THE AMERICAN STATISTICIAN
2022, VOL. 00, NO. 0, 1–9: Teacher's Corner
https://doi.org/10.1080/00031305.2022.2070279

The Probability Mass Function of the Kaplan–Meier Product–Limit Estimator

Yuxin Qin, Heather Sasinowska, and Lawrence Leemis

Department of Mathematics, William & Mary, Williamsburg, VA

ABSTRACT

Kaplan and Meier's 1958 article developed a nonparametric estimator of the survivor function from a right-censored dataset. Determining the size of the support of the estimator as a function of the sample size provides a challenging exercise for students in an advanced course in mathematical statistics. We devise two algorithms for calculating the support size and calculate the associated probability mass function for small sample sizes and particular probability distributions for the failure and censoring times.

ARTICLE HISTORY
Received March 2022
Accepted April 2022

KEYWORDS
Censoring; Induction; Nonparametric estimation; Probability mass function; Survival analysis; Survivor function

1. Introduction

Kaplan and Meier's 1958 article which established the nonparametric product–limit estimator (Kaplan and Meier 1958) is the most oft-cited article in the statistics literature (Noorden, Maher, and Nuzzo 2014). It is a nonparametric estimate of the survival function from a dataset of lifetimes that includes right-censored observations. The Kaplan–Meier product–limit estimator (KMPLE) is used in a variety of application areas.

- In reliability, the object of interest is a component or a system and the event of interest is failure.
- In biostatistics, the object of interest is often a patient and the event of interest might be death or the conclusion of remission.
- In actuarial science, the object of interest is often the insured (for life insurance) or property (for casualty insurance) and the associated event of interest is death (for life insurance) or claim (for casualty insurance).

For simplicity, we will refer to the object of interest generically as the *item* and the event of interest as the *failure*.

Let n denote the number of items on test. The KMPLE of the survival function $S(t)$ is given by

$$\hat{S}(t) = \prod_{i:t_i \le t} \left(1 - \frac{d_i}{n_i}\right),$$

for $t \ge 0$, where $t_1, t_2, \ldots, t_k$ are the times when at least one failure is observed (k is an integer between 1 and n, which is the number of distinct failure times in the dataset), $d_1, d_2, \ldots, d_k$ are the number of failures observed at times $t_1, t_2, \ldots, t_k$, and $n_1, n_2, \ldots, n_k$ are the number of items at risk just prior to times $t_1, t_2, \ldots, t_k$. It is common practice to have the KMPLE "cut off" after the largest time recorded if it corresponds to a right-censored observation (Kalbfleisch and Prentice 2002, p. 16). The KMPLE drops to zero after the largest time recorded if it is a failure; the KMPLE is undefined, however, after the largest time recorded if it is a right-censored observation. This convention will be followed in this article.

As a particular instance, consider the KMPLE when there are $n = 4$ items on test, failures occur at times $t = 1$ and $t = 3$, and right censorings occur at times $t = 2$ and $t = 4$. In this setting, the KMPLE is

$$\hat{S}(t) = \begin{cases} 1 & 0 \le t < 1 \\ \left(1 - \frac{1}{4}\right) = \frac{3}{4} & 1 \le t < 3 \\ \left(1 - \frac{1}{4}\right)\left(1 - \frac{1}{2}\right) = \frac{3}{8} & 3 \le t < 4 \\ \text{NA} & t \ge 4, \end{cases}$$

where NA indicates that the KMPLE is undefined. This example illustrates that the effect of right censoring is to selectively remove factors in the product

$$\left(1 - \frac{1}{4}\right)\left(1 - \frac{1}{3}\right)\left(1 - \frac{1}{2}\right)(1 - 1)$$

when determining the possible values (support) of $\hat{S}(t)$ for $n = 4$.

The goal of this article is to determine the support values of the KMPLE and their associated probabilities for a given value of n. Section 2 gives two algorithms for calculating the support values. The probability mass function of $\hat{S}(t)$ for one particular failure time distribution, right-censoring time distribution, and the time value of interest is calculated in Section 3. Section 4 contains conclusions and outlines further work.

CONTACT Lawrence Leemis leemis@math.wm.edu Department of Mathematics, William & Mary, Williamsburg, VA.

2. Determining Support Values and Support Size

Determining the support of the KMPLE as a function of n is nontrivial. The next two sections contain algorithms for enumerating the support values and the associated support size.

2.1. Kaplan and Meier Meet Pascal

Consider the case of n items on test and distinct failure times (i.e., $d_1 = d_2 = \cdots = d_k = 1$). When all failure times are observed, the survival function estimate after the last observed failure is

$$\hat{S}(t) = \left(1 - \frac{1}{n}\right)\left(1 - \frac{1}{n-1}\right)\cdots\left(1 - \frac{1}{2}\right)(1-1),$$

which is zero. The KMPLE at any fixed time value consists of up to n of these factors, which are denoted by $f_1, f_2, \ldots, f_n$, where $f_i = 1 - 1/(n-i+1)$ for $i = 1, 2, \ldots, n$. The support of the KMPLE can be written as a product of some permutation of these factors:

$$\hat{S}(t) \in \left\{ \prod_{i \in P} f_i; P \in P_n \right\}$$

for all times $t \geq 0$, where P_n is the set of all permutations of $\{1, 2, \ldots, n\}$. The effect on the KMPLE of right censoring is to remove factors in the product that are associated with right-censored observations. In the example from Section 1 with $n = 4$, for instance, the second factor, $f_2 = (1 - \frac{1}{3})$, was eliminated from the product because the second item was censored at $t = 2$. Using counting methods we can calculate the number of possible combinations of factors used in the calculation for $\hat{S}(t)$ for x right-censored observations for $x = 0, 1, \ldots, n-1$. Disregarding the last factor and the case in which the last item is right-censored, there are $\binom{n-1}{x}$ ways to combine the first $n-1$ factors after eliminating the x factors associated with the right-censored observations. The number of permutations of possible factors corresponds to a row in Pascal's triangle.

There are two extreme cases. In the first case, $x = 0$ corresponds to a complete dataset with no right-censored data values, and therefore, no factors are eliminated in the calculation. In the second case, $x = n-1$ corresponds to no observed failures in the first $n-1$ observations, and therefore all factors are eliminated. In this special case there is one failure time observed after the $n-1$ right-censored items.

To illustrate, consider the case of $n = 4$ items on test. We consider only the first $n - 1 = 3$ factors f_1, f_2, and f_3. The last factor, f_4, will generate either a support value of zero (if the largest recorded time corresponds to a failure) or an NA (if the largest recorded time corresponds to a right censoring). Table 1 contains all the possible support values corresponding to $x = 0, 1, 2, 3$. The first column gives the number of right-censored observations, $x = 0, 1, 2, 3$. The second column gives the associated values of $\binom{n-1}{x}$. This corresponds to the number of ways that the x censored observations can be ordered among the failure times, since we temporarily ignored the last factor in the KMPLE $\left[\hat{S}(t) = 0\right]$ and the case in which the last item is right-censored $\left[\hat{S}(t) = \text{NA}\right]$. The values in this column are the values in the fourth row of Pascal's triangle. The third column lists the potential factors for the KMPLE. The fourth column lists the possible cumulative products. These cumulative products are the support values associated with a particular ordering of failure and right censoring times. These values are sorted and displayed in lowest terms in the fifth column. We

Table 1. All possible KMPLE support values for $n = 4$.

Number of censored observations	$\binom{n-1}{x}$	KMPLE factors	Support value calculations	Possible support values
$x = 0$	1	$\left(1-\frac{1}{4}\right), \left(1-\frac{1}{3}\right), \left(1-\frac{1}{2}\right)$	$\left(1-\frac{1}{4}\right)$	$1, \frac{3}{4}, \frac{1}{2}, \frac{1}{4}, 0$
			$\left(1-\frac{1}{4}\right)\left(1-\frac{1}{3}\right)$	
			$\left(1-\frac{1}{4}\right)\left(1-\frac{1}{3}\right)\left(1-\frac{1}{2}\right)$	
$x = 1$	3	$\left(1-\frac{1}{4}\right), \left(1-\frac{1}{3}\right)$	$\left(1-\frac{1}{4}\right)$	$1, \frac{3}{4}, \frac{2}{3}, \frac{1}{2}, \frac{3}{8}, \frac{1}{3}, 0, \text{NA}$
			$\left(1-\frac{1}{4}\right)\left(1-\frac{1}{3}\right)$	
		$\left(1-\frac{1}{4}\right), \left(1-\frac{1}{2}\right)$	$\left(1-\frac{1}{4}\right)$	
			$\left(1-\frac{1}{4}\right)\left(1-\frac{1}{2}\right)$	
		$\left(1-\frac{1}{3}\right), \left(1-\frac{1}{2}\right)$	$\left(1-\frac{1}{3}\right)$	
			$\left(1-\frac{1}{3}\right)\left(1-\frac{1}{2}\right)$	
$x = 2$	3	$\left(1-\frac{1}{4}\right)$	$\left(1-\frac{1}{4}\right)$	$1, \frac{3}{4}, \frac{2}{3}, \frac{1}{2}, 0, \text{NA}$
		$\left(1-\frac{1}{3}\right)$	$\left(1-\frac{1}{3}\right)$	
		$\left(1-\frac{1}{2}\right)$	$\left(1-\frac{1}{2}\right)$	
$x = 3$	1	No factors		$1, 0, \text{NA}$

include a zero, a one, and NA (for $x > 0$) after all other support values have been determined. The one is included as $\hat{S}(t) = 1$ for all $t < t_1$ before any failure times occur. The zero is included for the last factor that we temporarily ignored and occurs after the last (nth) item fails. One additional special case occurs when the last (nth) item is right censored. It results in $\hat{S}(t)$ being undefined. For $x > 0$, we have included NAs to represent this case. Removing the duplicate values and NAs from the rightmost column of Table 1 gives the set of support values for KMPLE when there are $n = 4$ items on test: $\{1, \frac{3}{4}, \frac{2}{3}, \frac{1}{2}, \frac{3}{8}, \frac{1}{3}, \frac{1}{4}, 0\}$.

The computations associated with $n = 4$ given in the previous paragraph have shown that there are 8 possible defined support values on $[0, 1]$ (not including NA as a support value) for the KMPLE associated with $n = 4$ items on test. This example exposes some computational issues in terms of both speed and memory for larger values of n. In addition to the number of combinations from a row of Pascal's triangle, the associated cumulative products must be calculated, the resulting fractions must be converted to lowest terms, and the duplicate fractions must be eliminated, requiring significant computer time and memory as n increases.

2.2. Induction Algorithm

The computational issues associated with the previous algorithm prompted us to consider an algorithm based on induction. We continue to assume that all failure times are distinct. Two key observations are necessary to devise the induction algorithm.

1. A KMPLE support value associated with $n - 1$ items on test must necessarily be a KMPLE support value for n items on test. The rationale for this observation is the fact that the KMPLE simply adds one new factor for each additional item on test. The factors associated with $n - 1$ items on test, $f_1, f_2, \ldots, f_{n-1}$, are all present in the case of n items on test.
2. Let χ_{n-1} be the set of KMPLE support values associated with $n - 1$ items on test. Let $|\chi_{n-1}|$ be the cardinality of χ_{n-1}. The set of KMPLE support values associated with n items on test is the union of χ_{n-1} and the set consisting of the elements of χ_{n-1} multiplied by $(1 - \frac{1}{n}) = \frac{n-1}{n}$, the first factor in the KMPLE.

Determining χ_n via induction for the first few values of n is described below.

- When $n = 1$, the set of support values for $\hat{S}(t)$ is $\chi_1 = \{1, 0\}$.
- When $n = 2$, in addition to the existing support values for $n = 1$, the potential new support values can be calculated by multiplying $(1 - \frac{1}{2})$ by the existing support value set, which is $\chi_1 = \{1, 0\}$, resulting in a support value set for $n = 2$: $\chi_2 = \{1, \frac{1}{2}, 0\}$.
- When $n = 3$, we repeat the calculation above by multiplying $(1 - \frac{1}{3})$ by the existing support value set $\chi_2 = \{1, \frac{1}{2}, 0\}$, which gives the result $\chi_3 = \{1, \frac{2}{3}, \frac{1}{2}, \frac{1}{3}, 0\}$.

In this way, to generate the support value set for a particular n, we multiply $(1 - \frac{1}{n})$ by the support value set for $n - 1$, and then add the new values to the support set. Note that duplicate values in addition to 0 will emerge during the process when we calculate χ_n for larger n. Those duplicate values should be removed to get the final support value set.

Table 2. Number of support values on [0, 1] for the KMPLE for n from 1 to 40.

n	$\lvert\chi_n\rvert$	n	$\lvert\chi_n\rvert$	n	$\lvert\chi_n\rvert$	n	$\lvert\chi_n\rvert$
1	2	11	409	21	76889	31	10275645
2	3	12	681	22	115397	32	16487301
3	5	13	1361	23	230793	33	22679853
4	8	14	2307	24	383753	34	33790243
5	15	15	3597	25	536994	35	48842489
6	25	16	5088	26	820907	36	60737510
7	49	17	10175	27	1189517	37	121475019
8	83	18	16711	28	1597245	38	204647341
9	134	19	33421	29	3194489	39	303830465
10	205	20	55211	30	5137823	40	391169317

The Maple computer algebra system is particularly well suited for implementing the induction algorithm because the language includes a data structure for sets which automatically eliminates duplicate support values. The Maple code used to compute the support values (excluding zero) for $n = 1, 2, \ldots, 40$ follows.

```
support := {1};
for n from 2 to 40 do
  support := support union
    {(n - 1) / n * op(support)}:
od:
```

Table 2 shows the number of the support values for $\hat{S}(t)$ for n items on test, which is computed by including the additional statement `print(nops(support) + 1);` at the bottom of the `for` loop. Adding one is to account for the value of 0 in χ_n. Table 2 only includes support values on [0, 1]; NA is not included in the counts. The appendix contains R code that implements the induction algorithm.

As observed from Table 2, every time n is incremented from a composite number to a prime number, $|\chi_n|$ almost doubles. In particular, $|\chi_{n+1}| = 2|\chi_n| - 1$. Our induction process shows why this happens. When we multiply the existing support values (which are all unique) in the set by the new initial factor f_1, which has a prime denominator, the updated support values will all be distinct from the existing values, except for 0. This is discussed in more detail in Section 2.3.

Figure 1 displays the support values for $n = 1, 2, \ldots, 8$. We observe that as n increases, there is a trend that more support values are added below $\frac{1}{2}$ than above $\frac{1}{2}$. In the figure, blue points are "predictable" values (i.e., $\frac{x}{n}$, where $x = 0, 1, \ldots, n$). The red points are what we call the "unpredictable" values. The first unpredictable value is $\left(1 - \frac{1}{4}\right)\left(1 - \frac{1}{2}\right) = \frac{3}{8}$, which occurs at $n = 4$. Unpredictable values will always be due to censoring, although not all censored data points will cause unpredictable values. The gray line depicts the average of the support values. The blue lines show the symmetric envelope of the support values that extends from $\frac{1}{n}$ to $\frac{n-1}{n}$.

2.3. Percent Increase in Support Size

After calculating the support sizes for $n = 1, 2, \ldots, 40$, we explored the percent increase in the support size as n increases by 1. As illustrated in Figure 2, the highest percent increases

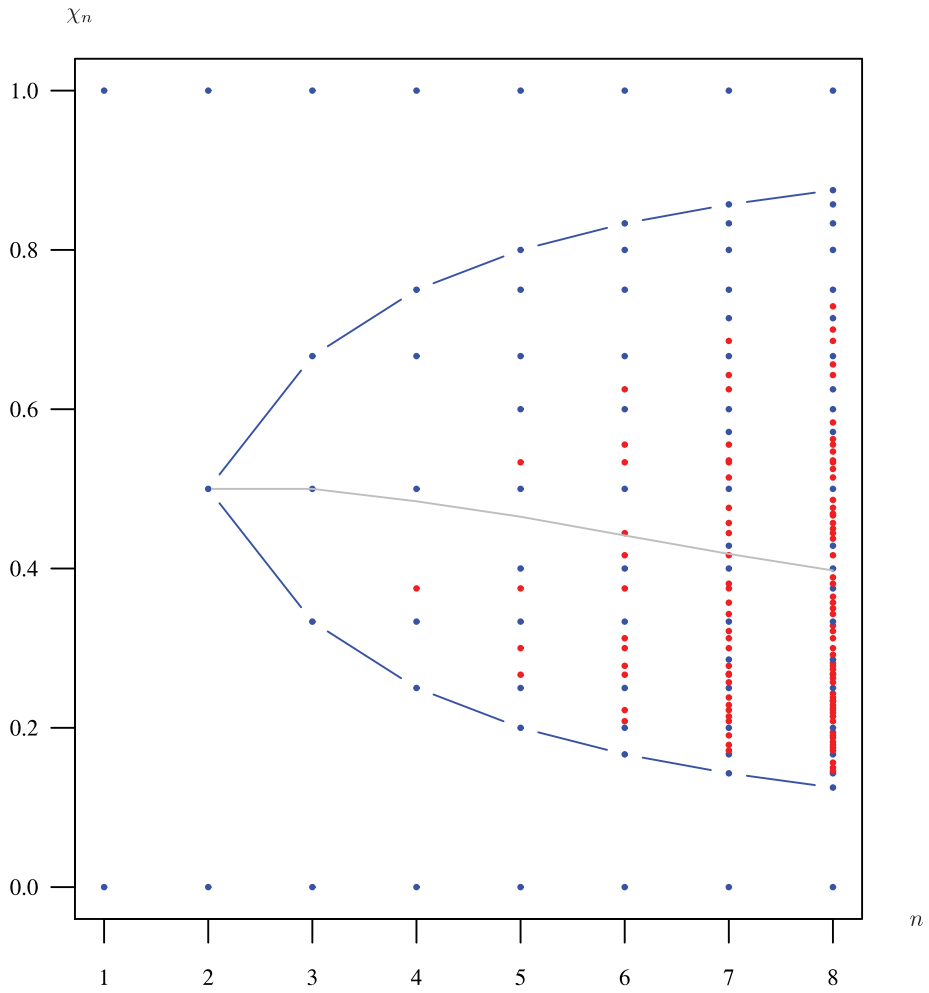

Figure 1. Support values for KMPLE for n from 1 to 8. The blue points are x/n, where $x = 0, 1, \ldots, n$ and all other support values are red dots. The blue lines give the envelope of fractional support values and the gray line gives the average of the support values.

occur at prime numbers due to the fact that the new support values for n will be distinct from the original support set for $n-1$, except for $\hat{S}(t) = 0$. Therefore, the dotted segments at the top of the graph show that the highest percent increases converge to 100%. Also, notice that the percent increases at the even numbers immediately following the primes are significantly higher than the other composite values.

Using the induction algorithm, we know that

$$\chi_{n+1} = \left\{\frac{n}{n+1} \cdot \chi_n\right\} \cup \chi_n,$$

where $\{\frac{n}{n+1} \cdot \chi_n\}$ denotes the set of support values in χ_n, each multiplied by $\frac{n}{n+1}$. There is a special relationship that exists between $|\chi_n|$ and $|\chi_{n+1}|$ when $n+1$ is prime. Consider the case of $n = 6$. Temporarily ignoring the support values of 0 and NA by suppressing the final factor f_6, we know that χ_6 consists of the products of all combinations of the factors

$$\begin{aligned} f_1 f_2 f_3 f_4 f_5 &= \left(1-\frac{1}{6}\right)\left(1-\frac{1}{5}\right)\left(1-\frac{1}{4}\right)\left(1-\frac{1}{3}\right)\left(1-\frac{1}{2}\right) \\ &= \frac{5}{6} \cdot \frac{4}{5} \cdot \frac{3}{4} \cdot \frac{2}{3} \cdot \frac{1}{2} \end{aligned}$$

once duplicate support values (for example, $\frac{2}{5}$ can be obtained as $\frac{4}{5} \cdot \frac{3}{4} \cdot \frac{2}{3}$ or as $\frac{4}{5} \cdot \frac{1}{2}$) have been removed, and 0 has been added. Since $n+1 = 7$ is prime, the set $\chi_7 = \left\{\frac{6}{7} \cdot \chi_6\right\} \cup \chi_6$ consists of 0 and the products of all combinations of the factors

$$f_1 f_2 f_3 f_4 f_5 f_6 = \frac{6}{7} \cdot \frac{5}{6} \cdot \frac{4}{5} \cdot \frac{3}{4} \cdot \frac{2}{3} \cdot \frac{1}{2}.$$

Notice that the 7 in the denominator of the first factor, f_1, is prime and therefore will not have any cancellation with any of the numerators. Zero will be the only duplicate value in the two support sets χ_6 and χ_7. This means that in the general case when $n+1$ is prime,

$$\left\{\frac{n}{n+1} \cdot \chi_n\right\} \cap \chi_n = \{0\}$$

so that

$$|\chi_{n+1}| = 2|\chi_n| - 1$$

and the associated percent increase is

$$\frac{2|\chi_n| - 1 - |\chi_n|}{|\chi_n|} \cdot 100\% = \frac{|\chi_n| - 1}{|\chi_n|} \cdot 100\%.$$

This maximum possible percent increase is reflected in Figure 2 as a dotted line. This maximum possible percent increase is only achieved when there is a prime number of items on test.

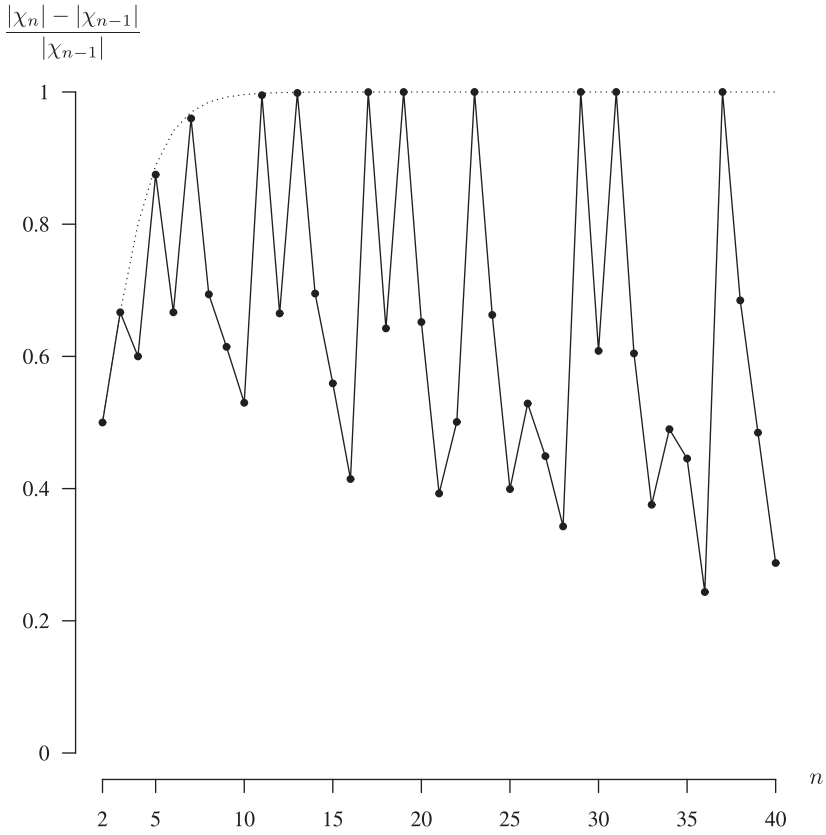

Figure 2. Increase in support size for $n = 2, 3, \ldots, 40$.

2.4. Tied Observations

So far, we have assumed that the failure times and censoring times are all distinct. Since time is continuous, the probability that two events occur at the same time is 0. However, in some applications, tied observations can occur. In survival analysis, for example, the survival time of patients might be measured in days; therefore, multiple patients could die or leave the study on the same day. In this section, we will show that the case of tied observations will not affect our algorithms that generate the KMPLE support values.

There are three possibilities for tied observations: the tied observations are a mix of failure and censoring times, all tied observations are censoring times, and all tied observations are failure times.

The first case is one or more failure time(s) being tied with one or more censoring time(s) at time t. In this case, we follow the convention of assuming that the failure times are treated as occurring just prior to time t, while the censoring times are treated as occurring slightly after t (Kaplan and Meier, 1958, p. 461). Therefore, the calculation of KMPLE will be the same as if the failure times and the censoring times are distinct.

The second case is two or more censoring times being tied at time t. Since the KMPLE is calculated only when a failure is observed, not when an item on test is censored, these censored items will not be taken into account until the next failure time after t during the calculation. It is the same case as having the same number of distinct and consecutive censoring times before the next observed failure and, therefore, does not affect our algorithm.

Finally, when two or more failure times are tied at time t, we illustrate below that it is effectively equivalent to the same number of items failing close together.

To illustrate, consider the case of $n = 6$ with the following observations:

- at time t_1, a failure is observed and an item is censored,
- at time t_2, two failures are observed, and
- at time t_3, a failure is observed and an item is censored,

where $0 < t_1 < t_2 < t_3$. Before t_1, no failures are observed, so the KMPLE on the time interval $0 \le t < t_1$ is $\hat{S}(t) = 1$.

At time t_1, we use the convention of treating the first failure time as occurring just prior to t_1 and the first censoring time as occurring slightly after t_1. So $\hat{S}(t) = \left(1 - \frac{d_1}{n_1}\right)$, where the number of failures at time t_1 is $d_1 = 1$, and the number of items at risk just prior to time t_1 is $n_1 = 6$. The KMPLE on $t_1 \le t < t_2$ is

$$\hat{S}(t) = \left(1 - \frac{1}{6}\right) = \frac{5}{6}.$$

At time t_2, $\hat{S}(t) = \left(1 - \frac{1}{6}\right)\left(1 - \frac{d_2}{n_2}\right)$. Since there are two failures observed at time t_2, $d_2 = 2$, and since one failure was observed and one item was censored before t_2, the number of items at risk just prior to time t_2 is $n_2 = 6 - 2 = 4$. The KMPLE on $t_2 \le t < t_3$ is

$$\hat{S}(t) = \left(1 - \frac{1}{6}\right)\left(1 - \frac{2}{4}\right) = \frac{5}{6} \cdot \frac{2}{4} = \frac{5}{12}.$$

Now, we compare $\hat{S}(t)$ with the case in which those two failure times are not tied. Let one of the two failures be observed at time $t_2 - \epsilon$, where $0 < \epsilon < t_2 - t_1$, while the other failure is still observed at time t_2. The KMPLE values are calculated as follows:

$$\begin{aligned}\hat{S}(t) &= \left(1-\frac{1}{6}\right)\left(1-\frac{1}{4}\right)\\ &= \frac{5}{6}\cdot\frac{3}{4} = \frac{5}{8} \qquad (t_2-\epsilon \le t < t_2),\\ \hat{S}(t) &= \left(1-\frac{1}{6}\right)\left(1-\frac{1}{4}\right)\left(1-\frac{1}{3}\right)\\ &= \frac{5}{6}\cdot\frac{3}{4}\cdot\frac{2}{3} = \frac{5}{12} \qquad (t_2 \le t < t_3).\end{aligned}$$

Notice that $\hat{S}(t) = \frac{5}{12}$ in the case of tied observations also occurs in the case of distinct observations. The factor $\left(1-\frac{2}{4}\right)$ in the calculation of KMPLE at t_2 in the former case has the same effect as the factors $\left(1-\frac{1}{4}\right)\left(1-\frac{1}{3}\right)$ in the latter case. As ϵ goes to zero, the value $\frac{5}{8}$ will be excluded from the support of KMPLE in this particular case. At time t_3, following the convention again, for the observed failure we get the KMPLE for $t = t_3$:

$$\hat{S}(t) = \left(1-\frac{1}{6}\right)\left(1-\frac{2}{4}\right)\left(1-\frac{1}{2}\right) = \frac{5}{6}\cdot\frac{1}{2}\cdot\frac{1}{2} = \frac{5}{24}.$$

However, the last item is censored, and since we treat the censored item as occurring slightly after t_3, the KMPLE is undefined (NA) for $t > t_3$.

Finally, the realized KMPLE values in this particular illustration are $\{1, \frac{5}{6}, \frac{5}{12}, \frac{5}{24}, \text{NA}\}$, which is a subset of the full support for KMPLE for $n = 6$ when we have distinct failure and censoring times:

$$\left\{1, \frac{5}{6}, \frac{4}{5}, \frac{3}{4}, \frac{2}{3}, \frac{5}{8}, \frac{3}{5}, \frac{5}{9}, \frac{8}{15}, \frac{1}{2}, \frac{4}{9}, \frac{5}{12}, \frac{2}{5}, \frac{3}{8}, \frac{1}{3}, \frac{5}{16}, \frac{3}{10}, \frac{5}{18}, \frac{4}{15}, \frac{1}{4}, \frac{2}{9}, \frac{5}{24}, \frac{1}{5}, \frac{1}{6}, 0, \text{NA}\right\}.$$

In general, tied failure and censoring times, as well as tied censoring times, have no effect on the calculation of the KMPLE, while tied failure times will only remove support values instead of adding new ones. Therefore, our algorithms for generating the support values of the KMPLE can still be applied to situations in which there are tied observations.

3. Probability Mass Function of the KMPLE

Given a set of support values of the KMPLE for a particular number of items on test n, we devised an algorithm to generate the probability mass function associated with the support. In particular, given a fixed time t_0, we are interested in the probability mass function of the random variable $S = \hat{S}(t_0)$, where $S \in \chi_n$. There are an infinite number of settings of

- the number of items on test,
- the probability distribution of the failure times,
- the probability distribution of the censoring times, and
- the fixed time of interest t_0.

In this section, we choose one particular setting of these elements in order to illustrate the calculation of the probability mass function.

The notation associated with the algorithm is established in this paragraph.

- Let n be the number of items on test.
- Let the random failure times be $T_1, T_2, \ldots, T_n$.
- Let the random censoring times be $C_1, C_2, \ldots, C_n$.
- Let $X_i = \min\{T_i, C_i\}$, for $i = 1, 2, \ldots, n$. The observed values of $X_1, X_2, \ldots, X_n$ are denoted by $x_1, x_2, \ldots, x_n$.
- Let

$$\delta_i = \begin{cases} 1 & \text{if the } i\text{th item failed (that is, } T_i \le C_i) \\ 0 & \text{if the } i\text{th item is censored (that is, } C_i < T_i) \end{cases}$$

for $i = 1, 2, \ldots, n$.

- Let $t_0 \ge 0$ be the fixed time of interest.
- Let $S = \hat{S}(t_0)$ be the random KMPLE whose probability distribution is desired. The support of this random variable is χ_n, which can be determined by one of the two algorithms in the previous section.

The failure times and the censoring times are assumed to be continuous random variables with positive support. These are common assumptions in survival analysis.

Without loss of generality, assume that $X_1 < X_2 < \cdots < X_n$. There are $n+1$ time intervals before, after, and in between the $X_1, X_2, \ldots, X_n$ values in which t_0 can fall. Denote the interval into which t_0 falls by l, where $l = 0, 1, \ldots, n$. The first interval, $[0, X_1)$, corresponds to $l = 0$. The second interval, $[X_1, X_2)$, corresponds to $l = 1$. The last interval, $[X_n, \infty)$, corresponds to $l = n$.

This paragraph describes the five steps in the algorithm to compute the probability mass function of the KMPLE. First, the support χ_n is determined using the induction algorithm. The mass values associated with each support value are initialized to zero. The second step in the algorithm is to calculate the probability that $X_l \le t_0 < X_{l+1}$, for $l = 0, 1, \ldots, n$, where $X_0 = 0$ and $X_{n+1} = \infty$ in order to accommodate the extreme intervals. Let $f_{X_l, X_{l+1}}(x_l, x_{l+1})$ denote the joint probability density function of the adjacent observations X_l and X_{l+1}. For the fixed time t_0, the probability that the random interval $[X_l, X_{l+1})$ contains t_0, for $l = 0, 1, \ldots, n$, is

$$P(X_l \le t_0 < X_{l+1}) = \int_0^{t_0}\int_{t_0}^{\infty} f_{X_l, X_{l+1}}(x_l, x_{l+1})\, dx_{l+1}\, dx_l.$$

In the third step of the algorithm, all possible sequences of $\delta_1, \delta_2, \ldots, \delta_l$ are identified for $l = 0, 1, \ldots, n$. Interval l has 2^l possible sequences of $\delta_1, \delta_2, \ldots, \delta_l$ by the multiplication rule. The fourth step of the algorithm is to calculate the probabilities associated with each sequence of $\delta_1, \delta_2, \ldots, \delta_l$ identified in the third step of each algorithm. The final step is to accumulate these probabilities into the probability mass function values for the appropriate support value.

Table 3. Calculations for finding the probability mass function of the KMPLE at $t_0 = -\ln(1/2)/2$ for $n = 3$, $T_i \sim$ exponential(1), and $C_i \sim$ exponential(1), for $i = 1, 2, 3$.

l	$P(X_l \le t_0 < X_{l+1})$	$\delta_1, \delta_2, \ldots, \delta_l$	$\hat{S}(t_0) = s$	$P(\hat{S}(t_0) = s, X_l \le t_0 < X_{l+1})$
$l = 0$	$\frac{1}{8}$	—	1	$\frac{1}{8}$
$l = 1$	$\frac{3}{8}$	$\delta_1 = 0$	1	$\frac{3}{8} \cdot \frac{1}{2} = \frac{3}{16}$
		$\delta_1 = 1$	$\frac{2}{3}$	$\frac{3}{8} \cdot \frac{1}{2} = \frac{3}{16}$
$l = 2$	$\frac{3}{8}$	$(\delta_1, \delta_2) = (0, 0)$	1	$\frac{3}{8} \cdot \frac{1}{4} = \frac{3}{32}$
		$(\delta_1, \delta_2) = (0, 1)$	$\frac{1}{2}$	$\frac{3}{8} \cdot \frac{1}{4} = \frac{3}{32}$
		$(\delta_1, \delta_2) = (1, 0)$	$\frac{2}{3}$	$\frac{3}{8} \cdot \frac{1}{4} = \frac{3}{32}$
		$(\delta_1, \delta_2) = (1, 1)$	$\frac{1}{3}$	$\frac{3}{8} \cdot \frac{1}{4} = \frac{3}{32}$
$l = 3$	$\frac{1}{8}$	$(\delta_1, \delta_2, \delta_3) = (0, 0, 1)$		
		$(\delta_1, \delta_2, \delta_3) = (0, 1, 1)$	0	$\frac{1}{8} \cdot \frac{4}{8} = \frac{1}{16}$
		$(\delta_1, \delta_2, \delta_3) = (1, 0, 1)$		
		$(\delta_1, \delta_2, \delta_3) = (1, 1, 1)$		
		$(\delta_1, \delta_2, \delta_3) = (0, 0, 0)$		
		$(\delta_1, \delta_2, \delta_3) = (0, 1, 0)$	NA	$\frac{1}{8} \cdot \frac{4}{8} = \frac{1}{16}$
		$(\delta_1, \delta_2, \delta_3) = (1, 0, 0)$		
		$(\delta_1, \delta_2, \delta_3) = (1, 1, 0)$		

We will illustrate the algorithm with the case of $n = 3$ items on test. There is an infinite number of choices for the failure time distributions, right-censoring time distributions, and t_0 values. One set of these is illustrated here in order to demonstrate the process of finding the probability mass function of the KMPLE. We make the following further assumptions associated with a random censoring scheme:

- $T_1, T_2, \ldots, T_n$ are iid exponential(1) failure times,
- $C_1, C_2, \ldots, C_n$ are iid exponential(1) right-censoring times, and
- $t_0 = -\ln(1/2)/2$ is the time value of interest (t_0 is the median of an exponential(2) random variable).

With these assumptions, it is equally likely that an item on test is observed to fail or be right censored because the two exponential distributions have the same rate parameter. In other words, $P(\delta_i = 0) = P(\delta_i = 1) = 1/2$ for $i = 1, 2, \ldots, n$. Furthermore, since the minimum of two independent exponential random variables is also exponential, $X_i = \min\{T_i, C_i\} \sim$ exponential(2) for $i = 1, 2, \ldots, n$. So choosing the median of an exponential(2) random variable for t_0 means that X_i is equally likely to fall to the left of t_0 or to the right of t_0.

To implement the algorithm for determining the probability distribution of the KMPLE at t_0, the first step is to determine the support using the induction algorithm, yielding $\chi_3 = \left\{0, \frac{1}{3}, \frac{1}{2}, \frac{2}{3}, 1\right\}$. The second step is to calculate the probability that t_0 will fall in the random interval $[X_l, X_{l+1})$ by calculating the appropriate double integral, yielding

$$P(0 \le t_0 < X_1) = \frac{1}{8}, \quad P(X_1 \le t_0 < X_2) = \frac{3}{8},$$
$$P(X_2 \le t_0 < X_3) = \frac{3}{8}, \quad P(t_0 \ge X_3) = \frac{1}{8}.$$

The third step is to identify the potential sequences of $\delta_1, \delta_2, \ldots, \delta_l$ for interval l. These sequences are given in the third column of Table 3. The fourth step is to calculate the survivor function values and probabilities associated with each of these values, which are shown in the fourth and fifth columns of Table 3. As a particular instance of part of the calculation of one of the relevant probabilities in the fifth column,

$$P\left(\hat{S}(t_0) = \frac{1}{3} \,\middle|\, x_2 \le t_0 < x_3\right) = \frac{1}{4}.$$

The fifth and final step in the algorithm is to sum the probabilities associated with each of the support values for $l = 0, 1, 2, 3$, in order to generate the following probability mass function of support values for the KMPLE when

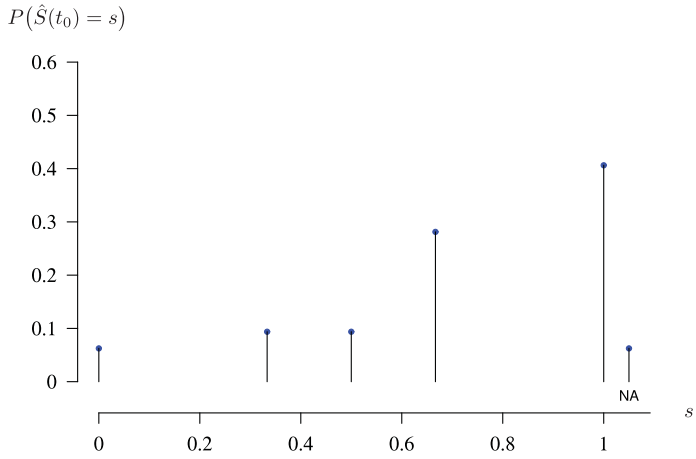

Figure 3. Probability mass function of the KMPLE for $n = 3$.

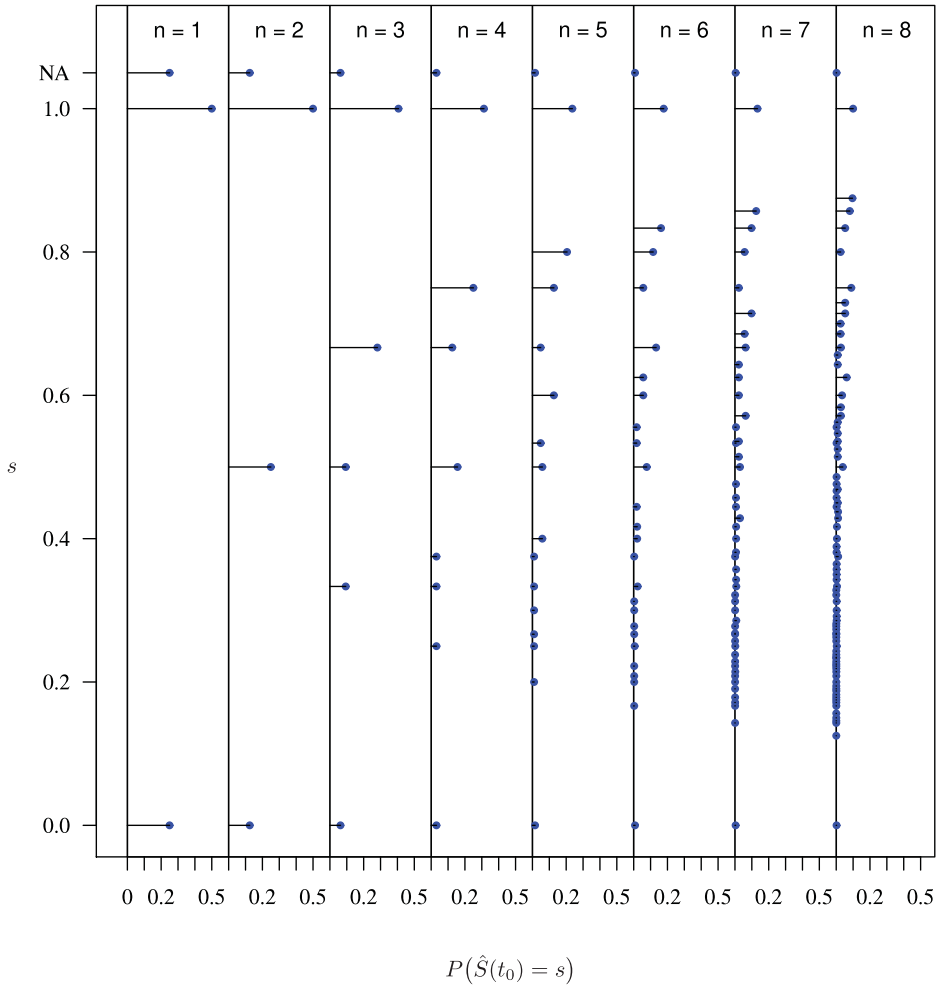

Figure 4. Probability mass functions of the KMPLE for n from 1 to 8.

$n = 3$:

$$P(\hat{S}(t_0) = s) = \begin{cases} \frac{1}{16} & s = 0 \\ \frac{3}{32} & s = \frac{1}{3} \\ \frac{3}{32} & s = \frac{1}{2} \\ \frac{9}{32} & s = \frac{2}{3} \\ \frac{13}{32} & s = 1 \\ \frac{1}{16} & s = \text{NA}. \end{cases}$$

Figure 3 displays the graph of the probability mass function for $n = 3$.

The probability mass function was generated using this algorithm for $n = 1, 2, \ldots, 8$, and the result is presented in Figure 4. Each probability mass function is rotated 90° clockwise and displayed on a common scale. These probability mass functions are supported by a Monte Carlo simulation experiment.

4. Conclusions and Further Work

The purpose of the work described here is to provide a challenging exercise for students in an advanced course in mathematical statistics, and to encourage these students to think in depth about the KMPLE. We have devised two algorithms for computing the support values of the KMPLE. As shown in Figure 1, there are significant gaps in the support of the KMPLE near 0 and 1 for small values of n. This should be recognized by the analyst when n is small and $S(t_0)$ is likely to be near 0 or 1. Figure 1 also shows the support values cluster tightly as n increases, with more support values appearing below $1/2$ than above $1/2$. A dataset that contains tied observations does not result in additional support values relative to a dataset with distinct observations. In addition, we have devised an algorithm for computing the probability mass function of the KMPLE. This algorithm has been implemented under simple conditions (exponentially distributed failure and censoring times) for $n = 1, 2, \ldots, 8$.

In terms of further work, we are interested in finding the lowest percentage increase in the support size. The limiting value of the average of the support values as $n \to \infty$ (see Figure 1) is an open question. Also, finding the probability mass function for more general conditions and larger sample sizes is another further direction of inquiry. Finally, we are exploring the application of this probability mass function in Bayesian survival analysis.

Appendix. Implementing the Induction Algorithm in R

The Maple language is ideal for generating the support values of the KMPLE via induction because (a) fractions are stored as exact values, (b) fractions are internally reduced to their lowest terms, and (c) it includes a set as a data structure, which means that duplicate support values are eliminated. The Maple code in Section 2.2 will generate the support values for any value of n subject to CPU and memory restrictions.

The induction algorithm can also be implemented in R even though exact fractions and sets are not supported. The most straightforward implementation of the induction algorithm uses the `as.fractions` function in the `MASS` package, as shown in the code below.

```
support = 1
for (n in 2:5) {
  support = unique(c(support,
                     (n - 1) * support / n))
  cat(paste("The", length(support) + 1,
            "support values for n =", n,
            "are:", paste(MASS::as.fractions(
            c(0, sort(support))),
            collapse = ", "), "\n"))
}
```

The output associated with the code for determining the support values on $[0, 1]$ associated with $n = 2, 3, 4, 5$ is shown below.

```
The 3 support values for n = 2 are: 0, 1/2, 1
The 5 support values for n = 3 are: 0, 1/3,
  1/2, 2/3, 1
The 8 support values for n = 4 are: 0, 1/4,
  1/3, 3/8, 1/2, 2/3, 3/4, 1
The 15 support values for n = 5 are: 0, 1/5,
  1/4, 4/15, 3/10, 1/3, 3/8, 2/5, 1/2, 8/15,
  3/5, 2/3, 3/4, 4/5, 1
```

Although this code should return the correct support values for any n in principle, it only works through $n = 7$; it fails for $n = 8$ and beyond. Roundoff in the values stored in the `support` vector, the tight clustering of support values illustrated in Figure 1, and failure of the `unique` and `as.fractions` functions for larger values of n mean that more sophisticated R programming is required.

The R code given below stores the numerators and denominators of the support values as integers. The outside `for` loop runs over the values of n. The inside `for` loop reduces each new support value to lowest terms. The numerators and denominators of the fractional support values are temporarily converted to the real and imaginary parts of complex numbers in order to use the `unique` function to eliminate duplicate fractions. At the bottom of the outside `for` loop, the `numer` and `denom` vectors contain the numerators and denominators of the support values for a particular value of n.

```
memory.limit(size = 160000)
n = 1
numer = 1L
denom = 1L
for (n in 2:35) {
  numer.new = (n - 1) * numer
  denom.new = n * denom
  for (i in 1:length(numer.new)) {
    numerator = numer.new[i]
    denominator = denom.new[i]
    remainder = -1
    while (remainder != 0) {
      remainder = numerator
      numerator = denominator
      if (remainder != 0)
        denominator = remainder
    }
    numer.new[i] = numer.new[i] / denominator
    denom.new[i] = denom.new[i] / denominator
 }
  numer = c(numer, numer.new)
  denom = c(denom, denom.new)
  temp  = complex(real = numer,
                  imaginary = denom)
  temp  = unique(temp)
  numer = Re(temp)
  denom = Im(temp)
  print(length(c(0, numer)))
}
```

This code has been tested for $n = 2, 3, \ldots, 35$ and the output matches the support sizes given in Table 2.

Acknowledgments

We would like to recognize the invaluable assistance of William Q. Meeker, Chris Krehbiel, and Eric Walter. We appreciate the helpful comments from an associate editor which have significantly improved the clarity of the article. The authors acknowledge William & Mary Research Computing for providing computational resources and/or technical support that have contributed to the results reported within this article. *https://www.wm.edu/it/rc*.

References

Kalbfleisch, J. D., and Prentice, R. L. (2002), *The Statistical Analysis of Failure Time Data* (2nd ed.), Hoboken, NJ: Wiley. [1]

Kaplan, E. L., and Meier, P. (1958), "Nonparametric Estimation from Incomplete Observations," *Journal of the American Statistical Association*, 53, 457–481. [1,5]

Noorden, R., Maher, B., and Nuzzo, R. (2014), "The Top 100 Papers," *Nature*, 514, 550–553. [1]