Kaplan-Meier in SAS

```
filename foo url "http://math.unm.edu/~james/small.txt";

data small;
  infile foo firstobs=2;
  input time censor;
run;

proc print data=small;
run;

proc lifetest data=small plots=survival;
  time time*censor(0); *first "time" is a keyword, second "ti
run;
```

Kaplan-Meier in SAS

Dealing with left-truncated data

The K-M curve allowed for right-censored data. This approach can be modified to allow left-truncated data as well.

The idea is that a patient might not have entered the study until a certain age (for example, only patients who are senior citizens are enrolled, or only patients at a retirement home). This can also be useful to determine the survival curve conditional on already have reached a certain age or conditional on having lived a certain amount of time past diagnosis or treatement.

For this type of data, the jth individual has a random age L_j at which the study was joined and a time of death T_j.

Dealing with left-truncated data

For this type of data, d_i and t_i are defined as before, with d_i being the number of deaths observed at time i and t_i being the ith distinct death time, but Y_i is modified. In particular, Y_i is the number of individuals satisfying $L_j < t_i \leq T_j$. In other words,

$$Y_i = \sum_{j \in \{\text{individuals}\}} I(L_j < t_i \leq T_j)$$

where $I(\cdot)$ is an indicator function equal to 1 if the condition is true and 0 otherwise.

Why are the inequalities strict and then not strict?

Dealing with left-truncated data

To understand the inequalities, we presume that someone is alive at the time they are enrolled, so that if $L_j = t_i$, then individual j couldn't haven't died at time t_i, so we exclude these cases.

Also, if $L_j < t_i$, then if $T_j = t_i$ (which is possible), then they were one of the patients available to die at time t_i who did in fact die at that time.

Dealing with left-truncated data

Essentially, Y_i is defined as before, but we reduce the counts for people available to die at time t_i by not included those individuals who are left-truncated at time t_i – they couldn't possibly have died by that time because of the way they were recruited into the study.
The product-limit estimator is then given conditional on survival past a certain age a as

$$\widehat{S}_a(t) = \prod_{a \le t_i \le t} \left[1 - \frac{d_i}{Y_i}\right], \quad t \ge a$$

Given the same sample of data, you could compute, for example

$$\widehat{S}_{70}(t)$$

and $\widehat{S}_{80}(t)$, which would give different survival curves conditional on having already lived to 70 and 80 years old, respectively. This is deliberately left-truncating your data to determine conditional survival curves.

Example, Channing house data

Data on survival times for patients in a retirement home.

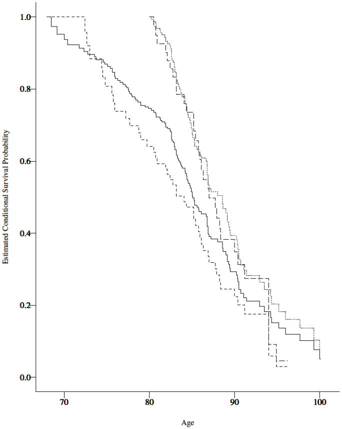


Figure 4.11 *Estimated conditional survival functions for Channing house residents. 68 year old females (———); 80 year old females (------); 68 year old males (— — —); 80 year old males (——— —).*

Example, psychiatric patients

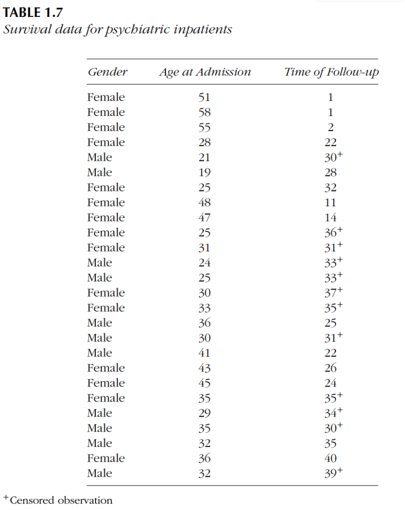


TABLE 1.7
Survival data for psychiatric inpatients

Gender	*Age at Admission*	*Time of Follow-up*
Female	51	1
Female	58	1
Female	55	2
Female	28	22
Male	21	30^{+}
Male	19	28
Female	25	32
Female	48	11
Female	47	14
Female	25	36^{+}
Female	31	31^{+}
Male	24	33^{+}
Male	25	33^{+}
Female	30	37^{+}
Female	33	35^{+}
Male	36	25
Male	30	31^{+}
Male	41	22
Female	43	26
Female	45	24
Female	35	35^{+}
Male	29	34^{+}
Male	35	30^{+}
Male	32	35
Female	36	40
Male	32	39^{+}

$^{+}$Censored observation

Example, psychiatric patients

Problem 4.8 in the book's homework problems asks you to plot Y_i as a function of time for this data, treating the data as left-censored based on the entry age of each patient. The problem also asks you to compute the conditional survival function given that the patient entered the hospital at age at least 30. (Note that the data is left-censored because a patient couldn't have been observed to die earlier than their entrance age, yet from this population, there might have been patients who had died (for example, by suicide) before being admitted.)

Example, psychiatric patients

The unique death times can be found using the `unique()` function in R

```
> enter <- c(58,58,59,60,60,61,61,62,62,62,63,63,64,66,
66,67,67,67,68,69,69,69,70,70,70,71,72,72,73,73)
> exit <- c(60,63,69,62,65,72,69,73,66,65,68,74,71,68,69,
70,77,69,72,79,72,70,76,71,78,79,76,73,80,74)
> censor
 [1] 1 1 0 1 1 0 0 0 1 1 1 0 1 1 1 1 1 1 1 0 1 1 0 1 0 0 1 1 0 1
> length(censor)
[1] 30
> length(enter)
[1] 30
> length(exit)
[1] 30
> ti <- sort(unique(exit[censor==1]))
> ti
 [1] 60 62 63 65 66 68 69 70 71 72 73 74 76 77
```

Example, psychiatric patients

To count Y_1, for example, this is the number of patients available to have died at age 60 in the hospital. This includes the one individual who did die at age 60, plus those who were admitted to the hospital at less than 60 years of age but who died (or were right-censored) at age 60 or later. From the table of data, there were only 3 such individuals.

For Y_2, the death time is $t_2 = 62$, so we want the number of individuals enrolled at age younger than 62 who were available to die at age 62. There were 7 individuals enrolled before age 62, one of whom died at age 60, so was unavailable. The rest were available, so $Y_2 = 60$.

Example, pyschiatric patients

Some code for computing Y_i from the other data is

```
> y <- 1:14 #number of unique death times
> for(i in 1:14) {
+ y[i] <- length(enter[enter < ti[i] & ti[i] <= exit])
+ }
> y
 [1]  3  6  8 10  8 12 11 10 11 10  9  9  7  5
```

It is typical for left-truncated data that the Y_i terms are small initially, then increase, then decrease again. The truncation can make it so that there aren't many patients avaiable for earlier times. For data we've seen previously that was only right-censored, the Y_i terms were decreasing.

Example, psychiatric patients

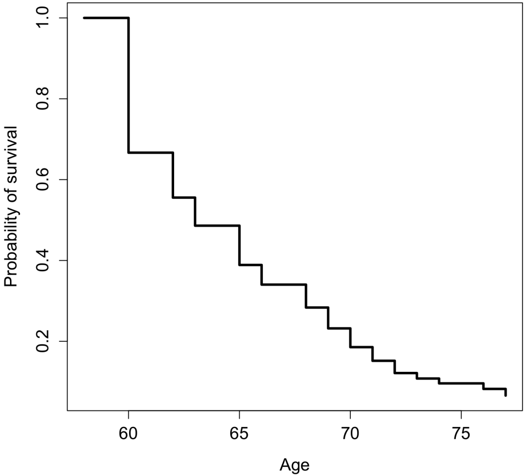
Computing d_i from the other data:

```
> d <- 1:14
> for(i in 1:14) {
+ d[i] <- sum(exit==ti[i] & censor==1)
+ }
> d
 [1] 1 1 1 2 1 2 2 2 2 2 1 1 1 1
> sum(d)
[1] 20 # check that the total number of deaths is correct
```

Example, psychiatric patients

```
> km <- 1:14
> for(i in 1:14) {
+ }
> km <- 1:15
> for(i in 1:14) {
+ km[i+1] <- km[i]*(1-d[i]/y[i])
+ }
> km
 [1] 1.00000000 0.66666667 0.55555556 0.48611111 0.38888889 0.34027
 [7] 0.28356481 0.23200758 0.18560606 0.15185950 0.12148760 0.10798
[13] 0.09599021 0.08227732 0.06582185
> ti <- c(58,ti)
> plot(ti,km,type="s",xlab="Age",ylab="Probability of survival",lwd
cex.axis=1.3,cex.lab=1.3)
```

Example, psychiatric patients



Example, psychiatric patients

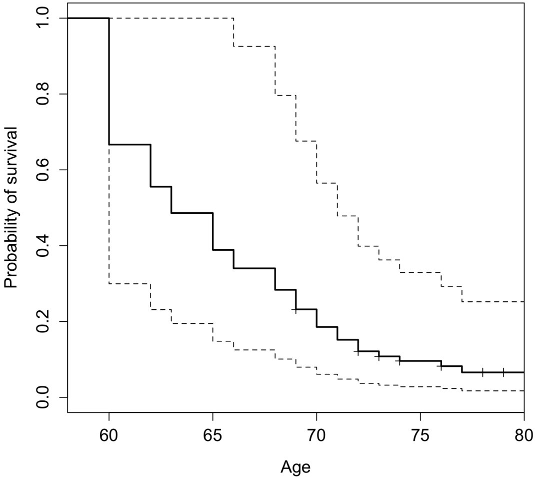
Of course, there is an easier way in R.

```
km2 <- survfit(Surv(enter,exit,censor)~1,type="kaplan-meier")
plot(km2,xlim=c(58,80),xlab="Age",ylab="Probability of survival",
lwd=c(2,1,1),cex.axis=1.3,cex.lab=1.3)
```

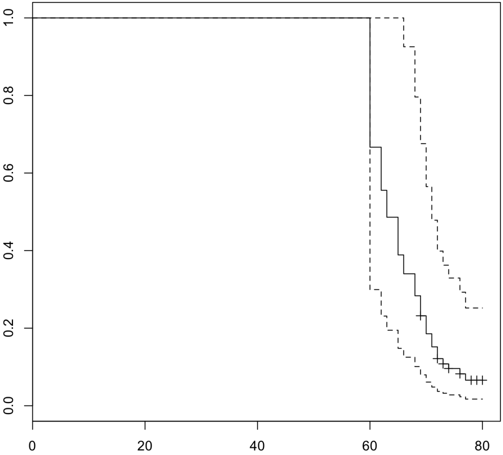
Example, psychiatric patients

```
> summary(km2)
Call: survfit(formula = Surv(enter, exit, censor) ~ 1, type = "kapla
 time n.risk n.event entered censored survival std.err lower 95% CI
   60      3       1       2        0   0.6667  0.2722       0.2995
   62      6       1       3        0   0.5556  0.2485       0.2312
   63      8       1       2        0   0.4861  0.2269       0.1947
   65     10       2       0        0   0.3889  0.1916       0.1480
   66      8       1       2        0   0.3403  0.1737       0.1251
   68     12       2       1        0   0.2836  0.1493       0.1010
   69     11       2       3        2   0.2320  0.1266       0.0796
   70     10       2       3        0   0.1856  0.1054       0.0610
   71     11       2       1        0   0.1519  0.0889       0.0482
   72     10       2       2        1   0.1215  0.0737       0.0370
   73      9       1       2        1   0.1080  0.0667       0.0322
   74      9       1       0        1   0.0960  0.0604       0.0280
   76      7       1       0        1   0.0823  0.0533       0.0231
   77      5       1       0        0   0.0658  0.0451       0.0172
```

Example, psychiatric patients



Example, psychiatric patients, default plot with `plot(km2)`



Other types of censoring and truncation (Chapter 5)

This chapter deals with methods for left-censoring and inter-censoring, as well as right-truncation. In the rare case of having left-censoring but no right-censoring, you can do a trick which is to pick a time τ which is larger than any observed time (censored or not). Then define all times as $x_i = \tau - t_i$. Then do you're analysis on the x_i times, treating observations that were censored as right-censored. The product-limit estimator still works, but then is estimating

$$P[\tau - X > t] = P[X < \tau - t]$$

which is still a decreasing function of t.

Ohter types of censoring and truncation (Chapter 5)

We'll first consider an example with left- and right-censoring.

The example is in section 1.17, on the time to first marijuana use for high school boys in California (based on a study in the 1970s). Boys were asked "When did you first use marijuana", and ages are given, so data are to the nearest year. If a 15-year old responds "I have never used it?", how should this be recorded?

Marijuana example

If the boy says he has never used it, then we can treat this as right-censored at age 15 in this case since he might use it in the future. If the boy says he's been using it at least one year, then it would be left-censored at age 14 since he might have started at age 13 or earlier.

Marijuana example

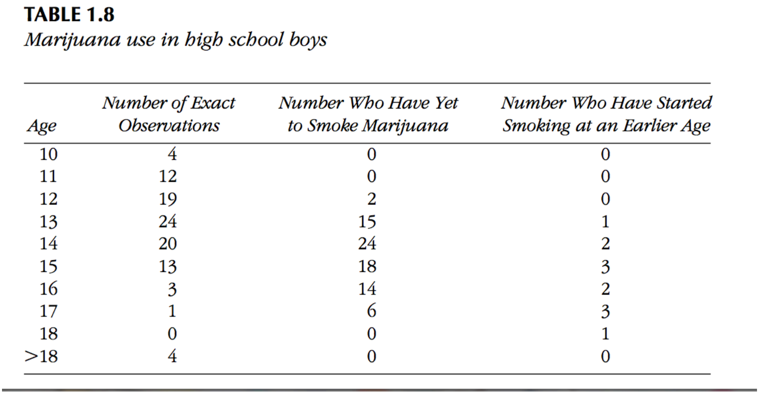


TABLE 1.8

Marijuana use in high school boys

Age	*Number of Exact Observations*	*Number Who Have Yet to Smoke Marijuana*	*Number Who Have Started Smoking at an Earlier Age*
10	4	0	0
11	12	0	0
12	19	2	0
13	24	15	1
14	20	24	2
15	13	18	3
16	3	14	2
17	1	6	3
18	0	0	1
>18	4	0	0

Marijuana example

Having both left- and right-censoring in the same data is a bit tricky to deal with. One approach is an iterative procedure that starts by ignoring the left-censored observations and starts with the product-limit estimator as the initial guess S_0 of the survival curve. The curve is then adjusted to deal with the left-censored observations.

For this algorithm, define the times to be t_i (these can include times where only censoring has been observed). Let c_i be the number of left-censored observations for time t_i and let r_i be the number of right-censored observations at time t_i. d_i is the number of events at time t_i. You can define $Y_i = \sum_{j:j \geq i} d_j + r_j$. Then define $S_0(t)$ as the usual product-limit estimator.

Marijuana example

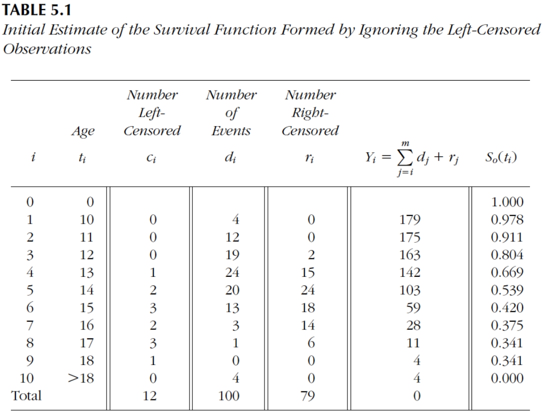


TABLE 5.1

Initial Estimate of the Survival Function Formed by Ignoring the Left-Censored Observations

i	*Age* t_i	*Number Left-Censored* c_i	*Number of Events* d_i	*Number Right-Censored* r_i	$Y_i = \sum_{j=i}^{m} d_j + r_j$	$S_o(t_i)$
0	0					1.000
1	10	0	4	0	179	0.978
2	11	0	12	0	175	0.911
3	12	0	19	2	163	0.804
4	13	1	24	15	142	0.669
5	14	2	20	24	103	0.539
6	15	3	13	18	59	0.420
7	16	2	3	14	28	0.375
8	17	3	1	6	11	0.341
9	18	1	0	0	4	0.341
10	>18	0	4	0	4	0.000
Total		12	100	79	0	

Marijuana example

The method is based on taking an initial estimate of the survival function, and using this to estimate the expected number of deaths (or events) at time t_i. So given $S_0(t)$, we re-estimate d_i as $\widehat{d}_i$ for each t_i. The new estimate of the d_i terms leads to another estimate of the survival function, $S_1(t)$. This in tern can be used to re-estimate the d_i terms. Reiterating back and forth, we get $S_1(t), S_2(t), \ldots, S_k(t), \ldots$, a sequence of estimated survival functions.

The sequence of survival functions should converge, so you can stop the algorithm once the estimated survival functions don't change much. This idea is based on the EM (Expectation-Maximization) algorithm which is used in other areas of statistics as well. The algorithm was formalized in a famous paper in 1977 (Dempster, Laird, and Rubin).

Intuitively, the idea is that if we don't know how many events occurred at time t_i (due to left-censoring), then we replace d_i with $E[d_i|S_k(t)]$, where $S_k(t)$ itself is estimated, so the expectation isn't exact. When we use these updated d_i values, we can get an improved estimate $S_{k+1}(t)$ of the survival function, so we can repeat the procedure to get an even better estimate of d_i, and so on.

Marijuana example

The algorithm can be described as

1. Let $k = 0$ and estimate the survival function $S_k(t)$ ignoring left-censored observations at times t_i.
2. Using the current estimate $S_k(t)$, estimate $p_{ij} = P[t_{j-1} < X \leq t_j | X \leq t_i]$ by $\frac{S_k(t_{j-1}) - S_k(t_j)}{1 - S_k(t_i)}$ for $j \leq i$.
3. Replace the current estimate d_i with $\widehat{d}_i = d_i + \sum_{i \geq j} c_i p_{ij}$
4. Compute the survival curve $S_k(t)$ based on only right-censored data but with the updated d_i values. If $S_k(t)$ is close to S_{k-1} (for $k > 1$), then stop. Otherwise, let $k = k + 1$ and go to step 2.

Computing p_{ij}

Note that $\sum_{j=1}^{i} p_{ij} = 1$, this gives the probability that the event occurred at time j given that it occurred some time at or before time j. So the p_{ij} give the probabilities of the possible earlier times when the event could have occurred. The numerators are successive differences in the survival function, and the denominators don't depend on j.

$$p_{41} = \frac{1.000 - 0.978}{1 - 0.669} = 0.067; \quad p_{42} = \frac{0.978 - 0.911}{1 - 0.669} = 0.202;$$

$$p_{43} = \frac{0.911 - 0.804}{1 - 0.669} = 0.320; \quad p_{44} = \frac{0.804 - 0.669}{1 - 0.669} = 0.410.$$

Computing p_{ij}: columns are i

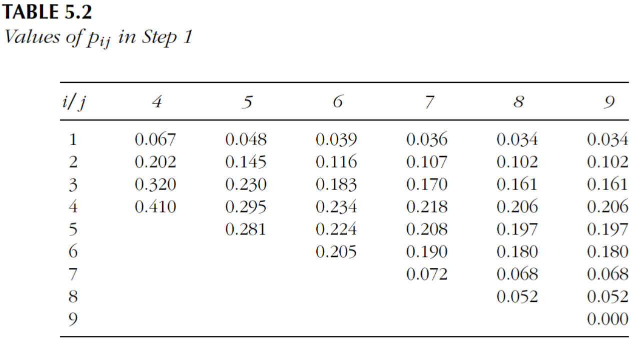


TABLE 5.2
Values of p_{ij} in Step 1

i/j	4	5	6	7	8	9
1	0.067	0.048	0.039	0.036	0.034	0.034
2	0.202	0.145	0.116	0.107	0.102	0.102
3	0.320	0.230	0.183	0.170	0.161	0.161
4	0.410	0.295	0.234	0.218	0.206	0.206
5		0.281	0.224	0.208	0.197	0.197
6			0.205	0.190	0.180	0.180
7				0.072	0.068	0.068
8					0.052	0.052
9						0.000

Updated survival function

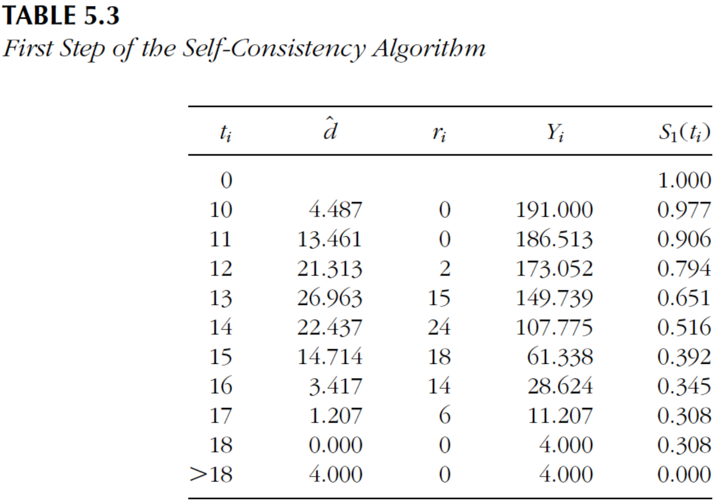


TABLE 5.3

First Step of the Self-Consistency Algorithm

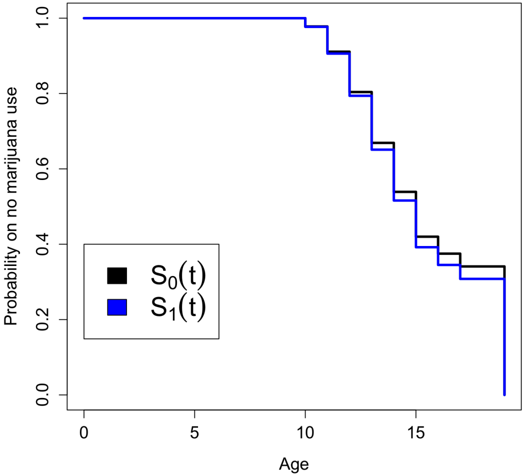
t_i	$\hat{d}$	r_i	Y_i	$S_1(t_i)$
0				1.000
10	4.487	0	191.000	0.977
11	13.461	0	186.513	0.906
12	21.313	2	173.052	0.794
13	26.963	15	149.739	0.651
14	22.437	24	107.775	0.516
15	14.714	18	61.338	0.392
16	3.417	14	28.624	0.345
17	1.207	6	11.207	0.308
18	0.000	0	4.000	0.308
>18	4.000	0	4.000	0.000

Updated survival function

The updated version is fairly close to the estimate ignoring left-censoring. We use the following R code to plot the two estimates simultaneously:

```
> x <- c(1,.978,.911,.804,.669,.539,.420,.375,.341,.341,0)
> x2 <- c(1,.977,.906,.794,.651,.516,.392,.345,.308,.308,0)
> t <- c(0,10,11,12,13,14,15,16,17,18,19)
> plot(t,x,xlab="Age",ylab="Probability on no marijuana use",
cex.axis=1.3,cex.lab=1.3,lwd=3,type="s")
> points(t,x2,lwd=3,type="s",col="blue")
> legend(0,.4,legend=c(expression(S[0](t)),expression(S[1](t)
)),fill=c("black","blue"),cex=2)
```

Updated survival function



Repeating the procedure

The book stops at one iteration, and the curves are fairly close, so you might think it is not important to iterate again. We'll illustrate iterating again since in other data sets it might be more important to iterate more than once. To iterate again, we need to compute a new set of p_{ij} values. For example, we now have

$$p_{41} = \frac{1.0 - 0.977}{1 - 0.651} = 0.066$$

$$p_{42} = \frac{0.977 - 0.906}{1 - 0.651} = 0.203$$

$$p_{43} = \frac{0.906 - 0.794}{1 - 0.651} = 0.321$$

$$p_{44} = \frac{0.794 - 0.651}{1 - 0.651} = 0.410$$

Computing pij

To fill in the matrix

```
> pij <- matrix(ncol=9,nrow=9)
> for(i in 1:9) {
+ for(j in 1:9) {
+ if(j <= i) pij[i,j] <- (S[j]-S[j+1])/(1-S[i+1])
+ else pij[i,j] <- 0
+ }
+ }
```

Repeating the procedure

```
> options(digits=3)
> t(pij)
      [,1]  [,2]  [,3]   [,4]   [,5]   [,6]   [,7]   [,8]   [,9]
 [1,]    1 0.245 0.112 0.0659 0.0475 0.0378 0.0351 0.0332 0.0332
 [2,]    0 0.755 0.345 0.2034 0.1467 0.1168 0.1084 0.1026 0.1026
 [3,]    0 0.000 0.544 0.3209 0.2314 0.1842 0.1710 0.1618 0.1618
 [4,]    0 0.000 0.000 0.4097 0.2955 0.2352 0.2183 0.2066 0.2066
 [5,]    0 0.000 0.000 0.0000 0.2789 0.2220 0.2061 0.1951 0.1951
 [6,]    0 0.000 0.000 0.0000 0.0000 0.2039 0.1893 0.1792 0.1792
 [7,]    0 0.000 0.000 0.0000 0.0000 0.0000 0.0718 0.0679 0.0679
 [8,]    0 0.000 0.000 0.0000 0.0000 0.0000 0.0000 0.0535 0.0535
 [9,]    0 0.000 0.000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
> colSums(t(pij))
[1] 1 1 1 1 1 1 1 1 1
```

Updating the $\widehat{d}_i$ values

```
> dHatOld <- c(4.487,13.461,21.313,26.963,
22.437,14.714,3.417,1.207,0,4)
> ci <- c(0,0,0,1,2,3,2,3,1) # left-censoring times
(these don’t change over iterations)
> for(i in 1:9) {
+ dhatNew[i] <- dHatOld[i] + sum(pij[i,]*ci)
+ }
> dHatOld
 [1]  4.49 13.46 21.31 26.96 22.44 14.71  3.42  1.21  0.00  4.00
> dHatNew # oops case-sensitive
Error: object ’dHatNew’ not found
> dhatNew
 [1]  4.49 13.46 21.31 27.37 23.29 16.01  4.76  2.64  1.43  4.00
```

So the $\widehat{d}$ values have changed a little bit.

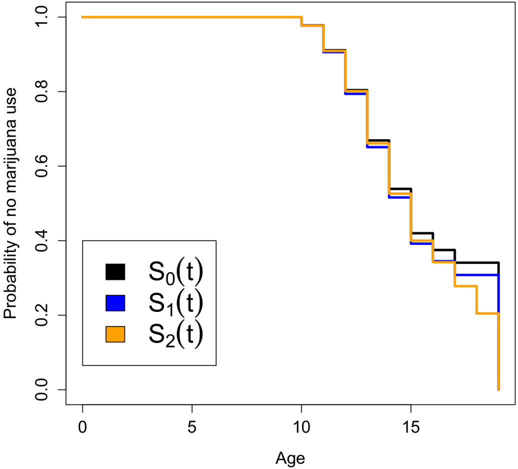
Computing the survival function

```
> ri <- c(0,0,2,15,24,18,14,6,0,0)
> Y <- (1:10)*0
> for(i in 1:10) {
+ for(j in i:10) {
+ Y[i] <- Y[i] + dhatNew[j] + ri[j]
+ }
+ }
> Y
 [1] 197.76 193.27 179.81 156.50 114.12 66.83 32.83 14.07
5.43 4.00
```

Computing the survival function

```
> S2 <- 1:11
> for(i in 2:11) {
+ S2[i] = S2[i-1]*(1-dhatNew[i-1]/Y[i-1])
+ }
> S2
 [1] 1.000 0.977 0.909 0.801 0.661 0.526 0.400 0.342 0.278 0.205 0.000
> S1
 [1] 1.000 0.977 0.906 0.794 0.651 0.516 0.392 0.345 0.308 0.308 0.000
> S0
 [1] 1.000 0.978 0.911 0.804 0.669 0.539 0.420 0.375 0.341 0.341 0.000
```

Updated survival function



Interval-censored data

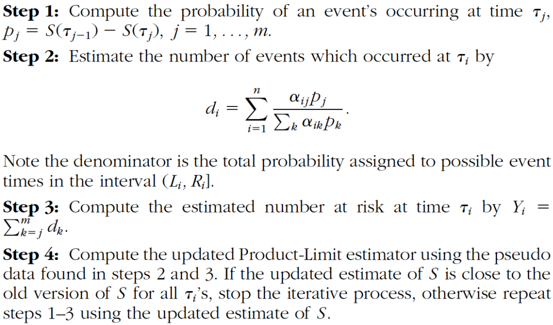
Interval-censored data can be dealt with in a similar way, making an initial guess about the survival function, and then iteratively computing updated estimated survival curves. The algorithm is outlined below. For the algorithm, an interval-censored observation has the form $(L_i, R_i]$, meaning that the event occurred after time L_i but before or at time R_i.

To deal with this case, let $\tau_0 < \tau_1 < \cdots < \tau_m$ be a set of time points that includes all L_i and R_i values. Let

$$\alpha_{ij} = \begin{cases} 1 & L_i \le \tau_{j-1} < \tau_j \le R_i \\ 0 & \text{otherwise} \end{cases}$$

Then if $\alpha_{ij} = 1$, then an event with time $(L_i, R_i]$ could have occurred at time τ_j.

Updated survival function



Step 1: Compute the probability of an event's occurring at time τ_j, $p_j = S(\tau_{j-1}) - S(\tau_j),\ j = 1, \ldots, m$.

Step 2: Estimate the number of events which occurred at τ_i by

$$d_i = \sum_{i=1}^{n} \frac{\alpha_{ij} p_j}{\sum_k \alpha_{ik} p_k}.$$

Note the denominator is the total probability assigned to possible event times in the interval $(L_i, R_i]$.

Step 3: Compute the estimated number at risk at time τ_i by $Y_i = \sum_{k=j}^{m} d_k$.

Step 4: Compute the updated Product-Limit estimator using the pseudo data found in steps 2 and 3. If the updated estimate of S is close to the old version of S for all τ_i's, stop the iterative process, otherwise repeat steps 1–3 using the updated estimate of S.