Chapter 2

The Kaplan–Meier estimate of the survival function

2.1 Introduction to the Kaplan–Meier method

2.1.1 Introduction

The Kaplan–Meier estimate of the survival function is an empirical or non-parametric method of estimating $S(t)$ from non- or right-censored data. It is extremely popular as it requires only very weak assumptions and yet utilises the information content of both fully observed and right-censored data. It comes as standard in most statistical packages (such as R) and can also be calculated by hand (e.g. in exams. . .).

2.1.2 Who were Kaplan and Meier

Both were students of the famous John Tukey. In 1952, Paul Meier started working on the duration of cancer while at Johns Hopkins University, Chicago, USA. Edward Kaplan later started working on the lifetime of vacuum tubes

in the repeaters of sub-oceanic telephone cables while at Bell labs. They independently submitted their research on survival times to the Journal of the American Statistical Association, whose editor encouraged them to submit a joint paper, which they did in 1958: Kaplan, E. L. and P. Meier (1958). Nonparametric Estimation from Incomplete Observations. *J. Am. Stat. Assoc.*, 53:457–481. Google Scholar has 20 000 citations for this paper.

2.1.3 Motivating example: Leukæmia data

We consider remission times for two groups of leukæmia patients. Freireich *et al.* (1963, *Blood*, 21:699:716) applied 6-Mercaptopurine and a placebo to 42 youths ($\leq$ 20 years) with leukæmia. The times of interest are the duration of remission in weeks. These are:

6-MP: 6, 6, 6, 7, 10, 13, 16, 22, 23, 6+, 9+, 10+, 11+, 17+, 19+, 20+, 25+, 32+, 32+, 34+, 35+

Placebo: 1, 1, 2, 2, 3, 4, 4, 5, 5, 8, 8, 8, 8, 11, 11, 12, 12, 15, 17, 22, 23

using the notation $t+$ to indicate a right censored observation at time t.

Let us rewrite the data using the following notation: $d(t)$ is the number of deaths or failures (recorded) at time t, $q(t)$ is the number of right-censorings at time t, and $n(t^-)$ is the total number of individuals *at risk* an instant before time t.

t	$d(t)$	$q(t)$	$n(t^-)$	

2.1.4 The estimate in the absense of censoring

Question: What proportion of the sample given a placebo survived to:

- time 0.0?
- time 0.9?
- time 1.0?
- time 1.1?
- time 2.0?

In the absence of censoring, the empirical survival function $\hat{S}(t)$ is a step function with heights equal to the proportion of the starting population surviving to the instant after t, i.e. $\hat{S}(t) = n(t^+)/n(0)$.

2.1.5 Kaplan–Meier's method in the presense of censoring

What do we do when there is censoring? The above formula requires generalising to the case when there are changing numbers of active participants in the study. The generalisation accounting for $q(t)$ is called the Kaplan–Meier estimate.

The Kaplan–Meier estimate of $S(t)$ is $\hat{S}(t) = \hat{S}(t^-)\hat{\mathrm{p}}(T > t|T \geq t)$.

If no failures occur at time t, $\hat{\mathrm{p}}(T > t|T \geq t) = 1$.

If one or more failures occur at time t,

$$\hat{\mathrm{p}}(T > t|T \geq t) = \frac{n(t^-) - d(t)}{n(t^-)}. \tag{2.1}$$

Note: that the Kaplan–Meier estimate does not change between events, nor at times when only censorings occur. It drops only at times when a failure has been observed. If we write $t_{(i)}$ as the ith ordered event time, and $d_{(i)}$, $q_{(i)}$ and $n_{(i^-)}$ accordingly, the Kaplan–Meier formula can be rewritten

$$\hat{S}(t) = \prod_{t_{(i)} \leq t} \frac{n_{(i^-)} - d_{(i)}}{n_{(i^-)}} \tag{2.2}$$

with $\hat{S}(t) = 1$ for $t < t_{(1)}$.

What about $\hat{S}(t)$ for t greater than the maximum observed event time, $t_{\max}$, say? Various methods have been proposed: Efron (1967) suggested setting $\hat{S}(t) = 0$ for $t > t_{\max}$, Gill (1980) suggested $\hat{S}(t) = \hat{S}(t_{\max})$, and Brown *et al.* (1974) suggested $\hat{S}(t) = \exp\{\log(\hat{S}(t_{\max}))t/t_{\max}\}$. In truth, the best policy is not to attempt to estimate it as the validity of the estimate cannot be assessed. It is better to stop the graph at the last observed event time.

2.2 Variance of the KM estimate

It is natural for we statisticians to want to know how certain our estimate of $S(t)$ is. Within the frequentist framework, this is found via

$$\mathbf{V}\{\hat{S}(t)\} = \mathbf{V}\left\{\prod_{t_{(i)} \leq t} \frac{n_{(i^-)} - d_{(i)}}{n_{(i^-)}}\right\} = \mathbf{V}\left\{\prod_{t_{(i)} \leq t} \hat{p}_{(i)}\right\}. \tag{2.3}$$

The variance of a sum of independent events is easy (the sum of the variances), the variance of a product is hard. So it would be easier to work with the log survival function and then to seek to convert it back later:

$$\mathbf{V}\{\log \hat{S}(t)\} = \mathbf{V}\left\{\sum_{t_{(i)} \leq t} \log \hat{p}_{(i)}\right\} = \sum_{t_{(i)} \leq t} \mathbf{V}\{\log \hat{p}_{(i)}\} \tag{2.4}$$

under the assumption that failures arise independently among the population (usually a safe assumption, but not for contagious diseases, for instance).

2.2.1 The delta method

Reminder Obtaining the variance of a function of a random variable is a problem commonly faced in statistics. It is often approximated via a Taylor expansion around the mean, also known as the "delta method".

Taylor's expansion gives $f(x) = f(a) + (x-a)f'(a) + (x-a)^2 f''(a)/2! + \ldots$

The delta method uses $a = \mu = \mathbf{E}(X)$ and considers just the first order terms above. For example, $\log X \approx \log \mu + (X-\mu)\frac{1}{\mu}$. Thus

$$\begin{aligned}
\mathbf{E}(\log X) &\approx \mathbf{E}(\log \mu) + \mathbf{E}\left\{(X-\mu)/\mu\right\} && (2.5)\\
&= \log \mu && (2.6)\\
\mathbf{V}(\log X) &\approx \mathbf{V}(\log \mu) + \mathbf{V}\left\{(X-\mu)/\mu\right\} && (2.7)\\
&= 0 + \mathbf{V}(X/\mu) + \mathbf{V}(\mu/\mu) && (2.8)\\
&= \frac{\mathbf{V}(X)}{\mu^2}. && (2.9)
\end{aligned}$$

Thus

$$\mathbf{V}\{\log \hat{p}_{(i)}\} \approx \frac{\mathbf{V}\{\hat{p}_{(i)}\}}{\mathbf{E}\{\hat{p}_{(i)}\}^2}. \tag{2.10}$$

Recall that

$$\hat{p}_{(i)} = \frac{n_{(i^-)} - d_{(i)}}{n_{(i^-)}}. \tag{2.11}$$

Under the assumption that failures arise independently with probability $p_{(i)}$, $d_{(i)} \sim \mathrm{Bin}(n_{(i^-)}, 1-p_{(i)})$ and so

$$\begin{aligned}
\mathbf{E}\{\hat{p}_{(i)}\} &= \frac{n_{(i^-)} - n_{(i^-)}(1-p_{(i)})}{n_{(i^-)}} && (2.12)\\
&= p_{(i)} && (2.13)\\
\mathbf{V}\{\hat{p}_{(i)}\} &= \mathbf{V}\left\{\frac{d_{(i)}}{n_{(i^-)}}\right\} && (2.14)\\
&= \frac{1}{n_{(i^-)}^2} n_{(i^-)} p_{(i)}(1-p_{(i)}) && (2.15)\\
&= \frac{p_{(i)}(1-p_{(i)})}{n_{(i^-)}}. && (2.16)
\end{aligned}$$

Therefore

$$\mathbf{V}\{\log \hat{p}_{(i)}\} \approx \frac{1-p_{(i)}}{p_{(i)}n_{(i^-)}}. \tag{2.17}$$

But we don't know $p_{(i)}$! We therefore use our estimate of it to get the following estimated, approximated variance:

$$\begin{aligned}
\hat{\mathbf{V}}\{\log \hat{p}_{(i)}\} &= \frac{1-\hat{p}_{(i)}}{\hat{p}_{(i)}n_{(i^-)}} && (2.18)\\
&= \frac{\frac{n_{(i^-)}-n_{(i^-)}+d_{(i)}}{n_{(i^-)}}}{n_{(i^-)}-d_{(i)}} && (2.19)\\
&= \frac{d_{(i)}}{n_{(i^-)}(n_{(i^-)}-d_{(i)})}. && (2.20)
\end{aligned}$$

We therefore have

$$\mathbf{V}\{\log \hat{S}(t)\} \approx \sum_{t_{(i)}\le t} \frac{d_{(i)}}{n_{(i^-)}(n_{(i^-)}-d_{(i)})}. \tag{2.21}$$

But we really want $\mathbf{V}\{\hat{S}(t)\} = \mathbf{V}\left\{e^{\log \hat{S}(t)}\right\}$! Again, we use the delta method, this time using $e^X \approx e^\mu + (X-\mu)e^\mu$, so $\mathbf{V}(e^X) \approx (e^\mu)^2\mathbf{V}(X)$.

Thus,

$$\begin{aligned}
\mathbf{V}\{\hat{S}(t)\} &= \mathbf{V}\left\{e^{\log \hat{S}(t)}\right\} && (2.22)\\
&\approx e^{2\mathbf{E}\{\log \hat{S}(t)\}}\mathbf{V}\{\log \hat{S}(t)\} && (2.23)\\
&\approx e^{2\mathbf{E}\{\log \hat{S}(t)\}}\sum_{t_{(i)}\le t}\frac{d_{(i)}}{n_{(i^-)}(n_{(i^-)}-d_{(i)})} && (2.24)
\end{aligned}$$

and so we have

$$\hat{\mathbf{V}}\{\hat{S}(t)\} = \hat{S}(t)^2\sum_{t_{(i)}\le t}\frac{d_{(i)}}{n_{(i^-)}(n_{(i^-)}-d_{(i)})}. \tag{2.25}$$

This is Greenwood's estimator of the variance of the survival function (Greenwood, 1926).

2.2.2 Confidence interval for $S(t)$

We could try using Greenwood's estimate to construct asymptotic confidence intervals for $S(t)$.

Example: Pollock *et al.* (1989) radio-tagged 18 quail (*Colinus virginianus* L.) and followed their survival. The following are death or censoring times in weeks, using the same notation as before:

3, 3, 6, 8, 8+, 9, 9+, 9+, 10, 10+, 12+, 13+, 13+, 13+, 13+, 13+, 13+, 13+.

The Kaplan–Meier estimate of the survival function, the variance of this estimate, and a 95% confidence interval constructed in the usual way, are as below.

t	$\hat{S}(t)$	$\mathbf{V}\{\hat{S}(t)\}$	95%CI
1	1.00	0.00^2	(1.00,1.00)
2	1.00	0.00^2	(1.00,1.00)
3	0.89	0.07^2	(0.74,1.03)
4	0.89	0.07^2	(0.74,1.03)
5	0.89	0.07^2	(0.74,1.03)
6	0.83	0.09^2	(0.66,1.01)
7	0.83	0.09^2	(0.66,1.01)
8	0.78	0.10^2	(0.59,0.97)
9	0.72	0.11^2	(0.51,0.93)
10	0.65	0.12^2	(0.41,0.88)
11	0.65	0.12^2	(0.41,0.88)
12	0.65	0.12^2	(0.41,0.88)
13	0.65	0.12^2	(0.41,0.88)

This approach may lead to CIs that exceed the range of $S(t)$, which look non-sensical.

An alternative was proposed by Kalbfleisch and Prentice (2002) that gets around this problem, namely to use

$$\hat{\mathbf{V}}\{\log(-\log \hat{S}(t))\} = \frac{1}{(\log \hat{S}(t))^2} \sum_{t_{(i)} \le t} \frac{d_{(i)}}{n_{(i^-)}(n_{(i^-)} - d_{(i)})} \tag{2.26}$$

(again using the delta method) to get a confidence interval of $\log(-\log \hat{S}(t))$, and then to transform that back into the original scale. The motivation is the fact that if $a = \log(-\log b)$ then $a \in (\infty, \infty) \iff b \in (0, 1)$.

Precisely, they propose letting

$$c_1 = \log(-\log \hat{S}(t)) + z_{1-\alpha/2}\sqrt{\mathbf{V}\{\log(-\log \hat{S}(t))\}} \quad (2.27)$$

$$c_2 = \log(-\log \hat{S}(t)) - z_{1-\alpha/2}\sqrt{\mathbf{V}\{\log(-\log \hat{S}(t))\}} \quad (2.28)$$

so that the $1-\alpha$ confidence interval for $S(t)$ is $(\exp\{-e^{c_2}\}, \exp\{-e^{c_1}\})$.

Note that this doesn't work for $\hat{S}(t) = 0$ or 1. In such cases, I suggest using $(0, 0)$ or $(1, 1)$ as the confidence interval if needed graphically, and otherwise not to offer a confidence interval for those points.

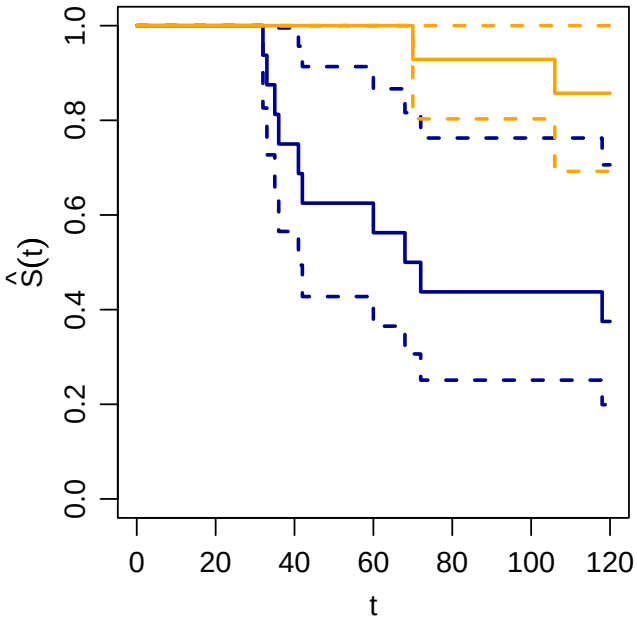
2.2.3 Simultaneous confidence interval for $S(t)$

If you connect up the confidence intervals for all times, you get a *pointwise confidence band* for $S(t)$. This has the interpretation that the true $S(t)$ for any particular t will be within the band for 95% (or, generally, $1-\alpha$) of experiments you conduct. If you wish the coverage to be 95% *for all times jointly*, the confidence band must be expanded in size to obtain a *simultaneous confidence band*. Hall and Wellner (1980) proposed a method for doing so, but it is fairly complex and not readily available in computer packages, so we shall not cover it here.

2.3 Testing differences in survival curves

2.3.1 Introduction

As in most statistics, a key objective is to test whether subpopulations behave in the same way.



For example, a group of patients has been allocated to treatment with omega 3 oils or a placebo by Stoll *et al.* (1999) and the duration without an attack recorded and plotted (orange for omega 3, blue for placebo). It appears that the two subpopulations do differ, with the survival curve of the patients on the drug lying above that of the placebo patients. Is this a genuine finding, or can it be explained by a small sample size giving the spurious impression of a difference?

Various tests have been proposed for testing for differences in survival between categorical covariates; we present three named ones. They all have a very similar structure, but have different power depending on what the exact nature of the difference between the survival curves is.

2.3.2 Notation

Let there be K categories.

Let $d_{k,(i)}$ be the number of failures in group k at ordered time $t_{(i)}$, where the ordering is over all categories.

Let $d_{(i)} = \sum_{k=1}^{K} d_{k,(i)}$ be the total number of failures at time $t_{(i)}$.

Let $n_{k,(i^-)}$ be the number of members of group k at risk an instant before $t_{(i)}$ and $n_{(i^-)}$.

2.3.3 Test with two categories

Initially we limit attention to two categories.

Hypotheses

H_0: $S_1(t) = S_2(t)$

H_1: $S_1(t) \neq S_2(t)$

Test-statistic

If H_0 is true, the expected number of deaths in group k at time $t_{(i)}$ is

$$\hat{e}_{k,(i)} = \frac{n_{k,(i^-)} d_{(i)}}{n_{(i)}}. \tag{2.29}$$

The $n_{k,(i^-)}$ term is the number at risk in category k. The ratio $d_{(i)}/n_{(i)}$ is the overall proportion in both populations failing at time $t_{(i)}$. The variance in $d_{k,(i)}$ is given by the variance of the hypergeometric distribution:

$$\hat{V}_{1,(i)} = \hat{V}_{2,(i)} = \frac{n_{1,(i^-)} n_{2,(i^-)} d_{(i)} (n_{(i^-)} - d_{(i)})}{n_{(i^-)}^2 (n_{(i^-)} - 1)}. \tag{2.30}$$

The test-statistic is then

$$q = \frac{\left[\sum_{i=1}^{m} w_i (d_{1,(i)} - \hat{e}_{1,(i)})\right]^2}{\sum_{i=1}^{m} w_i^2 \hat{v}_{1,(i)}} \tag{2.31}$$

for some weights w_i (see later) where m is the number of distinct death/failure times. Note that the statistic is based on one subpopulation's sample moments only, as the other is deterministic conditional on the first.

If H_0 is true, $q \sim \chi^2_1$ asymptotically.

Assumptions:

- Censoring is independent of group.
- $\sum_{i=1}^{m} d_{(i)}$ is large.
- $\sum_{i=1}^{m} e_{k,(i)}$ is large.

p-value

The p-value, $p = \mathrm{p}(Q > q|\mathrm{H}_0 \text{ true})$, follows from the CDF of the χ^2_1 distribution.

Weights

The various tests differ in terms of the weights used, and hence the kind of discrepancies between the survival functions that the test is best able to pick up.

- The *log-rank test* uses $w_i = 1$. It puts emphasis on larger values of time.
- The *(generalised) Wilcoxon test* uses $w_i = n_{(i^-)}$. It puts emphasis on smaller values of time.
- The *Tarone–Ware test* uses $w_i = \sqrt{n_{(i^-)}}$. It puts emphasis on intermediate values of time.

The three tests can simply be effecting in R by setting `rho=0, 1` or `0.5`, respectively. See next lecture.

2.3.4 Tests with multiple categories

If there are $K \geq 3$ subpopulations of interest, similar tests can be constructed by generalising notation to use matrix algebra.

Notation

Let $\mathbf{d_{(i)}}^{\mathrm{T}} = (d_{1,(i)}, d_{2,(i)}, \ldots, d_{K-1,(i)})$.

Let $\hat{\mathbf{e}}_{\mathbf{(i)}}^{\mathrm{T}} = (e_{1,(i)}, e_{2,(i)}, \ldots, e_{K-1,(i)})$.

Note that both are of length $K-1$ for the same reason as we used only one category in the two category case.

The $(K-1) \times (K-1)$ covariance matrix $\hat{\mathbf{V}}_{\mathbf{(i)}}$ has diagonal elements

$$\hat{v}_{k,k,(i)} = \frac{n_{k,(i^-)}(n_{(i^-)} - n_{k,(i^-)})d_{(i)}(n_{(i^-)} - d_{(i)})}{n_{(i^-)}^2(n_{(i^-)} - 1)} \tag{2.32}$$

and off-diagonal elements

$$\hat{v}_{k,j,(i)} = -\frac{n_{k,(i^-)}n_{j,(i^-)}d_{(i)}(n_{(i^-)} - d_{(i)})}{n_{(i^-)}^2(n_{(i^-)} - 1)}. \tag{2.33}$$

The weight matrix is $\mathbf{W_i} = w_i I_{K-1}$ where I_{K-1} is the $(K-1) \times (K-1)$ identity matrix.

Hypotheses

H_0: $S_1(t) = S_2(t) = \ldots = S_K(t)$.

H_1: There is at least one pair of categories k and j such that $S_j(t) \neq S_k(t)$.

Test-statistic

If H_0 is true, using the same logic as in the two-category case, the test-statistic is then

$$q = \left[\sum_{i=1}^{m} \mathbf{W_i}(\mathbf{d_{(i)}} - \hat{\mathbf{e}}_{\mathbf{(i)}})\right]^{\mathrm{T}} \left[\sum_{i=1}^{m} \mathbf{W_i}\hat{\mathbf{V}}_{\mathbf{(i)}}\mathbf{W_i}\right]^{-1} \left[\sum_{i=1}^{m} \mathbf{W_i}(\mathbf{d_{(i)}} - \hat{\mathbf{e}}_{\mathbf{(i)}})\right]. \tag{2.34}$$

If H_0 is true, under the same assumptions as for the 2 subpopulation case, $q \sim \chi^2_{K-1}$, and the p-value can be found in a similar fashion.

2.3.5 Discussion

These tests are very useful in assessing whether a covariate affects survival. However, they do not allow us to say *how* survival is affected. Ideally, we'd like to be able to say how much more at risk on group is than another. We'd also like to be able to incorporate non-categorical covariates, such as age. An *ad hoc* approach is to categorise the covariate arbitrarily. A more satisfying approach is to do some semi-parametric modelling to investigate the functional relationship between the covariates and survival. This can be done using Cox's proportional hazards model, which we will discuss in the next part of the course, chapter 3.

2.4 Finding the Kaplan–Meier estimate in R

In the earlier part of this chapter, we discussed how to calculate the Kaplan–Meier estimate of $S(t)$ by hand and to do tests of the hypothesis that different categories of individuals have the same survival functions. Luckily these routines have been incorporated in many standard statistical packages such as R, Splus, SAS, SPSS, etc. In this course, we will use R to analyse data, but you are welcome to use your preferred statistics package as long as you do not seek any support from me. The advantages of R over other packages are: it is open-source, free, uses easily-replicated command lines rather than opaque and forgettable combinations of menu clicks, and it has a vast collection of add-on packages contributed by the user community.

To start R in a GUI like windows, either double click on its icon if it is on the desktop or search for the program in the start menu. In a unix-like environment, open a command line terminal and type `R` at the prompt.

There are a variety of optional packages relating to survival analysis. To obtain a list of all such packages, type

```
> help.search("survival")
```

We shall use the survival package. To load this, type

```
> library(survival)
```

Complete documentation for this package can be obtained at

`http://cran.r-project.org/web/packages/survival/index.html`.

Documentation on any particular function can be obtained with the `?functionname` command, e.g.

```
> ?survfit
```

2.4.1 Loading data

The first thing we want to do is to load data in to R. There are three scenarios:

- Use data R knows about already. For pedagogic and illustrative purposes, some data sets are available to R automatically. One such data set is the `aml` data in the survival package. This details survival in patients with acute myelogenous leukæmia, as described in Miller (1997) Survival Analysis, Wiley and Sons. Since we've already loaded the survival package, we can see these data by typing

  ```
  > aml
  ```

  The data are in a structure with three sub-elements: `time` (to death or censoring), `status` (indicating censoring if status = 0) and `x` (a factor indicating if maintainence chemotherapy was given). To see survival and censoring times, type

  ```
  > aml$time
  ```

  You can do a plot by typing

```
> hist(aml$time)
```

for example.

- Load your own data from a file. More often, you will wish to load in your own data from a text file. This can be done with the `read.table` or `scan` commands. If you are manually creating the data set file, use your favourite text editor and save your file as a `.txt` or `.dat` file. Good practice is to put it in its own directory with a suitable name and an associated readme file explaining its contents to posterity.

  If the file is in the current working directory, you can just give R its name in the `read.table` command and all will be well. To find the current working directory, type

  ```
  > getwd()
  ```

  Otherwise, you must specify the file's location in either relative or absolute terms. Here are some examples (not for real files!):

  ```
  #reads the file dataset.dat from the working directory:
  > read.table("dataset.dat")
  #reads the file dataset.dat from my web page:
  > read.table("http://courses.nus.edu.sg/course/stacar/internet/dataset.dat")
  #reads the file dataset.dat from the directory mydata
  #  on your computer's c drive:
  > read.table("C:/mydata/dataset.dat")
  ```

  Let us read the data `bipolar` from my webpage into a variable called `s_data`:

  ```
  > s_data =
    read.table("http://courses.nus.edu.sg/course/stacar/internet/bipolar.dat",col.names=T)
  ```

  These data are the times of no attack of bipolar disorder in a group of patients given omega 3 or a placebo. See Stoll *et al.* (1999, *Arch. Gen. Psychiatry*, 56:407–12). There are three columns, `t`, `d` and `o`. The first gives the survival or censoring time in days, the second indicates if the event is survival (`d=1`) or censoring (`d=0`), and the third indicates if the patient received omega 3 oil (`o=1`) or the placebo (`o=0`).

- Form your own data structure from disparate elements. Suppose that you have manipulated some vectors of data and with to combine them

together into a single structure. This could be simulated data (as follows) or could result from having each part of the data stored in its own file. Let us create a simulated data set with three elements: `time`, `censor` and an imaginary covariate `x`:

```
> N = 100 #the sample size
> covariate = rnorm(N,0,1) #100 draws from a N(0,1) distribution
#give them an exponential lifetime affected by the covariate:
> time = rexp(N,covariate^2)
> censored = (time>10) #ie =1 if time>10 and 0 else
#put the data into a structure called sim:
> sim = list(x=covariate,time=time,censor=censored)
> sim #look at the resulting data
```

The first thing you should do once the data are loaded is to plot them! For example:

```
> hist(aml$time,freq=F)
> lines(density(subset(aml$time,aml$x=="Maintained"),from=0),col=2)
> lines(density(subset(aml$time,aml$x=="Nonmaintained"),from=0),col=3)
```

2.4.2 Kaplan–Meier Plots in R

The main function for the Kaplan–Meier estimator in R is called `survfit`. It takes as its primary argument a "formula object". The formula object is of form:

```
a survival object ~ covariate terms
```

If you do not wish to use any covariates, the "$\sim$ `covariate terms`" can be omitted. Survival objects are created by the `Surv` (note capital S) function. This takes arguments `time` (i.e. event time) and `event` (0 for right censored, 1 for failure). For example, for the `aml` data we may use

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

The covariate terms in the formula object are specified via a symbolic model formula. This might resemble ∼ `x1 + x2 + x1*x2` for two covariates `x1` and `x2` including an interaction term. For example, the `aml` data have one binary categorical covariate `x`. Two Kaplan–Meier curves can be created by specifying ∼`aml$x`. Thus we can use the Kaplan–Meier routine on the `aml` data by entering:

```
survfit(Surv(aml$time,aml$status)~aml$x)
```

An alternative that avoids all the `aml`s is

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

Let's store the fit in an eponymous variable:

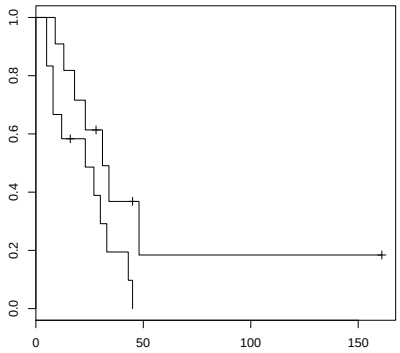
```
fit = survfit(Surv(aml$time,aml$status)~aml$x)
```

to be manipulated later. The function `survfit` has various options, including

```
#just fit to elements 1--11 in the data set:
, subset=1:11
#just fit to elements that have an x equal to the string "Maintained":
, subset=(x=="Maintained")
#do a log-log transformation to create CIs:
, conf.type="log-log"
#create CIs on the original scale:
, conf.type="plain"
#constructs a 90%CI instead of the default of 95%:
, conf.int=0.9
```

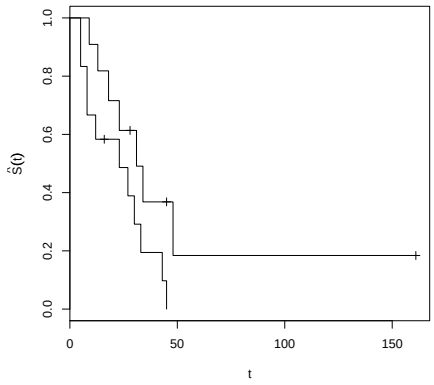
The `plot` function is preconfigured for `survfit` output. To plot the output (which we've stored in `fit`), type:

```
plot(fit)
```



Note that the axes are not labelled, a terrible sin. Correct it via

```
plot(fit,xlab="t",ylab=expression(hat(S)(t))
```

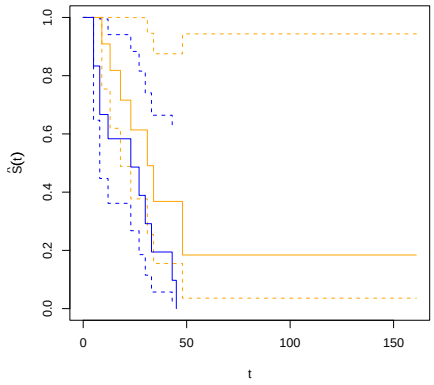


Syntax for the `expression` function can be obtained in Murrel & Ihaka (2000, *J. Comp. Graph. Stat.* 9:582–599). By default, `plot` plots just the estimators $\hat{S}_i(t)$ if there are more than one category i. With just one category it plots in addition a confidence interval. There are various additional options to `plot` that change the defaults:

```
#switch off the marks indicating censoring events:
, mark.time=F
#use a vertical line rather than a + to represent censoring events:
, mark="|"
#force CIs to be on:
, conf.int=T
#force CIs to be off:
, conf.int=F
#use NUS colours to distinguish the lines:
, col=c("orange","blue")
#use red and blue lines:
, col=c(2,4)
#put a legend with labels A and B on the plot
#(though informative labels would be far better):
, legend.text=c("A","B")
#put the legend in the top right corner of the plot area:
, legend.pos=1
, log=T #plots log S(t) versus t
, fun=log #ditto
, fun=sqrt #plots sqrt S(t) versus t
, fun=cumhaz #plots cumulative hazard
#plot a log log plot, changing the x axis too somehow:
, fun="cloglog"
```

I personally do not like these default settings. Here is my preferred version for a binary explanatory variable:

```
> fit$label=c(rep(1,fit$strata[1]),rep(2,fit$strata[2]))
> t1=c(0,subset(fit$time,fit$label==1));t2=c(0,subset(fit$time,fit$label==2))
> St1=c(1,subset(fit$surv,fit$label==1));St2=c(1,subset(fit$surv,fit$label==2))
> uSt1=c(1,subset(fit$upper,fit$label==1));uSt2=c(1,subset(fit$upper,fit$label==2))
> lSt1=c(1,subset(fit$lower,fit$label==1));lSt2=c(1,subset(fit$lower,fit$label==2))
> plot(0,0,pch=" ",ylim=0:1,xlim=range(t1,t2),xlab="t",ylab=expression(hat(S)(t)))
> lines(t1,uSt1,lty=2,type='s',col="orange");lines(t1,lSt1,lty=2,type='s',col="orange")
> lines(t2,uSt2,lty=2,type='s',col="blue");lines(t2,lSt2,lty=2,type='s',col="blue")
> lines(t1,St1,type='s',col="orange");lines(t2,St2,type='s',col="blue")
```

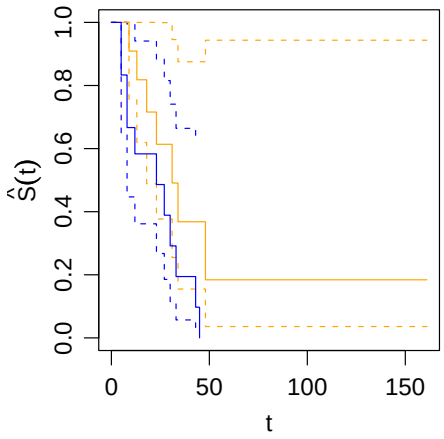


You can output your plots to a file to be included in reports, etc. To create a pdf file, do the following:

```
> cm=1/2.54;pdf("myplot.pdf",height=10*cm,width=10*cm)
#R uses inches by default.
#Change dimensions of the plot to suit your requirements.
> #put all your plotting commands here
> dev.off()
```

Again, the defaults are not very good—there is too much white space in the wrong place. Try

```
> cm=1/2.54;pdf("myplot.pdf",height=10*cm,width=10*cm)
#change margins and marginal spacing respectively
#(See ?par for details):
> par(mai=c(2,2,0.5,0.5)*cm,mgp=c(2,0.75,0))
#change size of text etc:
> par(cex=1.25)
> #put all your plotting commands here
> dev.off()
```



Alternatives to `pdf` include

```
> postscript("myplot.ps")
> jpeg("myplot.jpeg")
> png("myplot.png")
```

Note that postscript and pdf formats are vector-based and hence lossless: try zooming in on a pdf and it will still look good, but zoom in on a jpeg and it will look blocky. Note however that if you put any graphics into a Microsoft Office document, the chances are that no matter how good the original the final graphic will look terrible.

As well as plotting the output, try summarising them:

```
> summary(fit)
```

This prints out a table with rows corresponding to failure times, and columns giving these times, the number at risk, the number of failures, the KM estimate of the survival function, Greenwood's estimate of the standard error of this estimate, along with lower and upper bounds on a confidence interval. Note that the confidence interval can be inaccurate, for instance, it sets equal to 1 any upper bounds that go above 1.

2.4.3 Tests in R

Tests of the hypothesis that subpopulations have the same $S(t)$ can be done easily using the `survdiff` function. This takes a formula expression as its primary argument. It allows the following further arguments:

```
> subset= #as survfit
> rho=0
```

The `rho` option puts weights $\hat{S}(t)^\rho$ on the summands in the test statistic. If $\rho = 0$ we have the log-rank test as all event times have the same weight. If $\rho = 1$ we have the generalised Wilcoxon test. If $\rho = 1/2$ we have the Tarone–Ware test. The ρ notation is due to Harrington and Fleming (1982) Biometrika 69:553–66.

For example, let us test whether maintenance chemotherapy has any effect on survival times due to acute myelogenous leukæmia.

```
> survdiff(Surv(time,status)~x,data=aml,rho=1)
Call:
survdiff(formula = Surv(time, status) ~ x, data = aml, rho = 1)

                 N Observed Expected (O-E)^2/E (O-E)^2/V
x=Maintained    11     3.85     6.14      0.86      2.78
x=Nonmaintained 12     7.18     4.88      1.08      2.78

 Chisq= 2.8  on 1 degrees of freedom, p= 0.0955
> survdiff(Surv(time,status)~x,data=aml,rho=0)
Call:
survdiff(formula = Surv(time, status) ~ x, data = aml, rho = 0)

                 N Observed Expected (O-E)^2/E (O-E)^2/V
x=Maintained    11        7    10.69      1.27      3.40
x=Nonmaintained 12       11     7.31      1.86      3.40

 Chisq= 3.4  on 1 degrees of freedom, p= 0.0653
```

The second of these, the log-rank test, puts more emphasis on larger values of time, which is where the main difference appears to be (from the graphs), giving it a smaller p-value. Note that it's a bit unprincipled to keep doing tests until you get a p-value you want, so you should really decide on the weights before analysing the data.