male    68   <NA>     NA    563     1   2.9   0.9  11.5
##  7     7    72 female    68   <NA>     NA   1135     1   3.0   0.8  13.5
##  8     8    79 female    69   <NA>     NA   2016     1   3.1   0.8  15.5
##  9     9    85 female    70   <NA>     NA   2422     1   3.2   1.0  12.4
## 10    10    58   male    65   <NA>     NA   6155     1   3.5   1.0  14.8
## # ... with 231 more rows, and 4 more variables: mspike <dbl>,
## #   malignancy <lgl>, indicator <dbl>, time_t <dbl>
```

4. Estimate the naive K-M and the cumulative incidence functions

```
# Using survival
cuminc(mgus_df$time_t, mgus_df$indicator) %>%                    # ?cmprsk::cuminc
  plot(                                                        # ?cmprsk:::plot.cuminc
    main    = 'Cumulative Incidence Estimates curves',
    col     = c('blue', 'red'),
    xlab    = 'Days',
    curvlab = c('Transition', 'Death'),
    wh      = c(1, 1)                                          # legend position
)

survfit(Surv(time_t, malignancy) ~ 1,        # using `rms::npsurv()` is the same
  data = mgus_df
  ) %>%
  lines(                            # Use `lines()` to draw over the previous plot
    fun       = 'event',                               # plot the cumulative events
    conf.int   = FALSE,
    col       = 'black',
    lty       = 3
)
legend(x = 1, y = 0.86,
  legend = 'Naive K-M (Overestimation of Transitions)',
  col    = 'black',
  lty    = 3,
  bty    = 'n' # remove box arround the legend (because we have to add an entry)
)
```




## 3.3 Simulation of Competing risk

1. Specify two cause-specific exponential hazard $\lambda_1(t)$ and $\lambda_2(t)$ of means 0.8 and 1.2. (Set sample size as you like.)

```
n        <- 1e4
lambda_1 <- 0.8
lambda_2 <- 1.2
```

2. Simulate survival times $T$ based on the all causes hazard $\lambda_\cdot(t) = \lambda_1(t) + \lambda_2(t)$.

```
lambda    <- lambda_1 + lambda_2
surv_time <- rexp(n,
  rate = 1 / lambda
)
```

3. Generate Bernoulli $B(p)$ random variables, w/ $p = \lambda_1(t)/\lambda_\cdot(t)$, i.e. is the probability of occurrence of the event of type 1.

```
p_cens     <- lambda_1 / lambda
transition <- rbinom(n,
  size = 1,
  prob = p_cens
) %>%
  as.logical     # Set as logical to use the variable for conditional statements
```

4. Simulate uniform censoring times over $[0, 1]$.

```
censor_time <- runif(n,
  min = 0,
  max = 1
)
```

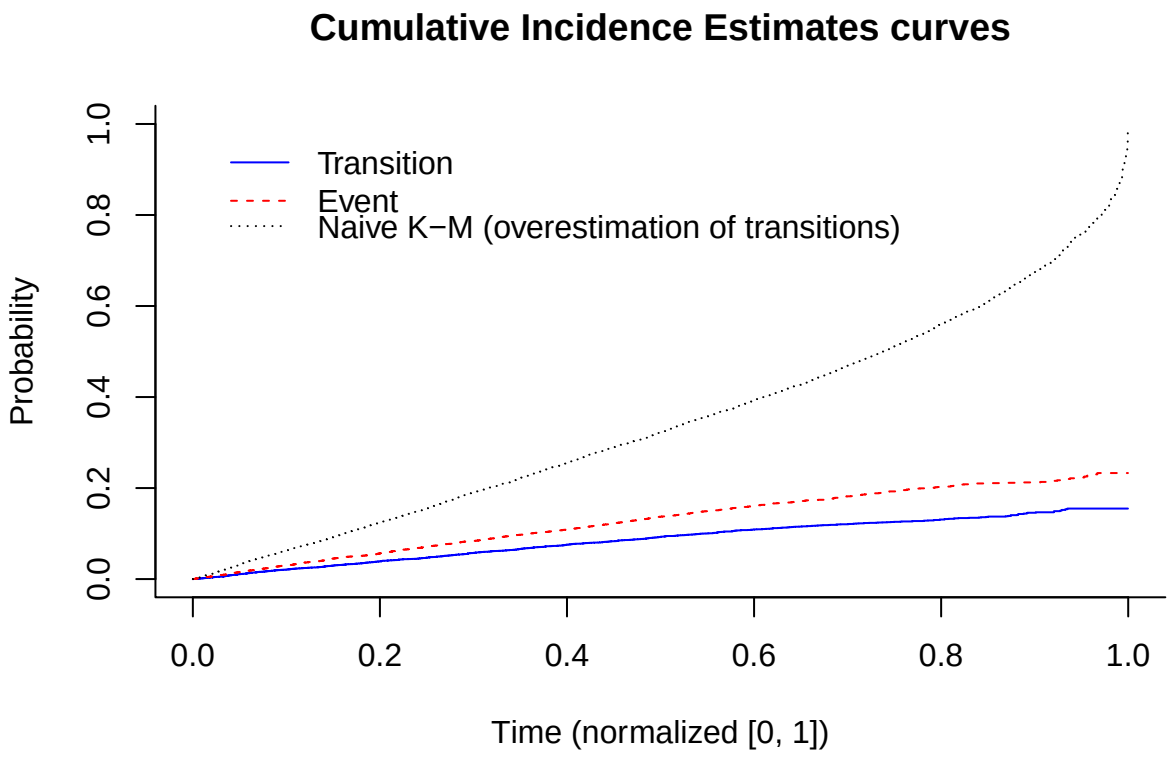
5. Estimate the Cumulative Incidence of each competing event, w/ and w/out censoring; discuss the results.

```
# create the dataset
sim_data <- data_frame(
  id         = seq_len(n),
  transition = transition,
  surv_t     = surv_time,
  cens_t     = censor_time,
  time_t     = pmin(surv_t, cens_t),
  status     = case_when(
    time_t == cens_t ~ 0L,              # All the censored patients has status 0
    transition       ~ 1L,    # Among the other, the ones which has a transition
                              # have state 1
    TRUE             ~ 2L     # All the other were dead (before the end of f-up)
  )
)

# Explore a (random) sample of three cases for each staus
sim_data %>%
  group_by(status) %>%
  sample_n(3)
```

```
## # A tibble: 9 x 6
## # Groups:   status [3]
##      id transition      surv_t    cens_t      time_t status
##   <int>      <lgl>       <dbl>     <dbl>       <dbl>  <int>
## 1  6759      FALSE 6.844765752 0.5175084 0.517508388      0
## 2   906       TRUE 1.378642827 0.5063402 0.506340163      0
## 3  3568       TRUE 0.623047318 0.5752797 0.575279657      0
## 4  2196       TRUE 0.506380841 0.9344626 0.506380841      1
## 5  8119       TRUE 0.157231928 0.9007847 0.157231928      1
## 6  5854       TRUE 0.742379822 0.8771130 0.742379822      1
## 7  1685      FALSE 0.539729742 0.9635529 0.539729742      2
## 8  4550      FALSE 0.001042022 0.3692372 0.001042022      2
## 9  6893      FALSE 0.410360153 0.9322769 0.410360153      2
```

```
# Using survival
cuminc(sim_data$time_t, sim_data$status) %>%                 # ?cmprsk::cuminc
  plot(                                                   # ?cmprsk:::plot.cuminc
    main    = 'Cumulative Incidence Estimates curves',
    col     = c('blue', 'red'),
    xlab    = 'Time (normalized [0, 1])',
    curvlab = c('Transition', 'Event'),
    wh      = c(0.01, 1)                                        # legend position
)

survfit(Surv(time_t, transition) ~ 1,        # using `rms::npsurv()` is the same
  data = sim_data
  ) %>%
```

```
  lines(                          # Use `lines()` to draw over the previous plot
    fun       = 'event',                          # plot the cumulative events
    conf.int   = FALSE,
    col       = 'black',
    lty       = 3
)
legend(x = 0.01, y = 0.86,
  legend = 'Naive K-M (overestimation of transitions)',
  col    = 'black',
  lty    = 3,
  bty    = 'n' # remove box arround the legend (because we have to add an entry)
)
```




## 3.4 Estimation of the effect of sex on MGUS incidence

1. Compare the results of Cox cause specific hazard model…

   For clinical questions, i.e. cause specific risk to experiment the event w/out taking into account the other couse(s)

```
dd <- datadist(mgus_df)

cox_sex <- cph(Surv(time_t, malignancy) ~ sex,
  data = mgus_df,
  x    = TRUE,
  y    = TRUE
```

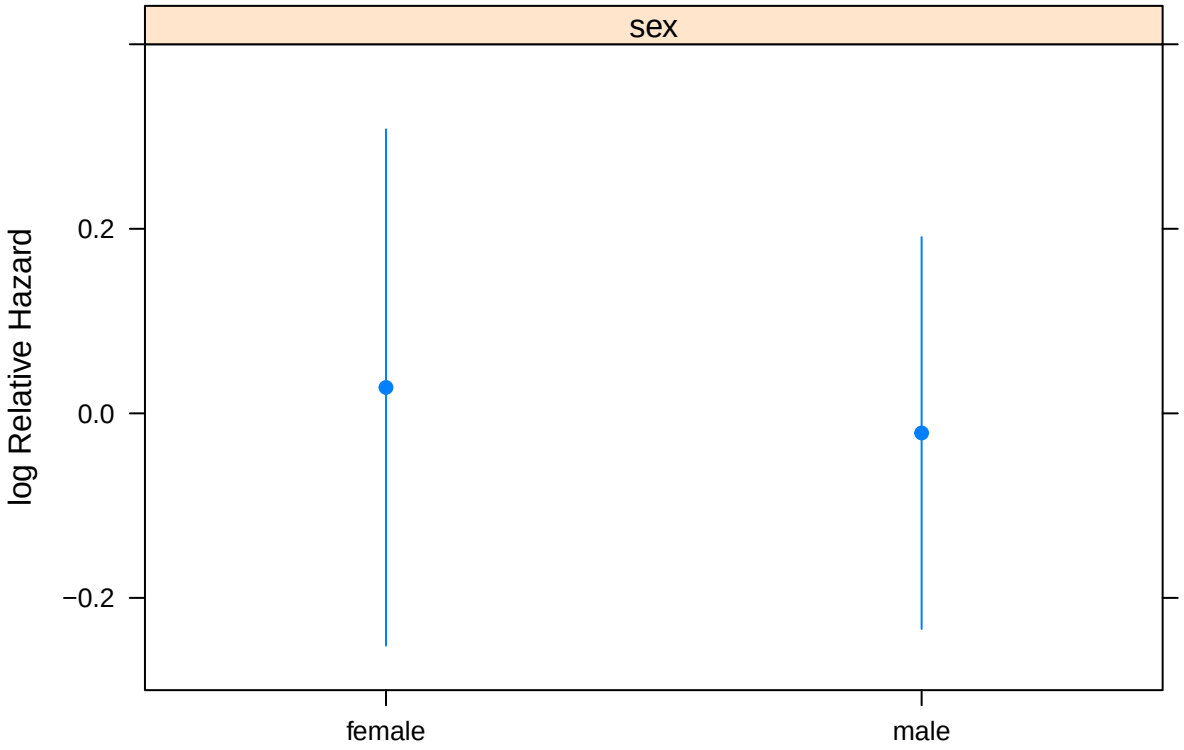
```
)

summary(cox_sex)                    # this is good for a clean view of the effects
```

```
##              Effects              Response : Surv(time_t, malignancy)
##
##  Factor            Low High Diff. Effect   S.E.    Lower 0.95 Upper 0.95
##  sex - female:male 2   1    NA    0.049342 0.25103 -0.44268   0.54136
##   Hazard Ratio     2   1    NA    1.050600      NA  0.64232   1.71830
```

```
cox_sex                     # Here there are more informations (and the p-values)
```

```
## Cox Proportional Hazards Model
##
##  cph(formula = Surv(time_t, malignancy) ~ sex, data = mgus_df,
##      x = TRUE, y = TRUE)
##
##                       Model Tests       Discrimination
##                                            Indexes
##  Obs       241    LR chi2      0.04    R2       0.000
##  Events     64    d.f.            1    Dxy     -0.039
##  Center -0.028    Pr(> chi2) 0.8441    g        0.024
##                   Score chi2   0.04    gr       1.025
##                   Pr(> chi2) 0.8441
##
##           Coef    S.E.   Wald Z Pr(>|Z|)
##  sex=male -0.0493 0.2510 -0.20  0.8442
##
```

```
Predict(cox_sex) %>%      # It is necessary to have the predictions for the plot
  plot
```




```
cox_sex_death <- cph(Surv(time_t, indicator == 2) ~ sex,
  data = mgus_df,
  x    = TRUE,
  y    = TRUE
)

summary(cox_sex_death)
```

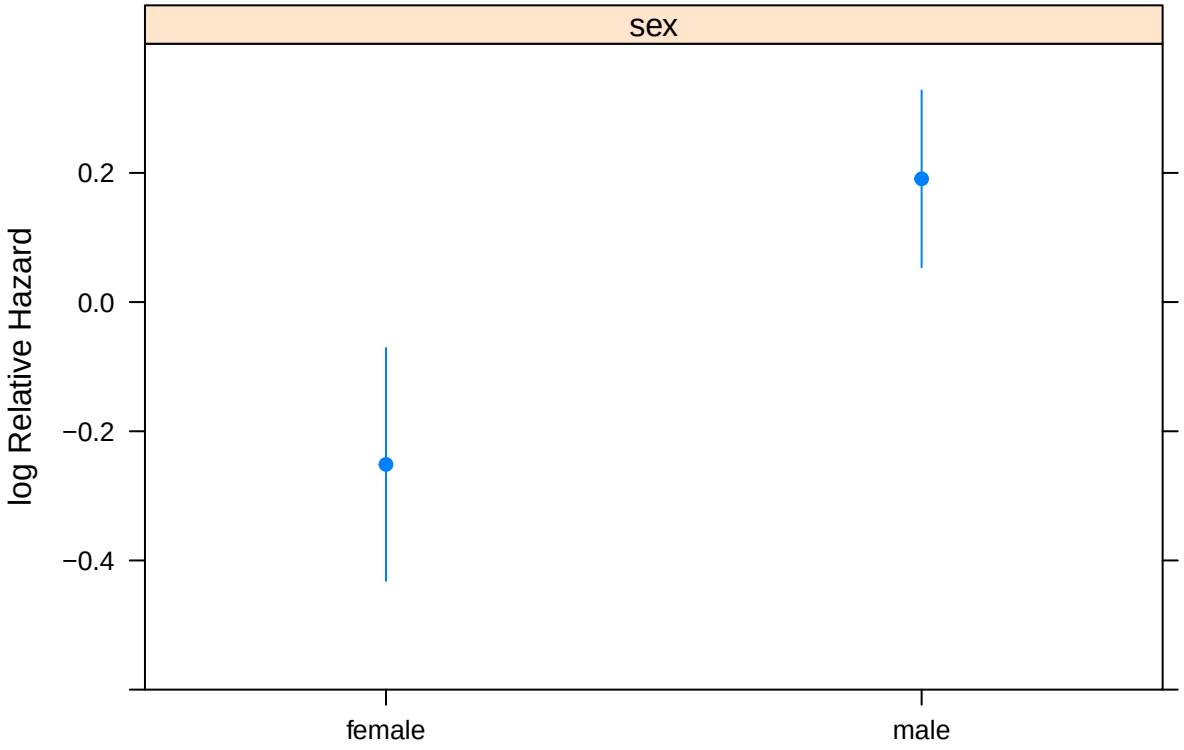
```
##              Effects              Response : Surv(time_t, indicator == 2)
##
##  Factor            Low High Diff. Effect   S.E.    Lower 0.95 Upper 0.95
##  sex - female:male 2   1    NA    -0.44221 0.16183 -0.75939   -0.12502
##   Hazard Ratio     2   1    NA     0.64262      NA  0.46795    0.88248
```

```
cox_sex_death
```

```
## Cox Proportional Hazards Model
##
##  cph(formula = Surv(time_t, indicator == 2) ~ sex, data = mgus_df,
##      x = TRUE, y = TRUE)
##
##                      Model Tests       Discrimination
##                                           Indexes
##  Obs       241    LR chi2      7.65    R2        0.031
##  Events    163    d.f.            1    Dxy       0.124
##  Center 0.2514    Pr(> chi2) 0.0057    g         0.218
##                   Score chi2   7.59    gr        1.243
##                   Pr(> chi2) 0.0059
```

```
##
##            Coef   S.E.   Wald Z Pr(>|Z|)
##  sex=male 0.4422 0.1618 2.73   0.0063
##
```

```
Predict(cox_sex_death) %>%
  plot
```




2. ...to those of the Fine and Gray model

   For administrative questions, i.e. overall risk of experiment each event taking into account the competing risk

```
mgus_num <- mgus_df %>%
  mutate(sex = as.numeric(sex))

crr(                                        # We do not have a plot method for crr
  ftime   = mgus_num$time_t,
  fstatus = mgus_num$indicator,
  cov1    = mgus_num$sex
) %>%
  summary
```

```
## Competing Risks Regression
##
## Call:
## crr(ftime = mgus_num$time_t, fstatus = mgus_num$indicator, cov1 = mgus_num$sex)
##
##                 coef exp(coef) se(coef)     z p-value
```

```
## mgus_num$sex1 -0.339     0.713    0.249 -1.36    0.17
##
##                exp(coef) exp(-coef)  2.5% 97.5%
## mgus_num$sex1     0.713        1.4 0.437  1.16
##
## Num. cases = 241
## Pseudo Log-likelihood = -341
## Pseudo likelihood ratio test = 1.83  on 1 df,
```

# Software

## Packages

All the exercise are solved using R (ver. 3.4.2) has been used provided with packages: `survival` (Therneau (2017)) for the survival data analyses (reference package), `survminer` (Kassambara and Kosinski (2017)) for advance survival plot using `ggplot2` (Wickham and Chang (2016)) package, `cmprsk` (Gray (2014)) for competing risk, `rms` (Harrell, Jr. (2017)) for additional features on regression modeling strategies (survival ones included).

With regards to the data management, the collection of package `tidyverse` (Wickham (2017)) is loaded, which includes: `dplyr` (Wickham et al. (2017)) for data manipulation, `purrr` (Henry and Wickham (2017)) for functional programming, `readr` (R-readr) for data import, `tidyr` (R-tidyr) for funtions to tidy the data, `tibble` (R-tibble) to take advantage of the *tible* data frame class and `ggplot2` as a interface for the Gramar of Grahics.

The present book was written in RMarkdown (R-rmarkdown), compiled using `knitr` (Xie (2017b)) and rendered as an HTML book by `bookdown` (Xie (2017a)).

## System Information

All the code is compiled on a system with the following overall characteristics and loaded packages.

```
devtools::session_info()
```

```
##  setting  value
##  version  R version 3.4.2 (2017-09-28)
##  system   x86_64, mingw32
##  ui       RTerm
##  language (EN)
##  collate  English_United States.1252
##  tz       Europe/Berlin
##  date     2017-10-05
##
##  package      * version date       source
##  acepack        1.4.1   2016-10-29 CRAN (R 3.4.1)
##  assertthat     0.2.0   2017-04-11 CRAN (R 3.4.1)
##  backports      1.1.0   2017-05-22 CRAN (R 3.4.0)
##  base         * 3.4.2   2017-09-28 local
##  base64enc      0.1-3   2015-07-28 CRAN (R 3.4.0)
##  bindr          0.1     2016-11-13 CRAN (R 3.4.1)
##  bindrcpp     * 0.2     2017-06-17 CRAN (R 3.4.1)
##  bookdown       0.5     2017-08-20 CRAN (R 3.4.1)
```

```
##  broom          0.4.2   2017-02-13 CRAN (R 3.4.0)
##  cellranger     1.1.0   2016-07-27 CRAN (R 3.4.1)
##  checkmate      1.8.3   2017-07-03 CRAN (R 3.4.1)
##  cluster        2.0.6   2017-03-16 CRAN (R 3.4.1)
##  cmprsk       * 2.2-7   2014-06-17 CRAN (R 3.4.1)
##  codetools      0.2-15  2016-10-05 CRAN (R 3.4.0)
##  colorspace     1.3-2   2016-12-14 CRAN (R 3.4.1)
##  compiler       3.4.2   2017-09-28 local
##  data.table     1.10.4  2017-02-01 CRAN (R 3.4.0)
##  datasets     * 3.4.2   2017-09-28 local
##  devtools       1.13.3  2017-08-02 CRAN (R 3.4.1)
##  digest         0.6.12  2017-01-27 CRAN (R 3.4.1)
##  dplyr        * 0.7.3   2017-09-09 CRAN (R 3.4.1)
##  evaluate       0.10.1  2017-06-24 CRAN (R 3.4.1)
##  forcats        0.2.0   2017-01-23 CRAN (R 3.4.1)
##  foreign        0.8-69  2017-06-21 CRAN (R 3.4.0)
##  Formula      * 1.2-2   2017-07-10 CRAN (R 3.4.1)
##  ggplot2      * 2.2.1   2016-12-30 CRAN (R 3.4.1)
##  ggpubr       * 0.1.5   2017-08-22 CRAN (R 3.4.1)
##  glue           1.1.1   2017-06-21 CRAN (R 3.4.1)
##  graphics     * 3.4.2   2017-09-28 local
##  grDevices    * 3.4.2   2017-09-28 local
##  grid           3.4.2   2017-09-28 local
##  gridExtra      2.3     2017-09-09 CRAN (R 3.4.1)
##  gtable         0.2.0   2016-02-26 CRAN (R 3.4.1)
##  haven          1.1.0   2017-07-09 CRAN (R 3.4.1)
##  Hmisc        * 4.0-3   2017-05-02 CRAN (R 3.4.1)
##  hms            0.3     2016-11-22 CRAN (R 3.4.1)
##  htmlTable      1.9     2017-01-26 CRAN (R 3.4.1)
##  htmltools      0.3.6   2017-04-28 CRAN (R 3.4.1)
##  htmlwidgets    0.9     2017-07-10 CRAN (R 3.4.1)
##  httr           1.3.1   2017-08-20 CRAN (R 3.4.1)
##  jsonlite       1.5     2017-06-01 CRAN (R 3.4.1)
##  km.ci          0.5-2   2009-08-30 CRAN (R 3.4.1)
##  KMsurv         0.1-5   2012-12-03 CRAN (R 3.4.0)
##  knitr          1.17    2017-08-10 CRAN (R 3.4.1)
##  labeling       0.3     2014-08-23 CRAN (R 3.4.0)
##  lattice      * 0.20-35 2017-03-25 CRAN (R 3.4.1)
##  latticeExtra   0.6-28  2016-02-09 CRAN (R 3.4.1)
##  lazyeval       0.2.0   2016-06-12 CRAN (R 3.4.1)
##  lubridate      1.6.0   2016-09-13 CRAN (R 3.4.1)
##  magrittr     * 1.5     2014-11-22 CRAN (R 3.4.1)
##  MASS           7.3-47  2017-04-21 CRAN (R 3.4.1)
##  Matrix         1.2-11  2017-08-16 CRAN (R 3.4.1)
##  MatrixModels   0.4-1   2015-08-22 CRAN (R 3.4.1)
##  memoise        1.1.0   2017-04-21 CRAN (R 3.4.1)
##  methods      * 3.4.2   2017-09-28 local
##  mnormt         1.5-5   2016-10-15 CRAN (R 3.4.0)
##  modelr         0.1.1   2017-07-24 CRAN (R 3.4.1)
##  multcomp       1.4-7   2017-09-07 CRAN (R 3.4.1)
##  munsell        0.4.3   2016-02-13 CRAN (R 3.4.1)
##  mvtnorm        1.0-6   2017-03-02 CRAN (R 3.4.0)
##  nlme           3.1-131 2017-02-06 CRAN (R 3.4.1)
##  nnet           7.3-12  2016-02-02 CRAN (R 3.4.1)
```

```
##  parallel       3.4.2   2017-09-28 local
##  pkgconfig      2.0.1   2017-03-21 CRAN (R 3.4.1)
##  plyr           1.8.4   2016-06-08 CRAN (R 3.4.1)
##  polspline      1.1.12  2015-07-14 CRAN (R 3.4.0)
##  psych          1.7.8   2017-09-09 CRAN (R 3.4.1)
##  purrr        * 0.2.3   2017-08-02 CRAN (R 3.4.1)
##  quantreg       5.33    2017-04-18 CRAN (R 3.4.1)
##  R6             2.2.2   2017-06-17 CRAN (R 3.4.1)
##  RColorBrewer   1.1-2   2014-12-07 CRAN (R 3.4.0)
##  Rcpp           0.12.12 2017-07-15 CRAN (R 3.4.1)
##  readr        * 1.1.1   2017-05-16 CRAN (R 3.4.1)
##  readxl         1.0.0   2017-04-18 CRAN (R 3.4.1)
##  reshape2       1.4.2   2016-10-22 CRAN (R 3.4.1)
##  rlang          0.1.2   2017-08-09 CRAN (R 3.4.1)
##  rmarkdown      1.6     2017-06-15 CRAN (R 3.4.1)
##  rms          * 5.1-1   2017-05-03 CRAN (R 3.4.1)
##  rpart          4.1-11  2017-04-21 CRAN (R 3.4.1)
##  rprojroot      1.2     2017-01-16 CRAN (R 3.4.1)
##  rstudioapi     0.7     2017-09-07 CRAN (R 3.4.1)
##  rvest          0.3.2   2016-06-17 CRAN (R 3.4.1)
##  sandwich       2.4-0   2017-07-26 CRAN (R 3.4.1)
##  scales         0.5.0   2017-08-24 CRAN (R 3.4.1)
##  SparseM      * 1.77    2017-04-23 CRAN (R 3.4.0)
##  splines        3.4.2   2017-09-28 local
##  stats        * 3.4.2   2017-09-28 local
##  stringi        1.1.5   2017-04-07 CRAN (R 3.4.0)
##  stringr        1.2.0   2017-02-18 CRAN (R 3.4.1)
##  survival     * 2.41-3  2017-04-04 CRAN (R 3.4.1)
##  survminer    * 0.4.0   2017-06-07 CRAN (R 3.4.1)
##  survMisc       0.5.4   2016-11-23 CRAN (R 3.4.1)
##  TH.data        1.0-8   2017-01-23 CRAN (R 3.4.1)
##  tibble       * 1.3.4   2017-08-22 CRAN (R 3.4.1)
##  tidyr        * 0.7.1   2017-09-01 CRAN (R 3.4.1)
##  tidyverse    * 1.1.1   2017-01-27 CRAN (R 3.4.1)
##  tools          3.4.2   2017-09-28 local
##  utils        * 3.4.2   2017-09-28 local
##  withr          2.0.0   2017-07-28 CRAN (R 3.4.1)
##  xml2           1.1.1   2017-01-24 CRAN (R 3.4.1)
##  xtable         1.8-2   2016-02-05 CRAN (R 3.4.1)
##  yaml           2.1.14  2016-11-12 CRAN (R 3.4.1)
##  zoo            1.8-0   2017-04-12 CRAN (R 3.4.1)
```

# Bibliography


Gray, B. (2014). *cmprsk: Subdistribution Analysis of Competing Risks*. R package version 2.2-7.

Gray, R. J. (1988). A class of k-sample tests for comparing the cumulative incidence of a competing risk. *The Annals of statistics*, pages 1141–1154.

Harrell Jr, F. E. (2015). *Regression modeling strategies: with applications to linear models, logistic and ordinal regression, and survival analysis*. Springer.

Harrell, Jr., F. E. (2017). *rms: Regression Modeling Strategies*. R package version 5.1-1.

Henry, L. and Wickham, H. (2017). *purrr: Functional Programming Tools*. R package version 0.2.3.

Kassambara, A. and Kosinski, M. (2017). *survminer: Drawing Survival Curves using 'ggplot2'*. R package version 0.4.0.

Therneau, T. M. (2017). *survival: Survival Analysis*. R package version 2.41-3.

Wickham, H. (2017). *tidyverse: Easily Install and Load 'Tidyverse' Packages*. R package version 1.1.1.

Wickham, H. and Chang, W. (2016). *ggplot2: Create Elegant Data Visualisations Using the Grammar of Graphics*. R package version 2.2.1.

Wickham, H., Francois, R., Henry, L., and Müller, K. (2017). *dplyr: A Grammar of Data Manipulation*. R package version 0.7.3.

Xie, Y. (2017a). *bookdown: Authoring Books and Technical Documents with R Markdown*. R package version 0.5.

Xie, Y. (2017b). *knitr: A General-Purpose Package for Dynamic Report Generation in R*. R package version 1.17.