Time-to-Event Analysis: Practical 1

Instructor: Graeme L. Hickey

27th October 2014

Contents

1 Introduction

At this point you should now be familiar with the key foundational concepts in time-to-event analysis. The purpose of this practical session is to learn how to apply those concepts to real data. We will do this using the R language and environment for statistical computing and graphics (http://www.r-project.org). You should already be familiar with R; however if you have any problems, do not hesitate to ask.

2 Getting started

2.1 Materials

The slides from the lecture, a PDF version of this exercise sheet, and relevant data files are all available on the course USB flash drive. It is recommended to have a digital version of this exercise sheet open on your laptop as it will allow you to copy-and-paste code into the R console. Note that all the code needed for the practical will be self-contained in these exercise sheets.

2.2 R `survival` package

In this practical we will be using the R package `survival`. This package was developed (and is still maintained by) Terry Therneau (Mayo Clinic). It will therefore not be a surprise that the book by Therneau and Grambsch (2000) is an excellent source of reference for survival analysis in R. The package comes pre-installed with R, meaning that you do not have to download it separately. However, you might wish to check if you are running the most recent version by running the R command `update.packages()`.

To load the survival package, we run

```
library(survival)
```

```
## Loading required package: splines
```

2.3 Data

R has many datasets already built-in. To check what datasets we have available, run the command `data()`. However, not all of these datasets are time-to-event data. To use a dataset from R's library, just type the name of it. You can also look up details of the data by using `help()` function, e.g.

```
aml
```

```
##   time status          x
## 1    9      1 Maintained
## 2   13      1 Maintained
## 3   13      0 Maintained
## 4   18      1 Maintained
## 5   23      1 Maintained
## 6   28      0 Maintained
```

```
help("aml")
```

For datasets saved on your hard drive or USB flash drive, we can use the `read.table()` or `read.csv()`, e.g.

```
my.data <- read.table(file.choose(), header = TRUE)
```

which will bring up a pop-up selection box where you can navigate to the data file and open it.

Once you have read your data into R you should first check that it has interpreted the file correctly. The simplest ways to do this are to apply the `head()`, `summary()` and `str()` functions to your data, e.g.

```
head(my.data)
summary(my.data)
str(my.data)
```

2.4 Further information about a function

If you find yourself stuck, or want to understand a particular function in more detail, remember you can use the `help()` function. For example, running `help("plot")` will bring up a pop-up screen describing the the `plot()` function. Most of the functions shown here have further optional arguments that can be used to perform more complex analyses, for example interval-censored data, repeated events, etc.

3 Example 1: Introductory concepts

3.1 The `aml` data

Acute myelogenous leukemia (AML) is a cancer of the myeloid line of blood cells. A clinical trial to evaluate the efficacy of maintenance chemotherapy for acute myelogenous leukaemia was conducted by Embury et al. (1977) at Stanford University. After reaching a stage of remission through treatment by chemotherapy, patients were randomized into two groups. The first group received maintenance chemotherapy and the second group did not. The aim of the study was to see if maintenance chemotherapy increased the length of the remission. The data here formed a preliminary analysis which was conducted in October 1974.

The AML data is stored in R as `aml`, and is available after loading the `survival` package.[1] We begin by inspecting it:

```
head(aml)
```

```
##   time status          x
## 1    9      1 Maintained
## 2   13      1 Maintained
## 3   13      0 Maintained
## 4   18      1 Maintained
## 5   23      1 Maintained
## 6   28      0 Maintained
```

```
summary(aml)
```

```
##       time             status                   x
##  Min.   :  5.00   Min.   :0.0000   Maintained   :11
##  1st Qu.: 12.50   1st Qu.:1.0000   Nonmaintained:12
##  Median : 23.00   Median :1.0000
##  Mean   : 29.48   Mean   :0.7826
##  3rd Qu.: 33.50   3rd Qu.:1.0000
##  Max.   :161.00   Max.   :1.0000
```

[1] The description of the dataset provided here is more comprehensive than that given in the `survival` package as the identical dataset is stored in the `boot` package with additional background information.

We find it has 3 columns:

- `time`: The length of the complete remission (in weeks).
- `status`: An indicator of right censoring. 1 indicates that the patient had a relapse and so time is the length of the remission. 0 indicates that the patient had left the study or was still in remission in October 1974, that is the length of remission is right-censored.
- `x`: The group into which the patient was randomized.

3.2 Creating survival objects

We need to be able to tell R that this is survival data. To do this we will use the `Surv()` function. To understand what this function does, run:

```
Surv(aml$time, aml$status)
```

```
##  [1]   9   13   13+  18   23   28+  31   34   45+  48  161+   5    5    8
## [15]   8   12   16+  23   27   30   33   43   45
```

Numbers with a `+` postfix indicates right censoring, whereas all other numbers indicate the observed time to relapse.

3.3 Kaplan-Meier survival estimates

Before we can plot any Kaplan-Meier curves, we must first calculate the estimates. To do this we use the `survfit()` function, and save this object (below I call it `aml.km`).

```
aml.km <- survfit(Surv(time, status) ~ x, data = aml)
```

If we replaced the `~ x` with `~ 1` in the function above, then it would tell R to only estimate a single Kaplan-Meier curve. The current syntax is telling R to estimate **separate** Kaplan-Meier curves for each treatment group (referred to as strata, or in R language as `levels`).

We can get obtain descriptive statistics from examining `aml.km`.

```
aml.km
```

```
## Call: survfit(formula = Surv(time, status) ~ x, data = aml)
##
##                 records n.max n.start events median 0.95LCL 0.95UCL
## x=Maintained         11    11      11      7     31      18      NA
## x=Nonmaintained      12    12      12     11     23       8      NA
```

This tells us that there were 12 non-maintained and 11 maintained patients at the start of the study. A total of 7 and 11 patients respectively had observed clinical relapses during the follow-up period.

Recall that the Kaplan-Meier estimator is evaluated only at observed relapse times; censored values only contribute to the number of patients at risk. To get the equivalent tables that we derived by hand in lecture, we use the `summary()` function applied

```
summary(aml.km)
```

```
## Call: survfit(formula = Surv(time, status) ~ x, data = aml)
##
##                 x=Maintained
##  time n.risk n.event survival std.err lower 95% CI upper 95% CI
##     9     11       1    0.909  0.0867       0.7541        1.000
##    13     10       1    0.818  0.1163       0.6192        1.000
##    18      8       1    0.716  0.1397       0.4884        1.000
##    23      7       1    0.614  0.1526       0.3769        0.999
##    31      5       1    0.491  0.1642       0.2549        0.946
##    34      4       1    0.368  0.1627       0.1549        0.875
##    48      2       1    0.184  0.1535       0.0359        0.944
##
##                 x=Nonmaintained
##  time n.risk n.event survival std.err lower 95% CI upper 95% CI
##     5     12       2   0.8333  0.1076       0.6470        1.000
##     8     10       2   0.6667  0.1361       0.4468        0.995
##    12      8       1   0.5833  0.1423       0.3616        0.941
##    23      6       1   0.4861  0.1481       0.2675        0.883
##    27      5       1   0.3889  0.1470       0.1854        0.816
##    30      4       1   0.2917  0.1387       0.1148        0.741
##    33      3       1   0.1944  0.1219       0.0569        0.664
##    43      2       1   0.0972  0.0919       0.0153        0.620
##    45      1       1   0.0000     NaN           NA           NA
```

For each treatment group we get a similar table to what we derived earlier. R also gives an estimate of the standard error and a pointwise 95% confidence interval.[2] For example, in the maintained group at time 13 weeks, 1 patient relapsed. Just before this, there were 10 patients at risk. At the next failure time, $t = 18$ weeks, there were only 8 patients at risk — this means one patient must have been censored in the interval $(8, 13)$ weeks.

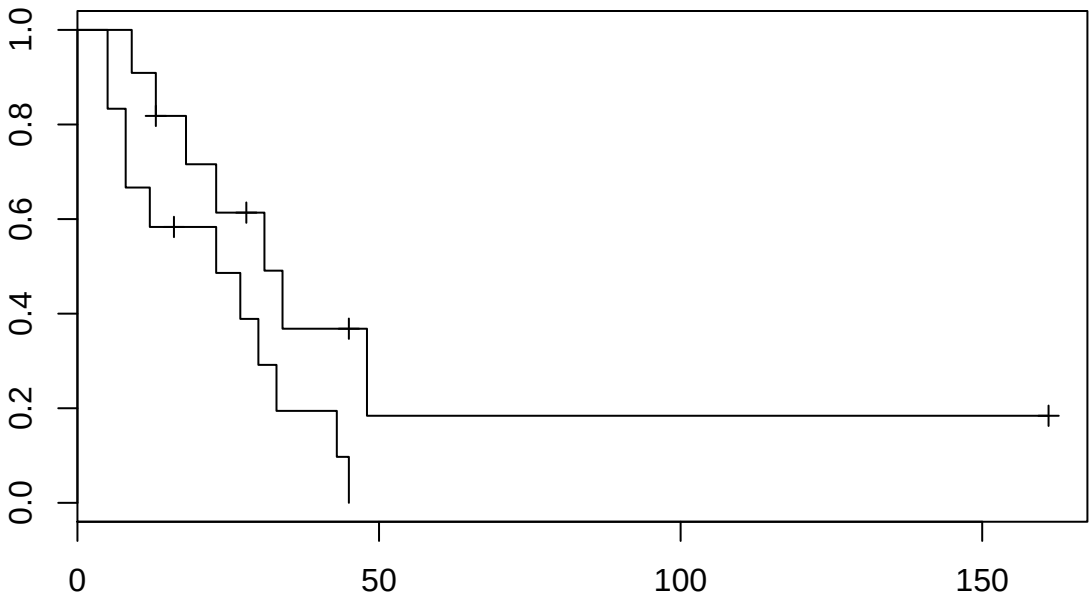
When we typed in `aml.km` we were also given a column with `median` in it. These are the median times to event, equivalent to drawing a horizontal line at $S(t) = 0.5$ on the Kaplan-Meier graph and identifying the x-axis coordinates (time) where it crosses the curves.

[2] We did not cover these in class, but R uses the Delta-method applied to the Kaplan-Meier estimator, which has since become known as Greenwood's formula.

3.4 Plotting the Kaplan-Meier curves

To plot the Kaplan-Meier curves we apply the `plot()` function

```
plot(aml.km)
```



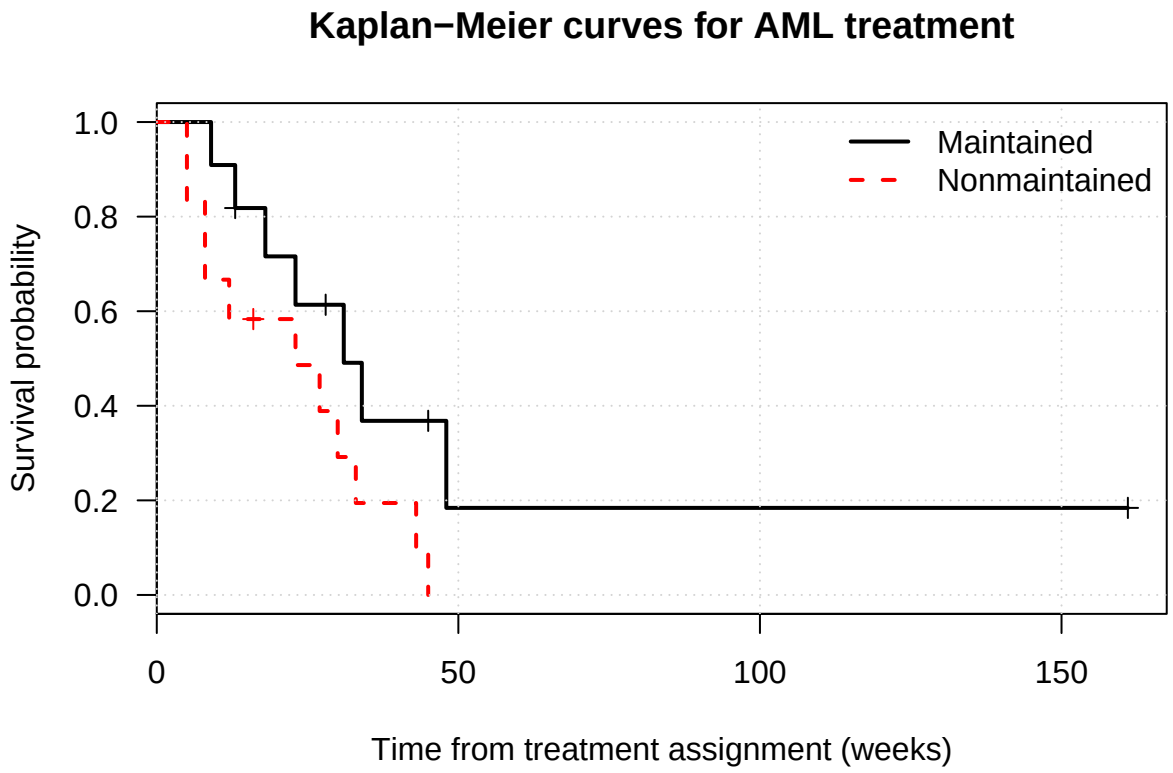
This plot is not particularly useful: we need to exploit the optional plotting parameters to optimize the graph for presentation. We should also add a legend, and perhaps some gridlines. You might like to look up the plotting options in the `par` and `plot.survfit` functions. In particular, I use the following arguments:

- `col`: specifies the line colours, and can take numeric arguments or names of colours
- `lty`: specifies the line type, e.g. solid, dashed, dotted
- `lwd`: specifies the line width; the larger the value the thicker it is
- `xlab` and `ylab`: specifies the axes labels
- `main`: specifies a title
- `las`: specifies the orientation of the axes values

```
plot(aml.km,
     col = 1:2,
     lty = 1:2,
     lwd = c(2, 2),
     xlab = "Time from treatment assignment (weeks)",
     ylab = "Survival probability",
     main = "Kaplan-Meier curves for AML treatment",
     las = 1
     )
```

```
grid() # Add some gridlines

legend("topright",
       levels(aml$x), # Run levels(aml$x) to see
       col = 1:2,     # what this does
       lty = 1:2,
       lwd = c(2, 2),
       bty = "n"
       )
```



Note also that the **tick lines** on the survival curves denote the censoring times in the data.

3.5 The log-rank test

The purpose of the AML clinical trial was to identify if there was a difference in prognosis between the two treatment protocols. We can use the log-rank test to check if the data supports the hypothesis that the failure times are from the same distribution. In R we use the `survdiff()` function using the same arguments as for the `survfit()` function.

```
survdiff(Surv(time, status) ~ x, data = aml)
```

```
## Call:
## survdiff(formula = Surv(time, status) ~ x, data = aml)
```

```
##
##                   N Observed Expected (O-E)^2/E (O-E)^2/V
## x=Maintained     11        7    10.69      1.27       3.4
## x=Nonmaintained 12       11     7.31      1.86       3.4
##
##  Chisq= 3.4  on 1 degrees of freedom, p= 0.0653
```

We see that the chi square test statistic, aggregated over all 18 times is 3.396, which on 1 degree of freedom, we find the P-value is

$$P(\chi_1^2 > 3.396) = 0.0653$$

As this is > 0.05, we would not say it is statistically significant; however there is moderate evidence of a difference in survival.

4 Problem 1: Introductory concepts

4.1 The `dialysis` data

The `dialysis` dataset was briefly shown during the lecture. It records the survival times of 124 renal patients from a hospital in Wrexham given renal dialysis by each of two methods, automated peritoneal dialysis (APD) and continuous ambulatory peritoneal dialysis (CAPD). This dataset was presented by Diggle and Chetwynd (2011; §8); see book for further details. The file is saved as `dialysis.data` on the USB flash drive.

The dataset contains 5 variables:

- `id`: patient identifier
- `days`: survival time (days since start of treatment, may be censored)
- `dead`: censoring indicator: if 0, patient was still alive after number of days indicated in `days` column; if 1, patient died after number of days indicated in `days` column
- `method`: dialysis method: 0 for CAPD (Continuous Ambulatory Peritoneal Dialysis), 1 for APD (Automated Peritoneal Dialysis)
- `age`: age (in years at start of treatment)

4.2 Excercises

Once familiar with the dataset and its description, attempt the exercises below. It is recommended you spend no more than 45 minutes on these exercises. There are some pointers below if you get stuck.

1. Calculate some summary statistics of the dataset and the survival data.

2. Calculate an estimate of the survival distribution functions for APD and CAPD using the Kaplan-Meier method.

3. Plot the Kaplan-Meier curves, making sure they are suitably polished for a research report. Do the data suggest any difference in the short-term or long-term survival probability according to the method of dialysis?

4. The research question of interest concerns whether there is any significant difference in the survival of patients receiving dialysis by each of these two methods. Using a suitable statistical test, answer the research question of interest.

5. The renal specialists also recorded the age of the patients, which is in the `dialysis` dataset. Is there any evidence to support a hypothesis of treatment selection bias on the basis of patient age?

6. Ignoring the treatment modality for now, is patient age at treatment allocation associated with survival time?

7. Add 95% confidence intervals to the Kaplan-Meier plot produced in part (3) and interpret.

4.3 Pointers

- **Boxplots** of a continuous variable `y` stratified by grouping variable `grp` can be shown using `boxplot(y ~ grp, data = my.dat)`.
- The **Student *t*-test** and the **Wilcoxon-Mann-Whitney *U*-test** are available in R under the functions `t.test()` and `wilcox.test()`.
- If we wanted to create and append a **new variable** to a `data.frame` we can use the `$` operator. E.g. `my.dat$x50 <- (my.dat$x > 50)` creates a new variable called `x50` that takes value `TRUE` (equivalently `1`) if `x` is > 50, or `FALSE` (equivalently `0`) if `x` is ≤ 50.
- When applying the `plot()` function to a `survfit` object, it is actually calling the function `plot.survfit()`. You can look at help file for this function to learn of additional **plotting arguments**, including for example, `conf.int`, which controls whether **confidence intervals** are plotted or not by setting `= TRUE` or `= FALSE` respectively. These are calculated using the Greenwood formula, which was briefly mentioned above.

5 Example 2: Cumulative hazard functions

During the lecture we saw the Nelson-Aalen estimator for the **cumulative hazard** function. Furthermore we noted that we could estimate the survival function by exploiting the relation between $S(t)$ and $H(t)$, namely

$$S(t) = \exp\left[-H(t)\right]$$

which gives the so-called Fleming-Harrington survival estimator.

5.1 Calculating the Nelson-Aalen estimator by hand

Let us calculate the Nelson-Aalen estimate of cumulative hazard for the non-maintained patients in the `aml` data.

We begin by extracting a subset of the dataset and then applying the `survfit()` function to it, using `~ 1` in the formula as there are no longer any stratification variables.

```
aml0 <- aml[aml$x == "Nonmaintained", ]
fit.aml0 <- survfit(Surv(time, status) ~ 1, data = aml0)
```

Contained inside the `fit.aml0` object are the:

- times (denoted t_i in lecture[3]) to either an event or [right] censoring time
- number of patients at risk just before the time (denoted n_i in lecture)
- number of events that occurred at the time (denoted d_i in lecture)

We extract these items and save them:

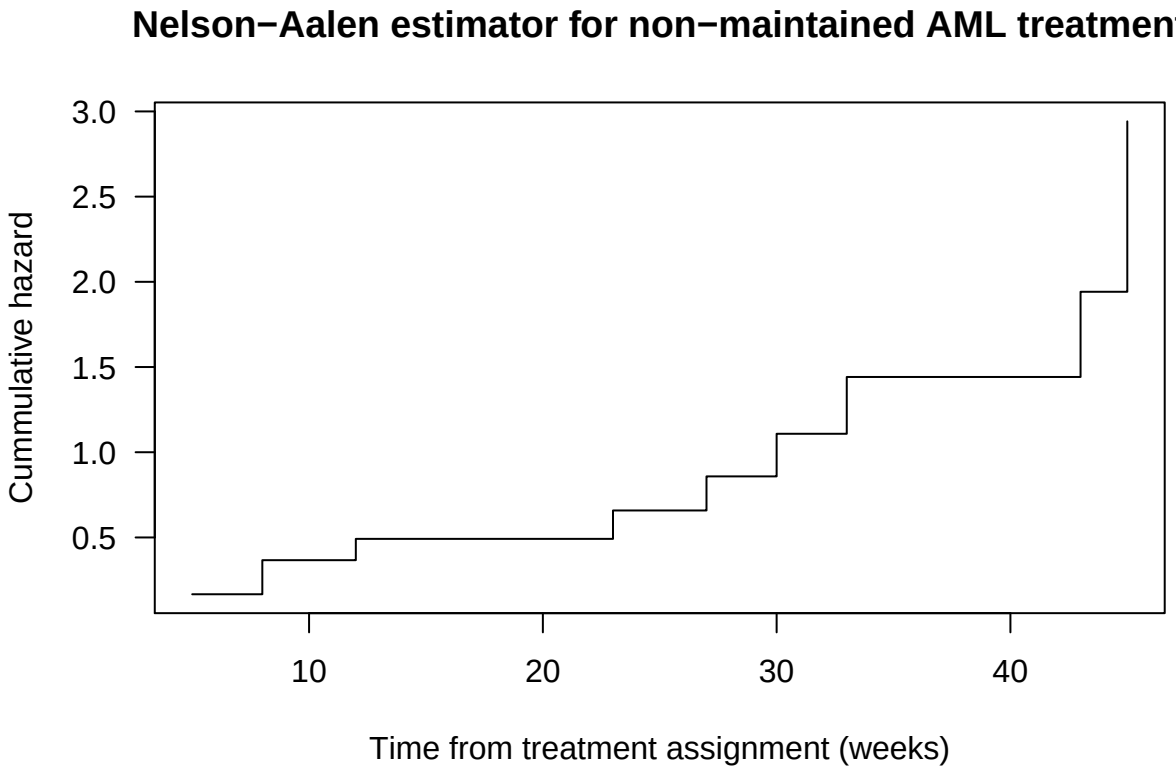
```
ti <- fit.aml0$time
ni <- fit.aml0$n.risk
di <- fit.aml0$n.event
```

Finally, we estimate the **cumulative** hazard using the `cumsum()` function.[4]

```
H <- cumsum(di / ni)
plot(ti, H,
     type = "s", # This produces a "staircase" plot
     xlab = "Time from treatment assignment (weeks)",
     ylab = "Cummulative hazard",
     main = "Nelson-Aalen estimator for non-maintained AML treatment",
     las = 1)
```

[3] In the lecture we denoted $t_1, t_2, \ldots, t_N$ to be the failure times only. However, now we include all times, including the right-censored times, for reasons that will become apparent shortly.

[4] `cumsum` is short for cumulative sum



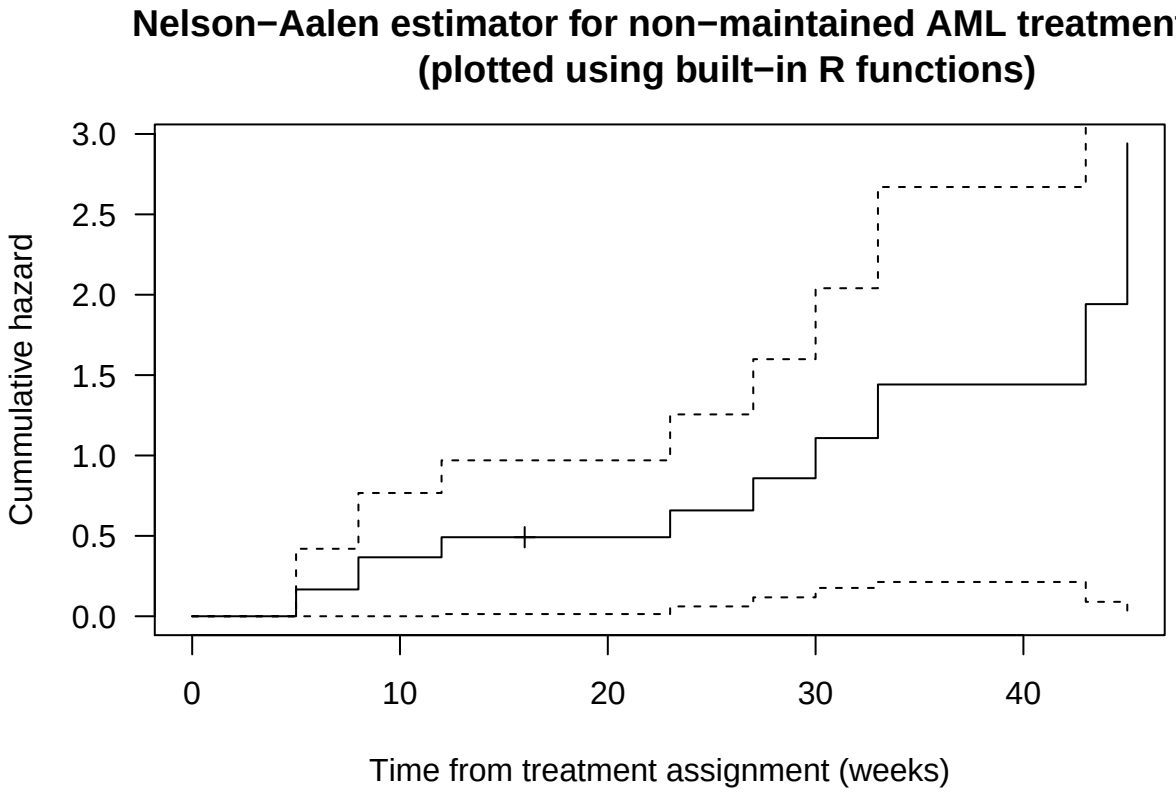
Note: although we wrote the Nelson-Aalen estimator in terms of event times only, it doesn't matter that we have included censoring times in the above calculation because $d_i = 0$ for these points, hence the terms $d_i/n_i = 0$ will not contribute anything to the cumulative sum.

5.2 Simpler method

As expected, R has a built in function for plotting non-parametric estimates of cumulative hazard function. There are two steps to plot the Nelson-Aalen estimator in R:

1. use `survfit` to fit the survival data using the Fleming-Harrington method
2. apply the `cumhaz` function to the saved survival fit object

```
fit.fh <- survfit(Surv(time, status) ~ 1, data = aml0,
                  type = "fleming")
plot(fit.fh, fun = "cumhaz",
     xlab = "Time from treatment assignment (weeks)",
     ylab = "Cummulative hazard",
     main = "Nelson-Aalen estimator for non-maintained AML treatment
             (plotted using built-in R functions)",
     las = 1)
```



6 Problem 2: Cumulative hazard functions

6.1 The `calf_pneu` data

The `calf_pneu` data records the time to occurrence of calf pneumonia in calves raised in two different housing systems: continuous and batch. Calves surviving to 150 days without pneumonia are considered censored at that time. This dataset was presented by Dohoo et al. (2009; §19); see the book for further details. The file is saved as `calf_pneu.data` on the USB flash drive.

The dataset contains 4 variables:

- `calf`: a calf identifier
- `stock`: housing system: 1 for continuous, 0 for batch
- `days`: survival time (days since birth, may be censored)
- `pn`: censoring indicator: if 0, calf was negative for pneumonia after number of days indicated in `days` column; if 1, calf contracted pneumonia after number of days indicated in `days` column

6.2 Exercises

Once familiar with the dataset and its description, attempt the exercises below. It is recommended you spend no more than 45 minutes on these exercises. There are some pointers below if you get stuck.

1. Estimate the Kaplan-Meier survival functions for continuous and batch housing and plot them. Do you think, based only on this plot, that there is a difference in survivorship? And if so, in what direction? (It might help to add the confidence intervals when evaluating the graphs.)
2. Using a suitable statistical test, confirm whether your judgement was correct or not.
3. Plot the cumulative hazard functions for the continuous and batch housing.
4. [**Optional challenge**] Plot the Kaplan-Meier curve for time to calf pneumonia in continuous housed calves, and overlay the Fleming-Harrington survival estimator. Comment on the difference between the two estimators.

6.3 Pointers

When using `survfit()`, there are a number of arguments that are set to the default settings. The argument `type` is one such example, and is set to `type = "kaplan"` by default. Another option is to use `type = "fleming"`. We could create two different survival curve estimates and plot them together.

7 Example 3: Parametric models

To fit a parametric survival model in R, we exploit the survival *regression* function, `survreg()`. We will cover regression modeling in detail during the afternoon lecture; however one can think of fitting a parametric model as fitting a regression model without any explanatory variables.

In Example 2 we generated a subset of the `aml` data named `aml0`, which featured data for all non-maintained chemotherapy patients.

```
aml0 <- aml[aml$x == "Nonmaintained", ]
```

We can fit an exponential survival model to this data using the following command.

```
fit.exp <- survreg(Surv(time, status) ~ 1, data = aml0,
                   dist = "exponential")
```

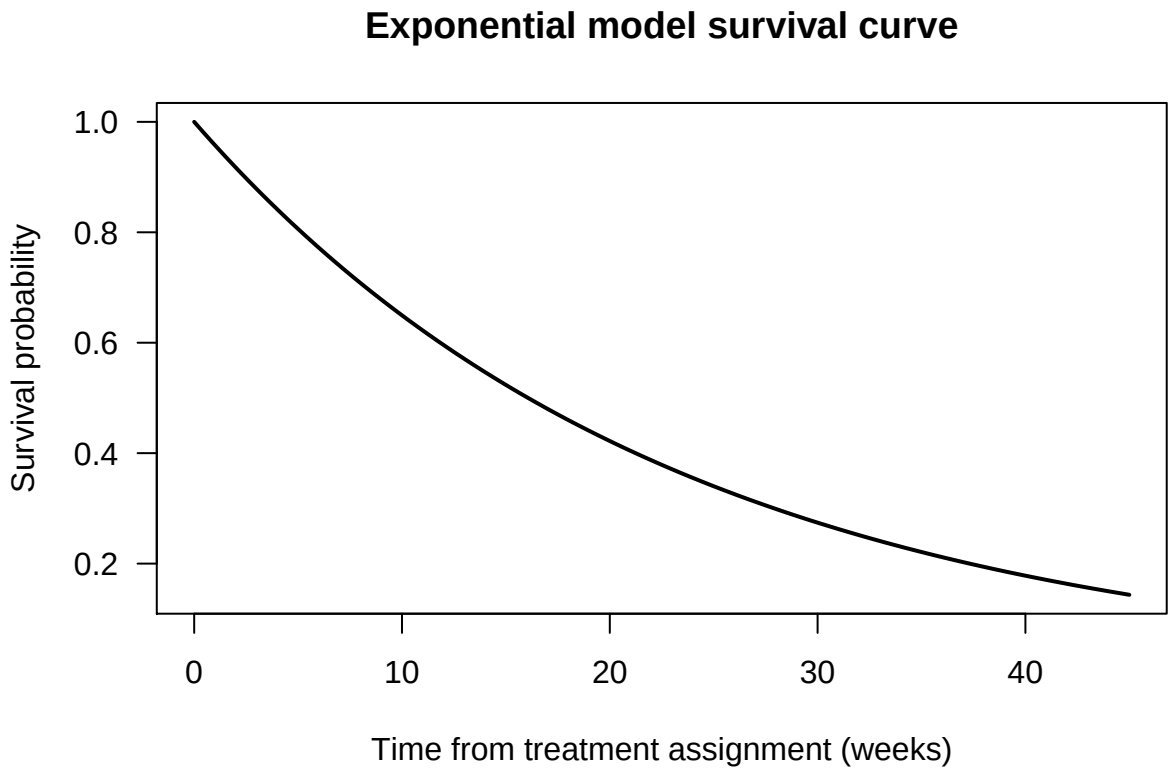
We want to obtain an estimate of the rate parameter λ in the exponential model. R instead gives the maximum likelihood estimate of $-\log(\lambda)$, and is what it calls the `Intercept`. We can therefore extract an estimate of λ by applying the `coef()` function and inverting the relationship. Namely,

```
intercept <- coef(fit.exp)
lambda.hat <- exp(-intercept)
lambda.hat
```

```
## (Intercept)
##  0.04313725
```

We can plot the estimated survival curve manually by recalling that for the exponential model $S(t) = \exp(-\lambda t)$.

```
curve(exp(-lambda.hat * x),
      xlim = c(0, max(aml0$time)),
      lwd = 2,
      xlab = "Time from treatment assignment (weeks)",
      ylab = "Survival probability",
      main = "Exponential model survival curve",
      las = 1)
```



8 Problem 3: Parametric models

8.1 Exercises

These exercises use the `aml` dataset already described above, in particular the data subset of non-maintained chemotherapy patients, `aml0`. It is recommended you spend no more than 30 minutes on these exercises. There are some pointers below if you get stuck.

1. Take the fitted model in Example 3 and overlay the non-parametric Kaplan-Meier estimator. Comment on difference between the two models.
2. Recall that the exponential model is a special case of the Weibull model (with shape parameter $p = 1$). Use the `survreg()` function to fit the Weibull model to the `aml0` data (above). You might want to look in the help file for details.
3. Overlap the Weibull model fit to the plot produced. Comment on the difference between this and the exponential model fit, making reference to the estimated value of p.
4. Which model do you think describes this data best? Explain your reasoning.

8.2 Pointers

Note that R does not automatically give estimates of the Weibull model parameters p and λ for reasons that will become apparent during the lecture on regression. In order to determine estimates of these you will need to know the re-parameterization formulae:

- $p = 1/$`scale`
- $\lambda = \exp(-p \times$ `(Intercept)`$)$

9 Example 4: Instantaneous hazard

If you have completed the exercises above and feel confident you understand them, below is an example for you to work through.

In lectures we described the hazard function, $h(t)$. In Example 2 we looked at how to estimate and plot the cumulative hazard function (or integrated hazard function as it is sometimes known), $H(t)$. Let us estimate the hazard function for the dialysis data, ignoring the different treatment regimens.

We begin by calculating the proportion of failures at each time point, similarly to the Nelson-Aalen calculation.

```r
# Use ~1 as ignoring treatment strata
fit.dial <- survfit(Surv(days, dead) ~ 1, data = dialysis)
ti <- fit.dial$time    # times
ni <- fit.dial$n.risk  # number at risk at ti
di <- fit.dial$n.event # number of relapses
pi <- di / ni
pi[1:20]               # Inspect the first 20 values
```

```
##  [1] 0.008064516 0.008130081 0.008196721 0.008264463 0.008333333
##  [6] 0.008403361 0.008474576 0.008547009 0.008620690 0.008695652
## [11] 0.017543860 0.008928571 0.009009009 0.009090909 0.009174312
## [16] 0.009259259 0.009345794 0.009433962 0.009523810 0.009615385
```

We next need to calculate the time periods that these failures occurred over

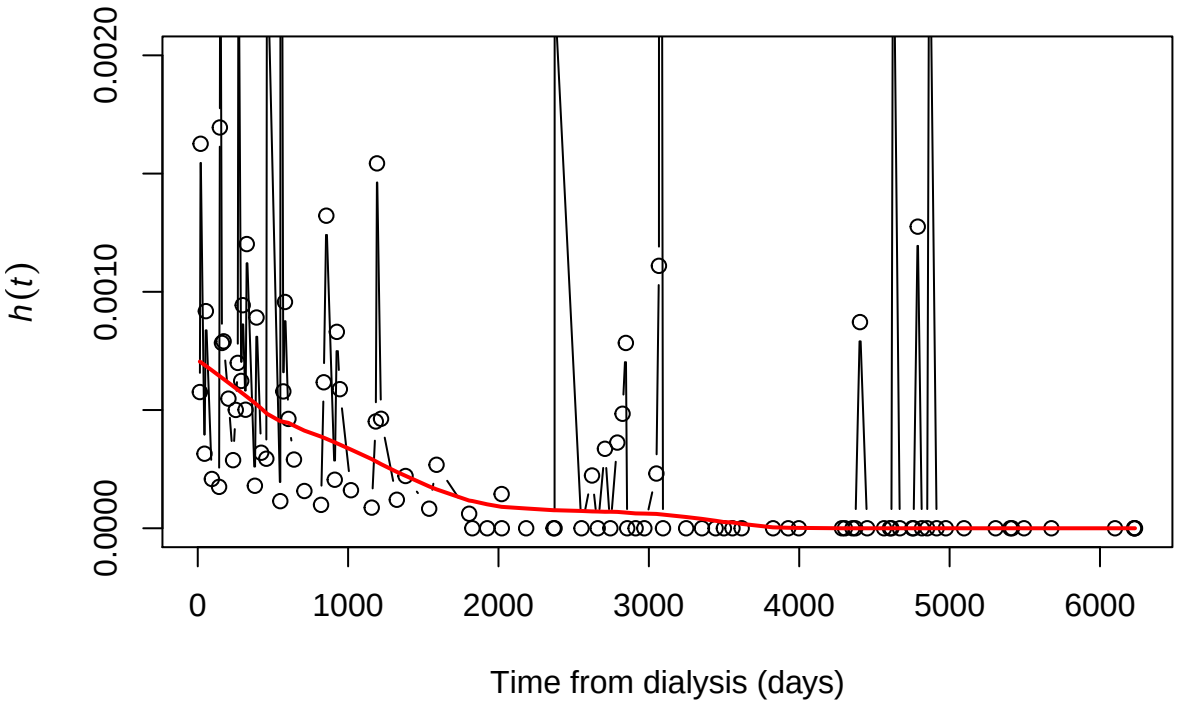
```r
ti0 <- c(0, ti[-length(ti)]) # Append 0 as the start time
ti0[1:20]                    # Inspect the first 20 values
```

```
##  [1]   0  14  19  45  54  94 142 147 150 161 172 204 235 253 266 270 274
## [18] 289 299 318
```

Finally, we divide by failure proportions by the time intervals to get the failure rate per *unit* time. As non-parametric estimates of the hazard function are 'noisy', we add a smoothing curve.[5]

```r
h <- pi / (ti - ti0)
plot(ti, h, type = "b",
     ylim = c(0, 0.002), # Zoom in on smoothed curve
     xlab = "Time from dialysis (days)",
     ylab = expression(italic(h(t)))
     )
lines(lowess(ti, h, f = 1/3), col = 2, lwd = 2)
```

[5] http://en.wikipedia.org/wiki/Local_regression



For reference, there are more straightforward means of estimating this function using packages. For example, the `epi.insthaz()` function in the package `epiR`, or the `kphaz.fit()` function in the `muhaz` package.

10 References

- Therneau TM, Grambsch PM (2000). Modeling Survival Data: Extending the Cox Model. New York: *Springer*.
- Embury SH, et al. (1977) Remission maintenance therapy in acute myelogenous leukaemia. *Western Journal of Medicine*, 126, 267-272.
- Greenwood M (1926). The natural duration of cancer. Reports on Public Health and Medical Subjects, 33: 1-26. London: *Her Majesty's Stationery Office*.
- Diggle PJ and Chetwynd AG (2011). Statistics and Scientific Method: an Introduction for Students and Researchers. Oxford: *Oxford University Press*.
- Dohoo I, et al. (2009). Veterinary Epidemiologic Research. 2nd Edition. *Charlottetown*: AVC Inc.