KERNEL ESTIMATION OF PROBABILITY DENSITY FUNCTIONS FOR INTERVAL-CENSORED SEXUALLY TRANSMITTED DISEASE DATA WITH DIARY INFORMATION

by

Martin Schmidt

A thesis submitted to the Graduate College of
Texas State University in partial fulfillment
of the requirements for the degree of
Master of Science
with a Major in Applied Mathematics
May 2018

Committee Members:

Qiang Zhao, Chair

Alex White

Shuying Sun

FAIR USE AND AUTHOR'S PERMISSION STATEMENT

Fair Use

Duplication Permission

ACKNOWLEDGEMENTS

I have the honor and privilage to work with my supervisor Dr. Qiang Zhao. I appreciate his patience and dedication to help me work through this process, as well as, his perserverance to help me understand survival analysis in a way that's condusive to my studies and for the production of my thesis.

Furthermore, I would like to extend my appreciation to Dr. White and Dr. Sun for making a tremendous effort to help me while time was a major nemesis. Thank you, and I could not be more grateful for your participation in this effort.

In addition, I would like to thank my parents, brother and significant other for their support and encouragement through this stressful process.

TABLE OF CONTENTS

Page

LIST OF TABLES

Table **Page**

LIST OF FIGURES

Figure **Page**

ABSTRACT

The purpose of this study is to propose a method to reliably estimate the survival function of the true infection time of a sexually transmitted disease (STD) based on interval-censored data with diary information. The survival function for interval-censored data can be estimated with Turnbull's self-consistency algorithm (Turnbull, 1976) and Braun and Stafford's (2005) proposed method. However, this data includes additional auxiliary behaviorial information, known as the diary information, in which patients record a list of sexual encounter times. In this study, we propose a method that incorporates a kernel smoothing (utilized by Braun and Stafford) and uses the addition diary information. The motivation for the study is with interval-censored data with auxiliary diary inforation provided by the Indiana University School of Medicine. Harzelak and Tu (2006) have a proposed method with the data we received but is a piecewise function like Turnbull's that incorporated a product limit estimator. Hence, we will briefly mention Turbull's algorthim and Harezlak and Tu's method in the methods section. Furthermore, the advantage of using a kernel density estimate over a piecewise estimate allows for a continuous, smooth estimate that is flexible and easy to interpret. So in this research, we will focus the estimate of the true survival function with Braun and Stafford's method and our proposed method. With data generated from a known true survival function in simulation, knowing the true survival function or desnity function we make comparisons between the two methods. We calculate the mean integrated squared error (MISE), mean square error (MSE) and bias estimates of the two methods. The results show that our method performs significantly better in most settings considered at different levels of right censoring (15%, 30%, and 40%).

I. INTRODUCTION

First, we will discuss definitions, concepts and notations in order to establish a foundation for survival estimation. Survival analysis is generally a collection of statistical methods for data analysis where the variable of interest, T, is *time until the occurance of an event* (KleinBaum, 1996). Survival analysis is used in many fields including, but not limited to: medicine, biology, economics, engineering, sociology and public health. Survival analysis is very important for attempts to predict the probability of something as important as organ failure, heart attack, relapse in cancer studies, or as technical as, reliability analysis, duration modeling or population modeling. We use these mathematical tools to measure the chances of survival as a function of the time that should be used to research solutions. Hence, survival analysis is an important tool for investigating solutions for data driven problems in contexts that's condusive to our well being. In our case, we investigate the survival time before an individual is infected with a sexually transmitted disease.

Time is measured in days and starts at the beginning of a study at time 0 until the end of a study. Each individual in the study will be encouraged to participate by attending a sequence of scheduled visits (or follow-up times), where the variable of interest is time to a STD infection. In survival analysis, the variable of interest T is the time in which the individual has "survived" up to some given time point.

1.0 Survival Function

Let T be a continuous random variable representing the infection (nonsurvival) time where $T \geq 0$. Let $F(t)$ denote the cumulative distribution function (c.d.f.)

of t with corresponding $f(t)$ the probability density function (p.d.f.). We have,

$$F(t) = P(T \leq t) = \int_0^t f(x)dx$$

When the p.d.f. is available the survival function may be computed (Kleinbaum, 1996). The probability that an individual survives beyond time t is defined by the survival function:

$$S(t) = P(T > t) = 1 - F(t) = \int_t^\infty f(x)dx$$

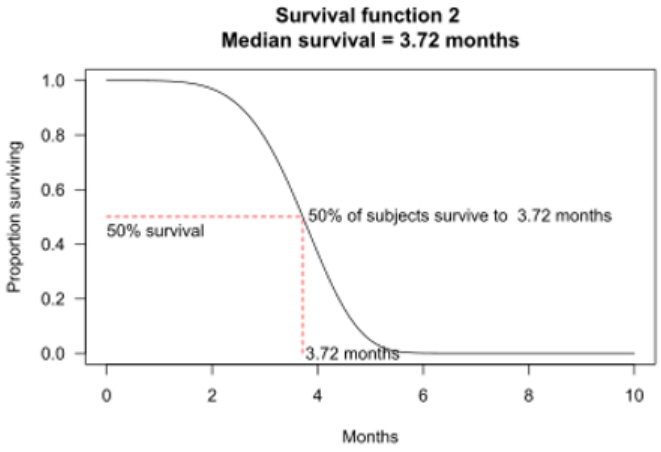

Figure 1.1: Illustration of a Survival Function

1.1 Censoring

Unfortunately, the survival time of interest may not be directly observed due to the design of a study (Rubin, 1987). Instead, we may only know that it lies in some interval resulting interval-censored data. For instance, patients will arrive to test for an infection and in one session may test negative and in a follow up session (some may potentially be skipped) will test positive some time later where we may conclude the infection time occured within that time interval. When data contains observations that are only partially known, the data is known as censored. Causes for censoring data result in an observation being

discontinued before the time of an event of interest is observed, but there exists a recent observed time past the beginning of the study that tells us the patient has survived. For example, consider a study conducted by a rehabilitation center to investigate when a patient starts suffering symptoms of withdrawals. If a subject were to arrive at a center but leave prior to a withdrawal, that observation would be considered censored. Although the withdrawal was not directly observed, we do know that the survival time is at least as long as their stay.

There are a variety of reasons censoring may occur, including but not limited to:

1. a subject is lost during the follow up study;
2. a subject may not experience the event time during the study period;
3. a subject (or patient) is dropped from the study due to death, lack of interest or fails to show up during trial times.

Depending on the behaviors of the patients, there will be a variety of different censoring types. Now, lets consider the most general types of censoring which has other types of censoring as special cases.

1.1(a) Interval-Censored

In longitudinal studies, the research practice doesn't enable the event of interest to be directly observed. Instead, there is a sequence of clinical visits to assess patients wherein the recorded times are the last visits the patient tested negative denoted L_i and the most recent visit the patient tested positive denoted R_i. This will inform us that the true infection T_i will sit inside the interval $(L_i, R_i]$ (Braun, Duchesne and Stafford, 2005). For example, a hospital may have patients come in for cancer treatment and the relapse time is the outcome of interest. Since some patients commute to the hospital for a series of routine visits where doctors won't necessarily observe the time of relapse but will know a patient has relapsed inbetween consecutive visits $\{v_1, ..., v_m\}$.

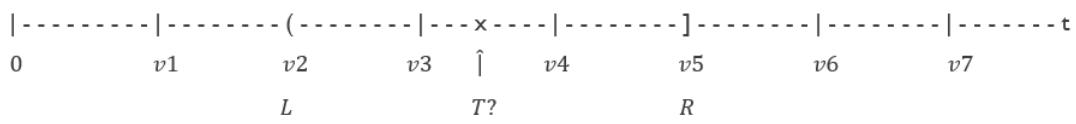

Figure 1.2: An example of an Interval-Censored Observation

Interval censoring is common in biology and medical studies, particularly when research depends on a sequence of visits where subjects are encouraged to follow-up on appointments and be assessed for the outcome of interest.

Now, other types of censoring are special cases of interval-censoring which includes: exact observation, right-censoring and left-censoring. Defined respectively, an observation is exactly observed if $L_i = R_i$, right-censored if $R_i = \infty$ and left-censored if $L_i = 0$. Note that all special cases are not exclusive to any given data set. Furthermore, there are a variety of methods that implement censored data in estimating the survival function.

Harezlak and Tu (2006) provided interval-censored data with diary information labeled *STD_Data* and *std.sextime*. Although there exists methods that consider interval-censored data, this data set is unique with the accompaniment of diary information. The objective of this research is to examine the efficiency of a variety of methods in estimating the survival function with interval-censored data in addition to the STD diary information. Furthermore, we use the software R to establish a working function to handle such data sets. Within R, we set to satisfy the following itinerary:

1. To estimate and interpret the survival function.

2. To simulate data in order to compare survival function estimates between Braun and Stafford's method, and our proposed method.

3. To assess the mean integrated square error (MISE), mean squared error (MSE) and biases for method accuracy.

II. METHODS

Methods for nonparametric estimations of the survival function will be discussed in this section. Turnbull's (1976) self-consistency algorithm is most widely used for interval-censored data as the piece-wise nonparametric estimation of the survival function. Braun and Stafford's (2005) method is a local likelihood density estimation for interval-censored data that utilizes a kernel density function for smoothing the probability density function. Lastely our method which is similar to the previously mentioned but will utilize additional auxiliary diary information.

2.0 Empirical Estimate

One non-parametric estimator of a survival function is the empirical survival function (Kleinbaum, 1996). Given complete data $T_1, ..., T_n$ for n number of patients, the empirical survival function is defined as:

$$\hat{S}(t) = \frac{\#\text{ of } T_i > t}{n}$$

2.1 Turnbull's Algorithm

This algorithm is an iterative procedure to estimate the survival function $S(t)$. Given an interval-censored data set $\{(L_i, R_i]\}_{i=1}^n$ for n patients, let $0 = \tau_1 < \tau_2 < ... < \tau_m$ be the an ordered grid of time including all the unique end points of L_i and R_i. For the ith observation, define a weight α_{ij} to indicate whether $(\tau_{j-1}, \tau_j]$ sits inside the interval $(L_i, R_i]$ where,

$$\alpha_{ij} = \begin{cases} 1 & \text{if } (\tau_{j-1}, \tau_j] \in (L_i, R_i] \\ 0 & \text{if otherwise} \end{cases}$$

With these objects the Turnbull's algorithm is as follows:

1. Make an initial guess of $\hat{p}_j^0$, where p_j is the probability mass over $(\tau_{j-1}, \tau_j]$

$$p_j = S(\tau_{j-1}) - S(\tau_j), \quad j = 1, 2, ..., m$$

2. Update the estimate of $\hat{p}_j$ by the following,

$$\hat{p}_j^l = \frac{1}{n} \sum_{i=1}^{n} \frac{\hat{p}_j^{l-1} \alpha_{ij}}{\sum_{k=1}^{m} \hat{p}_k^{l-1} \alpha_{ik}}, \quad j = 1, 2, ..., m$$

3. Repeat step 2 until convergance to a designated tolerance ϵ where $\sum_{j=1}^{m} |\hat{p_j}^l - \hat{p_j}^{l-1}| \leq \epsilon$.

The final vector $\hat{p} = (\hat{p_1}, ..., \hat{p_m})$ will give us the survival function estimate computed as:

$$\hat{S}(t) = \sum_{\tau_j > t} \hat{p}_j = 1 - \sum_{\tau_j \leq t} \hat{p}_j$$

Observe below for an example to illustrate of Turnbull's self-consistency algorithm.

Table 2.1: Simulated Interval-Censored Data

Left	**Right**	**Censored**
1	10	1
4	6	1
2	8	1
3	∞	0
6	∞	0
3	10	1
2	5	1
1	7	1

Notice $\tau = \tau_1, ..., \tau_{10}$ will be the following order of unique end points,

$$\{1, 2, 3, 4, 5, 6, 7, 8, 10\}$$

where the piecewise function will give us the following results in the intervals,

$$[1, 5), [5, 7), [7, 8), [8, 8]$$

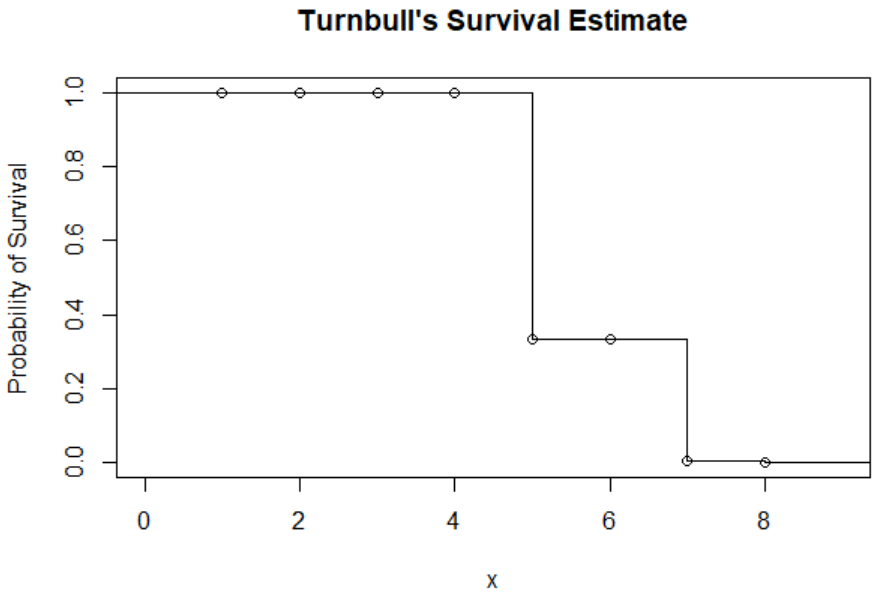

Figure 2.1: Illustration of a Turnbull's Estimate with Hypothetical Data

Table 2.2: Turnbulls Estimated Survival Values

Time	[1 - 5)	[5 - 7)	[7 - 8)	[8]
$\hat{S}(t)$	1.000	0.332	0.005	0.001

2.2 Harezlak and Tu's Method

In this section we will briefly introduce Harezlak and Tu's (2006) method to show there exists a method that has attempted to estimate the survival function with the diary information prompting this study. Harezlak and Tu propose a resampling based method that uses the auxiliary behavioral information provided by daily diaries. By imputing the unknown infection time from a list of sexual encounter times, the proposed procedure gets implemented by using a

product estimator procedure for right-censored data. The algorithm is as follows:

1. For the bth resampled data set, impute uniformly one infection (or right-censoring time) for the ith subject and denote it $X_i^{(b)}$.

   Create an indicator variable δ_i corresponding to each $X_i^{(b)}$.

   If the subject i is infected, $X_i^{(b)} = E_{ij}$ for some jth coital event and $\delta_i = 1$.

   If the subject i is right-censored $X_i^{(b)} = C_i$ and $\delta_i = 0$

   The process continues for all n subjects and will provide the complete right censored data set $\{X_i^{(b)}$, for $i = 1, ..., n\}$.

2. With data generated from Step 1, compute $\hat{S}^{(b)}(t)$ using Kaplan and Meier's (1958) product-limit estimate from the current data set. Let $t_1^{(b)}, ..., t_q^{(b)}$ be the distinct resampled infection times. $N_r^{(b)}$ is the number of infections at time $t_r^{(b)}$; $R(t_r^{(b)})$ the number of subjects at risk at time $t_r^{(b)}$, then we have $\hat{S}^{(b)}(t) = \prod_{r=1}^{q}\{1 - \frac{N_r^{(b)}}{R(t_r^{(b)})}\}$.

3. Repeat steps 1-2 up to a chosen B to obtain the estimate:

$$\hat{S}(t) = \frac{1}{B}\sum_{b=1}^{B}\hat{S}^{(b)}(t)$$

2.3 Braun and Stafford's Method

In this section we will briefly introduce the kernel density estimation method not introduced but utilized by Braun and Stafford (2005). Kernel density estimation (KDE) is a non-parametric way to estimate a propability density function of a random variable. Given independent random variables $X_1, ..., X_n$ drawn from an unknown distrubution continuous univariate density f. The kernel density estimate for f is defined as,

$$\hat{f}(x) = \frac{1}{n}\sum_{i=1}^{n}K_h(X_i - x) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{h}K(\frac{X_i - x}{h})$$

where K is the kernel, a non-negative function that integrates to 1, generally being standard normal or exponential density function. The coefficient h is a smoothing parameter called the *bandwidth*. An intepretation of $\hat{f}$ lies in the kernel weight $K_h(X_i - x)$ in terms of the proximity or rather the sum of the contributions of the observations of X_i to x.

Advantages to using a kernel density estimate for f includes providing a continuous function that is simple and flexible. As well as providing a standard technique of smoothing a nonparametric estimator of cumulative distribution functions. Hence, we will perform our comparisons solely with Braun and Stafford's method. Here is an example of the kernel density estimation of random variables drawn from a *gamma* distribution,

```
x = 5.57,  6.81,  3.78,  4.21,  3.59,  2.90, 11.23,  5.48,  5.57,  3.06
```

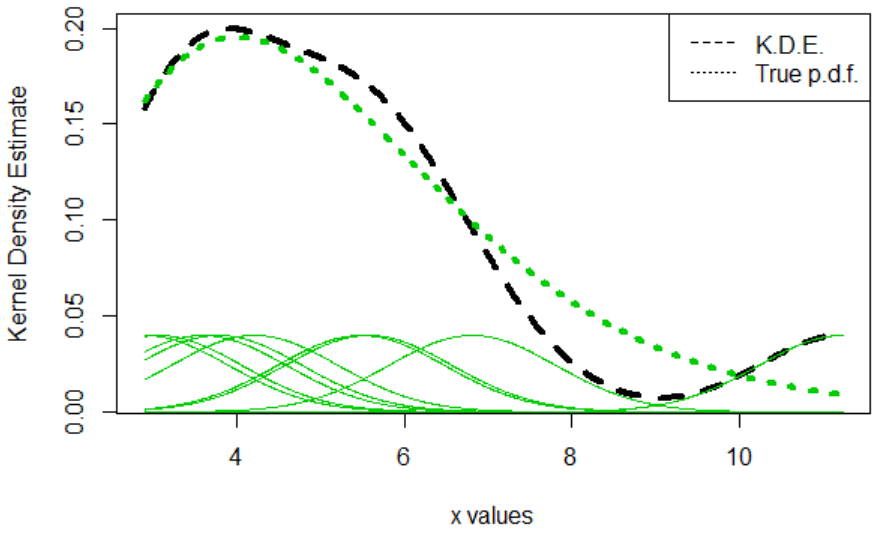

Figure 2.2: Illustration of a Kernel Density Estimate with $h = 3$ where X follows a *gamma* distribution.

When data is interval-censored (observations are recorded as $I_i = (L_i, R_i]$), a natural extenstion of the kernel density estimation is,

$$\hat{f}(x) = \frac{1}{n} \sum_{i=1}^{n} E\{K_h(X_i - x)|I_i\}$$

.

For interval-censored data, Braun and Stafford have proposed the following interative method that uses kernel estimation or smoothing,

$$\hat{f}_j(x) = \frac{1}{n}\sum_{i=1}^{n} E_{j-1}[K_h(x-X)|X \in I_i]$$

where,

$$E_k[g(X)|X \in I_i] = \begin{cases} \int_{L_i}^{R_i} g(t)\frac{\hat{f}_k(t)}{c_{k;i}}dt & L_i \neq R_i \\ g(X_i) & L_i = R_i = X_i \end{cases}$$

with,

$$c_{j-1:i} = \int_{I_i} \hat{f}_{j-1}(t)dt$$

2.4 Proposed Method

Now we will consider our proposed method. Our method will utilize the interval-censored data including the auxilary diary information that lies within the time interval of each patient.

- Given a interval-censored data set $\{(L_i, R_i]\}_{i=1}^n$ for n patients where $L_i = 0$ or otherwise and $R_i = \infty$ or otherwise (exclude patients with the interval $(0,\infty)$).
- For each patient i there will be a number of coital events b_i and $\{e_{i1}, ..., e_{ib_i}\}$ is the sequence of sexual event times.
- Consider all the unique endpoints of L_i and R_i from all patients and define the ordered grid time $0 \leq \tau_1 < ... < \tau_N$ for N unique values (ignore $R_i = \infty$).
- For each patient i we only count the coital events in $(\tau_{j-1}, \tau_j]$ that lie

within the interval $(L_i, R_i]$ and create the following weighted conditional coeifficent.

$$c_{ij} = \frac{\text{number of } e_{ik} \in (\tau_{j-1}, \tau_j]}{b_i}$$

For a given values in $(\tau_{j-1}, \tau_j]$ we consider measuring the contributions of the sexual events of the patients with the weighted conditional density $\frac{c_{ij}\hat{f}^{l-1}(s)}{\sum_{j=1}^{N}\int_{\tau_{j-1}}^{\tau_j} c_{ij}\hat{f}^{l-1}(u)du}$.

Our method is defined as follows:

$$\hat{f}^l(t) = \frac{1}{n}\sum_{i=1}^{n}\Big(\sum_{j=1}^{N}[\int_{\tau_{j-1}}^{\tau_j} K_h(t-s)\hat{f}^{l-1}(s)\frac{c_{ij}}{\sum_{j=1}^{N}\int_{\tau_{j-1}}^{\tau_j} c_{ij}\hat{f}^{l-1}(u)du}ds]\Big)$$

Make an initial guess $\hat{f}^0(t) = \frac{1}{lengthofdomain}$ and define $\hat{f}^l(t)$ for the lth iteration that takes updates based on the previous iteration and continue to update until reaching a chosen tolerance ϵ computed as $\int_{t=0}^{endOfStudy} |\hat{f}^l(t) - \hat{f}^{l-1}(t)|dt < \epsilon$.

III. SIMULATION

Next we need to compare our method to an existing method. To compute a meaningful comparison we need a data set with the true probability density function which is only known in simulated datasets. Thus, we simulate data with specified parametes that we may change to simulate other conditional behaviors that will be defined later in the section.

3.1 Data Generation Procedure

Below is the procedure in which we generate data giving patients a sequence of coital events, and true infection time along with the sequence of scheduled visits. We manage to model real life behaviors by selecting the true infection time from a sequence of coital events based on a propability of an STD infection (Katz, Fortenberry, Tu, Harezlak and Orr, 2001). Since the true infection needs to follow some type of distribution in order to measure the bias of each method, our strategy for simulating data will give us a sequence of true infections that follow a specific type of distribution.

1. Create a domain of length $endOfStudy$ (in days) and partition the domain into a sequence of equadistant visits $\{v_1, ..., v_m\}$ where $m = 20$.

2. Generate data for n patients (or observations) and for each patient i generate the total number of coital events b_i, where b_i follows a $poisson$ distribution with mean $\lambda \cdot endOfStudy$; λ is a chosen parameter that controls the average number of sexual events for the patients.

3. For each patient i we generate a sequence sexual events $\{e_{i1}, ...e_{ib_i}\}$ bounded by b_i, where $e_{ij} \sim uniform(0, ..., endOfStudy), \quad j = 1, ..., b_i$.

4. For each patient i we traverse the sequence of sexual events and select T_i where the first infection e_{ij} chosen based on Bernoulli distribution with

probability p (since getting an STD is dependent of a sexual encounter). With the distrubtion used to select b_i's and the numerical probability of infection, we can prove that T_i is exponential with mean $\lambda \cdot p$. Hence, we have a probability distribution we can use to compute the errors to evalute the proposed method and compare it to Braun and Stafford's.

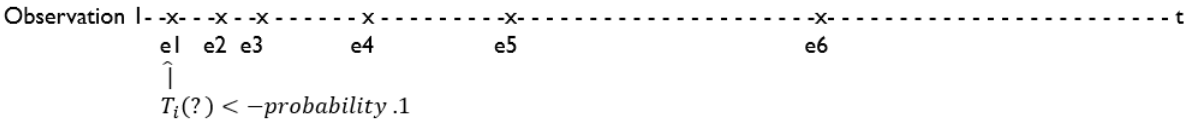

Figure 3.1: An example with generated sexual encounters with possible infection

5. Now that each patient has a true infection T_i, a sequence of sexual events $\{e_1, ..., e_{b_i}\}$. We now need to generate the right-censoring time C_i for each patient. Note that C_i needs to be independent of the true infection, so randomly select from the scheduled times where $C_i = v_k$ where $k \sim discreteuniform(1, ..., m)$.

   a) If $T_i \leq C_i$ then the infection time will be interval censored.

   b) If $T_i > C_i$ the infection time will be right censored.

   To create the censoring intervals we control a parameter B, in our case 60 days, which will dictate the average size of the intevals. First we look at the closest visit to the right of T_i we will call V_{T_i}. Next we randomly select a number from 1 to B with equal probabilities for L_i called B_{L_i} and another random number from 0 to B called B_{R_i}. Finally we will generate $(L_i, R_i]$ respectively, $L_i = round(v_{T_i} - B_{L_i})$ and $R_i = round(v_{T_i} + B_{R_i})$ (for interval censored) and $R_i = \infty$ (for right-censored) where we round the results to the closest visit.

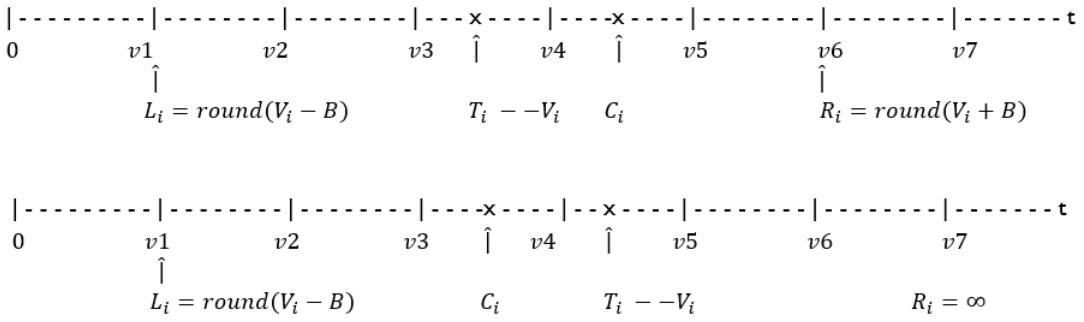

Figure 3.2: Examples of Simulated interval-censored observations (top interval-censored, bottom righ-censored)

3.2 Estimate the Probability Density Function of the Infection Time

Once the interval-censored data is obtained, Braun and Stafford's method as well as our proposed method will provide $\hat{f}$, an estimate for the probability density function which we may use to estimate the true survival function $\hat{S}(t) = \int_t^\infty \hat{f}(s)ds, t > 0$. The simulation will provide a true probability density function f, an exponential with parameter $\lambda \cdot p$. With this we will compare the estimates to the true p.d.f. function.

3.3 Evaluation of Estimation Error

In order to have a comparison between the two methods. We will repeat the simulation steps discussed in 3.1 and 3.2 M times for each setting. After we obtain M estimates of the p.d.f. of the infection time, we will estimate the mean integrated squared error by,

$$MISE = \frac{\sum_{i=1}^{M}\sum_{t=0}^{endOfStudy}((\hat{f}^{(i)}(t) - f(t))^2)\delta t}{M}$$

Next we will estimate the mean squared error and bias.

$$\hat{MSE}(t) = \frac{1}{M}\sum_{i=1}^{M}(\hat{f}^{(i)}(t) - f(t))^2$$

$$\hat{Bias}(t) = \frac{1}{M}\sum_{i=1}^{M}(\hat{f}^{(i)}(t) - f(t))$$

where $\hat{f}^{(i)}(t)$ is the estimate for the ith simulated data set and $f(t)$ is the true p.d.f.

For our investigation of MSE and $bias$ we select 30 equally spaced values of t for each data set for $M = 300$ data sets per setting.

IV. RESULTS

4.1 Applied Results

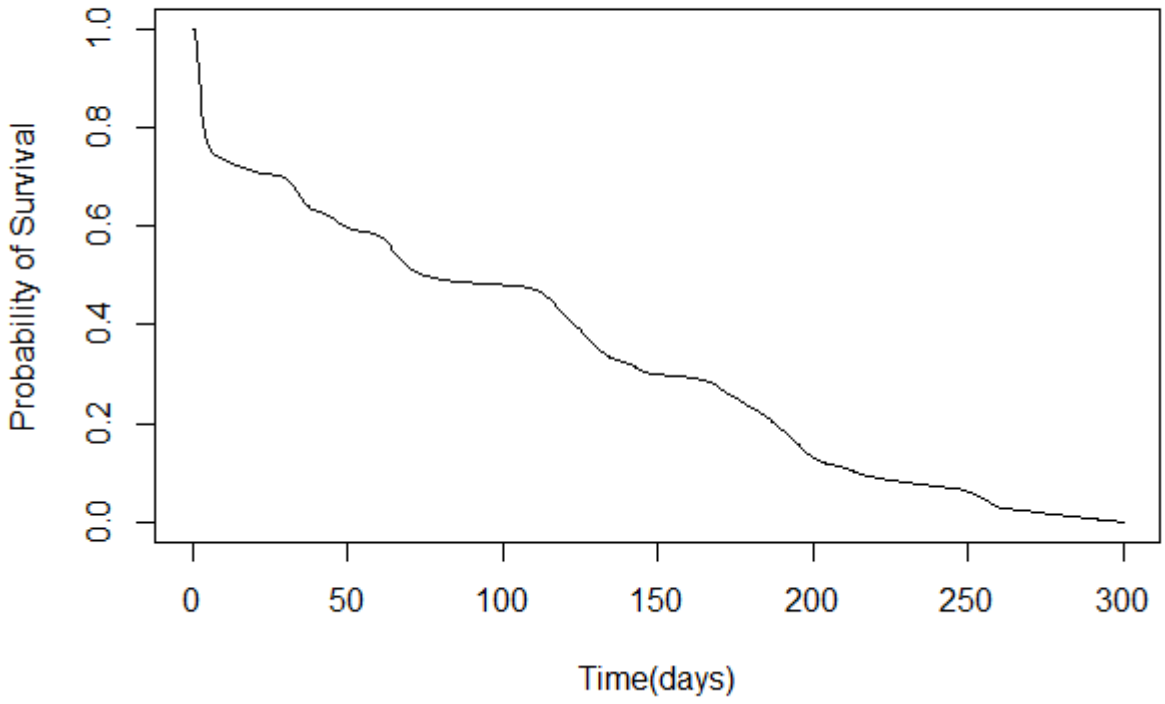

Figure 4.1: Estimated Survival Function using proposed method, $bandwidth = 2$.

Above is the results of our method applied to the STD_Data and $std.sextime$ data. We recieved this data set from Dr. Tu which was the primary motivator for this research. The survival estimate models the probability of survival from an infection where the x-axis depicts time in days and the y-axis depicts the probability of survival.

Based on the plot above, we can make the following interpretations. At 27 days there is a 75% chance for an individual to not become infected. At 101 days there is a 50% chance for an individual to not become infected. After 258 days there is a 25% chance for an individual to not become infected.

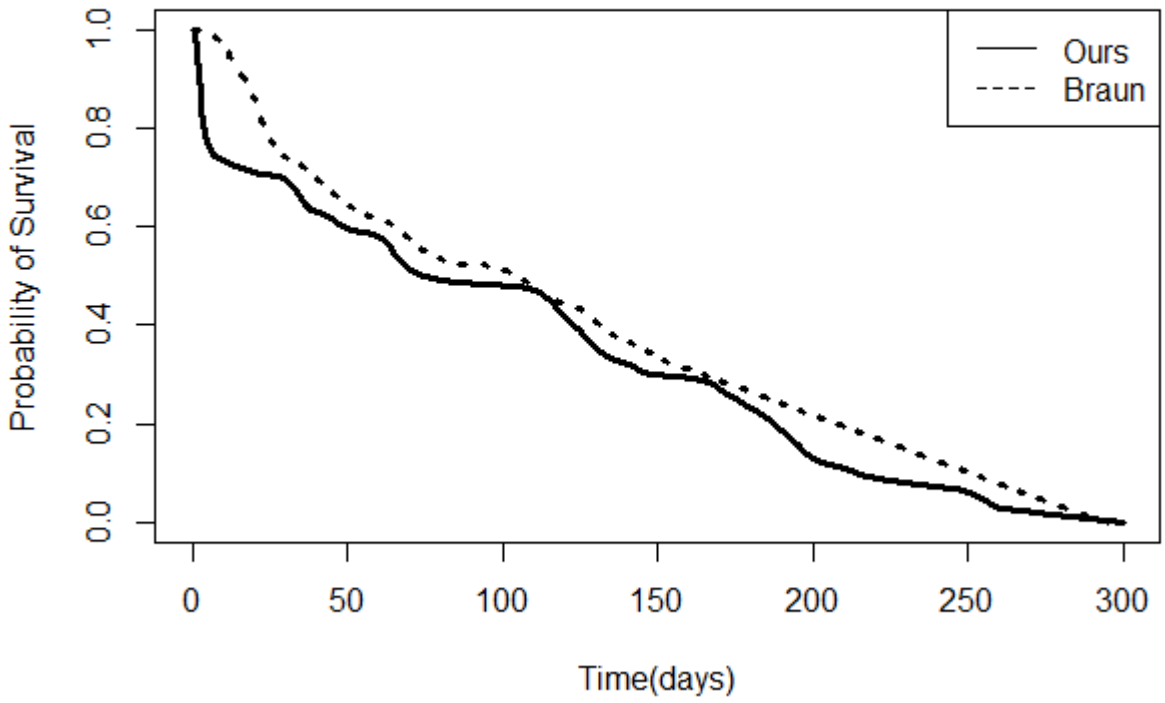

Figure 4.2: Estimated Probability Density Curves: Braun and Srafford's (dot) and proposed method (line). $bandwidth = 2$

The survival curves of the two methods are shown above. This indicates that the two methods have a roughly similar estimate, however as time progresses there is a large difference due to large amounts of right-censoring. Last, observe that our model predicts below Braun and Stafford's. This may be explained by large numbers of sexual events amongst the patients, some having as many as 96 unique values resulting in more contributions to the model.

4.2 Simulation Results

In this section we will report the results of the simulation study when comparing the two methods: Braun and Stafford and the proposed method using the R software with $M = 300$ and $B = 60$. To investigate the differences in the estimations, we consider the following settings: 15% right-censoring with shape parameters ($\lambda = .1, p > .1$ and $\lambda > .1, p = .1$), 30% right-censoring ($\lambda = p = .1$) ,

40% right-censoring ($\lambda = .1, p < .1$ and $\lambda < .1, p = .1$). The investigation of right-censoring being a function of λ and p is as follows:

1. λ is a parameter that controls the mean of sexual encounters of the patients. The larger the value of λ the more sexual encounters patients will have resulting in lower levels of right-censoring.

2. p is a parameter that controls the likelihood of any given sexual encounter to be the true infection. The larger the value of p will result in lower levels of right-censoring.

Now we make the comparisons by computing the $MISE$, MSE, and $Bias$.

4.2(a) MISE Results

Table 4.1: MISE values of two methods. M = 300, n = 100, with parameters of λ and p to correspond to levels of right-censoring.

% RC	**MISE-Braun**	**MISE-Ours**
15	0.00051612	0.00009678
15	0.00051613	0.00009673
30	0.0002444	0.00013633
40	0.0002102	0.00013865
40	0.0035181	0.0030119

The table shows that our method performs better under all settings. Observe that as the level of right-censoring increases so does the MISE for both methods. This phenomena is due to the number of patients dropping out of the study resulting in larger areas of integration for calculating the patients contribution to the overall model. Our method includes sexual encounters within the right-censored observations which results in more accurate contribution from those patients.

4.2(b) MSE Results

The mean square error (MSE) is a measure of the average square errors between the estimator and the true probability density function. The MSE differs from the MISE by measuring the squared error at one time point t and looks at the behavior of the estimators throughout the domain. This strategy allows for us to conclude where within the estimate produces the most error, and the closer the MSE value at a given t is to zero the better the estimate. Below is the sequence of tables measuring the MSE and bias for equadistant 30 t values $0, 10, 20, 30, ..., 270$ denoted $\{t_1, ..., t_{30}\}$, M = 300, n = 100, $bandwidth = 2$ (for both methods) with Parameters of λ and p to Correspond to Levels of Right-censoring. for all settings.

Table 4.2: MSE Comparison of Two Methods with 15% RC.

Time Points	**MSE-Braun**	**MSE-Ours**
t_1	0.0000125	0.00000206
t_2	0.0000164	0.00000354
t_3	0.0000148	0.00000453
t_4	0.0000135	0.00000298
t_5	0.0000149	0.00000453
t_6	0.0000147	0.00000387
t_7	0.0000105	0.00000334
t_8	0.0000182	0.00000318
t_9	0.0000184	0.00000318
t_{10}	0.0000142	0.00000193
t_{11}	0.0000151	0.00000387
t_{12}	0.0000169	0.00000268
t_{13}	0.0000161	0.00000349
t_{14}	0.0000153	0.00000208
t_{15}	0.0000252	0.00000604
t_{16}	0.0000141	0.00000236
t_{17}	0.0000208	0.00000286
t_{18}	0.0000101	0.00000238
t_{19}	0.0000169	0.00000235
t_{20}	0.0000155	0.00000271
t_{21}	0.0000158	0.00000274
t_{22}	0.0000149	0.00000214
t_{23}	0.0000211	0.00000416
t_{24}	0.0000146	0.00000262
t_{25}	0.0000117	0.00000213
t_{26}	0.0000201	0.00000364
t_{27}	0.0000191	0.00000208
t_{28}	0.0000134	0.00000346
t_{29}	0.0000161	0.00000460
t_{30}	0.0000207	0.00000328

Table 4.3: MSE Comparison of Two Methods with 30% RC. Every entry shifted by $x \cdot 10^{-5}$

Time Points	**MSE-Braun**	**MSE-Ours**
t_1	0.71665	0.46771
t_2	0.77139	0.42679
t_3	0.83990	0.53477
t_4	0.84300	0.47778
t_5	0.89047	0.46862
t_6	0.83183	0.50333
t_7	0.87151	0.55860
t_8	0.70748	0.38003
t_9	0.76809	0.40897
t_{10}	0.77523	0.33859
t_{11}	0.86330	0.48245
t_{12}	0.80084	0.45127
t_{13}	0.98349	0.63948
t_{14}	0.60839	0.37209
t_{15}	0.75784	0.44715
t_{16}	0.74965	0.34933
t_{17}	0.95912	0.47519
t_{18}	0.95865	0.57824
t_{19}	0.98602	0.44103
t_{20}	0.65883	0.39628
t_{21}	0.73865	0.49770
t_{22}	0.80541	0.47190
t_{23}	0.77089	0.39490
t_{24}	0.76345	0.38445
t_{25}	0.81866	0.49488
t_{26}	0.78248	0.34798
t_{27}	0.93389	0.54876
t_{28}	0.98702	0.48633
t_{29}	0.75231	0.39036
t_{30}	0.75270	0.42055

Table 4.4: MSE Comparison of Two Methods with 40% RC. Every entry shifted by $x \cdot 10^{-5}$

Time Points	**MSE-Braun**	**MSE-Ours**
t_1	0.58682	0.46834
t_2	0.72769	0.45232
t_3	0.67496	0.51947
t_4	0.61314	0.45100
t_5	0.79978	0.63243
t_6	0.71681	0.32143
t_7	0.76795	0.37944
t_8	0.63899	0.37794
t_9	0.67202	0.47277
t_{10}	0.67810	0.53283
t_{11}	0.74162	0.49132
t_{12}	0.58483	0.37030
t_{13}	0.64721	0.43990
t_{14}	0.73405	0.40619
t_{15}	0.65191	0.51104
t_{16}	0.67810	0.34950
t_{17}	0.73754	0.45884
t_{18}	0.81008	0.39697
t_{19}	0.77916	0.44461
t_{20}	0.62431	0.44467
t_{21}	0.72552	0.42245
t_{22}	0.81502	0.59080
t_{23}	0.72252	0.73403
t_{24}	0.61000	0.45322
t_{25}	0.64766	0.23138
t_{26}	0.71301	0.54330
t_{27}	0.89326	0.52711
t_{28}	0.78908	0.50837
t_{29}	0.64697	0.55309
t_{30}	0.59254	0.41637

Table 4.5: Bias Comparison of Two Methods with 15% RC. Every entry shifted by $x \cdot 10^{-3}$

Time Points	**Bias-Braun**	**Bias-Ours**
t_1	-0.2821	0.2643
t_2	-0.2550	-0.2384
t_3	-0.2362	-0.2250
t_4	-0.2505	-0.2331
t_5	-0.2777	-0.2564
t_6	-0.2595	-0.2412
t_7	-0.2545	-0.2314
t_8	-0.2452	-0.2319
t_9	-0.2668	-0.2412
t_{10}	-0.2526	-0.2314
t_{11}	-0.2453	-0.2220
t_{12}	-0.2392	-0.2227
t_{13}	-0.2852	-0.2585
t_{14}	-0.2613	-0.2369
t_{15}	-0.2549	-0.2376
t_{16}	-0.2705	-0.2452
t_{17}	-0.2201	-0.2094
t_{18}	-0.2549	-0.2297
t_{19}	-0.2187	-0.2046
t_{20}	-0.2315	-0.2183
t_{21}	-0.2603	-0.2469
t_{22}	-0.2392	-0.2258
t_{23}	-0.2524	-0.2442
t_{24}	-0.2559	-0.2333
t_{25}	-0.2664	-0.2443
t_{26}	-0.2738	-0.2485
t_{27}	-0.2454	-0.2268
t_{28}	-0.2394	-0.2222
t_{29}	-0.2458	-0.2254
t_{30}	-0.2606	-0.4381

Table 4.6: Bias Comparison of Two Methods with 30% RC. Every entry shifted by $x \cdot 10^{-3}$

Time Points	**Bias-Braun**	**Bias-Ours**
t_1	-0.0648	0.0583
t_2	-0.0608	-0.0549
t_3	-0.0620	-0.0567
t_4	-0.0588	-0.0516
t_5	-0.06115	-0.0545
t_6	-0.0672	-0.0602
t_7	-0.0680	-0.0607
t_8	-0.0613	-0.0544
t_9	-0.0618	-0.0566
t_{10}	-0.0703	-0.0636
t_{11}	-0.0655	-0.0610
t_{12}	-0.0594	-0.0541
t_{13}	-0.0485	-0.0434
t_{14}	-0.0605	-0.0525
t_{15}	-0.0634	-0.0552
t_{16}	-0.0634	-0.0567
t_{17}	-0.0532	-0.0481
t_{18}	-0.0610	-0.0549
t_{19}	-0.0604	-0.0527
t_{20}	-0.0645	-0.0564
t_{21}	-0.0680	-0.0593
t_{22}	-0.0577	-0.0513
t_{23}	-0.0593	-0.0522
t_{24}	-0.0574	-0.0527
t_{25}	-0.0594	-0.0537
t_{26}	-0.0671	-0.0594
t_{27}	-0.0578	-0.0531
t_{28}	-0.0680	-0.0619
t_{29}	-0.0555	-0.0488
t_{30}	-0.0561	-0.0479

Table 4.7: Bias Comparison of Two Methods with 40% RC. Every entry shifted by $x \cdot 10^{-3}$

Time Points	**Bias-Braun**	**Bias-Ours**
t_1	-0.1020	0.0908
t_2	-0.0986	-0.0850
t_3	-0.1017	-0.0883
t_4	-0.0897	-0.0806
t_5	-0.0890	-0.0775
t_6	-0.1000	-0.0848
t_7	-0.0967	-0.0842
t_8	-0.1149	-0.1037
t_9	-0.0947	-0.0836
t_{10}	-0.0910	-0.0806
t_{11}	-0.0988	-0.0863
t_{12}	-0.1073	-0.0967
t_{13}	-0.1087	-0.0974
t_{14}	-0.0991	-0.0838
t_{15}	-0.1026	-0.0896
t_{16}	-0.1061	-0.0906
t_{17}	-0.0933	-0.0828
t_{18}	-0.0922	-0.0814
t_{19}	-0.0891	-0.0769
t_{20}	-0.1110	-0.0993
t_{21}	-0.1008	-0.0892
t_{22}	-0.1054	-0.0954
t_{23}	-0.0718	-0.0607
t_{24}	-0.0791	-0.0668
t_{25}	-0.0922	-0.0812
t_{26}	-0.0909	-0.0835
t_{27}	-0.0882	-0.0766
t_{28}	-0.1091	-0.0985
t_{29}	-0.0958	-0.0855
t_{30}	-0.0936	-0.0833

V. DISCUSSION

The purpose of this thesis was to propose a procedure to estimate a survival function of STD infection time based on interval-censored data with auxiliary diary information. Without diary information there has been a variety of methods, including Turnbull's as well as Braun and Stafford's mentioned in this study, to estimate the survival function or p.d.f. based on interval -censored data. The simulation suggests that utilizing the diary information will result in our method performing significantly better against Braun and Stafford's method in settings mentioned in 4.2. In medical studies whose research practices result in data being interval censored with auixilary diary information having significant degrees of right censoring will result in our method performing better but still leaves room for improvement.

Our method was implemented in R, and the function used to create our estimate will take a variety of parameters which can adjust the characteristics of the method. The method we used to simulate data is also intuitive forward strategy to reflects reality. Given a sequence of sexual events, we know that the true infection time must come from one of the sexual encounters (given the patient was truthful when submitting the diary information). Furthermore, we have shown that changing certain parameters (such as for sexual encounters and the probability of infection) in the simulation we can still use a parametric distribution for a true survival function to perform comparisons. Lastly, the R function that runs our proposed method will be accessible and is intuitive in its implementation for future data sets that are similar to the STD data provided to us.

VI. CONCLUSION AND FURTHER RESEARCH

Based on the simulation results we can make the following remarks:

- Generally, our method performs better than Braun and Stafford's method.
- Our method has a slightly longer computation time in the event there is large numbers of sexual events.
- High levels of right censoring will lead to larger margins of error but our method still will have a better estimate.

In terms of life applications, our study illustrates that a subject's sexual behavior will greatly determine the likelihood of an STD infection and should adhere to safe practices.

Although the method performs, there are still questions that could not be answered in this investigation as well as general ideas that are worth continuing for further studies.

- Further investigations to the probability of infection with and without condom usage for a more accurate method.
- Other types of auxiliary behavioral information that could change the probability of infection such as condom use.
- The *bandwidth* used in the kernel estimation is the smoothing parameter that could be investigated to provide a more accurate estimation.

APPENDIX SECTION

APPENDIX A: R FUNCTIONS FOR METHODS

A.1 Turnbull's Algorithm

The set of functions needed to implement Turnbull's algorithm:

Function cria.tau, takes a data set of interval-censored data $(L_i, R_i]$ and returns a vector of unique end points of L_i and R_i.

Function S.ini, will take the tau vector and return a vector of probabilities of infection at each τ_i.

Function cria.A, will take the data and tau vector and return the α matrix where the entries of α_{ij} will be 1 if $(\tau_{j-1}, \tau_j] \in (L_i, R_i]$ and 0 otherwise.

Function Turnbull, will take the p vector, alpha matrix, data set, tolerance, and maximum number of iterations and return a matrix with a column of xvalues and a column of corresponding yvalues.

```
data$right[is.na(data$right)] <- Inf
cria.tau <- function(data){
  l <- data$left
  r <- data$right
tau <- sort(unique(c(l,r[is.finite(r)])))
  return(tau)
}

S.ini <- function(tau){
  m<-length(tau)
  ekm<-survfit(Surv(tau[1:m-1],rep(1,m-1))~1,data=data)
  So<-c(1,ekm$surv)
  p <- -diff(So)
```

```
  return(p)
}

cria.A <- function(data,tau){
  tau12 <- cbind(tau[-length(tau)],tau[-1])
  interv <- function(x,inf,sup)
    ifelse(x[1]>=inf
           & x[2]<=sup,1,0)
  A <- apply(tau12,1,interv,inf=data$left,sup=data$right)
  id.lin.zero <- which(apply(A==0,
                  1, all))
  if(length(id.lin.zero)>0)
    A <- A[-id.lin.zero,
           ]
  return(A)
}

Turnbull <- function(p,
            A, data,
            eps=1e-3,
            iter.max=200,
            verbose=FALSE){
  n<-nrow(A)
  m<-ncol(A)
  Q<-matrix(1,m)
  iter <- 0
  repeat
  {
    iter<- iter + 1
```

```
  diff<- (Q-p)
  maxdiff<-max(abs(as.vector(diff)))
  if (verbose)
    print(maxdiff)
  if (maxdiff<eps
      | iter>=iter.max)
    break
  Q<-p
  C<-A%*%p
  p<-p*((t(A)%*%(1/C))/n)
}
cat("Iterations
    = ", iter,"\n")
cat("Max
    difference
    = ", maxdiff,"\n")
cat("Convergence
    criteria:
    Max
    difference
    < 1e-3","\n")
dimnames(p)<-list(NULL,c("P
                         Estimate"))
surv<-round(c(1,1-cumsum(p)),digits=5)
right <- data$right
if(any(!(is.finite(right)))){
  t <- max(right[is.finite(right)])
  return(list(time=tau[tau<t],surv=surv[tau<t]))
}
```

```
  else
    return(list(time=tau,surv=surv))
}
```

A.2 Braun and Stafford's method

Braun and Stafford's method is applied using the "ICE package" in R. Using the "ickde" (interval-censored kernel density estimator) function takes in the following parameters:

I <- A matrix with two columns of left and right end points.

h <- bandwidth

f <- inital estimate of f

m <- number of gridpoints

n.iterations <- maximum number of iterations

x1 <- left most grid point

xm <- right most grid point

right.limit <- artificial right censored value

kernel <- kernel function used for estimation

old <- logical value to indicate conditional expectation value to use previous iteration density estimate

Function returns a matrix with a column of xvalues and a column of corresponding yvalues.

```
estimate <- ickde(I, h, f, m, n.iterations = 10, x1, xm,
right.limit = 1000,kernel="standardnorm", old=TRUE)
```

A.3 Proposed Method

Our proposed method implemented in R. Listed is the necessary parameters needed to run the function with corresponding default settings.

data <- takes in the data set (labeled or otherwise) where the entries are of the form L_i (left column) and R_i (right column) where right censored values

are $R_i = \infty$.

diaryInformation <- contains the corresponding recorded sex times where the rows correspond to the patient in the rows in data. Empty values should be recorded as NA.

bw <- bandwidth, which takes the necessary bandwidth for kernel smoothing [default $bw < -1$].

endOfStudy <- takes in the last day of the trial studies, also used as an artifical limit to right censored data [default $endOfStudy < -max(tau)$].

domx <- domain of x, a desired length for the discritizing of the domain [default $domx < -500$].

iter <- iterations, a desired number of iterations [default, iterations reaching convergance of ϵ].

kernelFunc <- kernel function, a desired probability distribution used for the kernel smoothing, use standard r syntax for function [default $kernelFunc < -dnorm$ "standard normal"].

Function returns a matrix with a column of xvalues and a column of corresponding yvalues.

Any additional information about the individual functions and variables are supplied in the comments of the code.

```
##########################################
# KernelEstimate which will take in the data and auxilary diary
#   and condom information to produce a survival function estimate
#
# data: will be an excel file that contains left end points and
#   respective right end points for each patient
#   NOTE: right censored observations should have right end points
#         with the entry "Inf"
#
```

```
# diaryInformation: will contain an excel file that contains the
#   the recorded sexual (or events of interest) for each patient
#   NOTE: any missing entries should contain the value "NA"
#
# bw (bandwidth) will be the prompted bandwidth for the kernel
#   estimation
#
# endOfStudy will contain the value for the last day or length
#   of the study
#
# domx will be the length of the domain
#
# iter (iterations)
##########################################
kernelEstimate
<- function(data,diaryInformation,bw,endOfStudy,domx,iter,kernelFunc){
data <- as.data.frame(data)
colnames(data) <- c("left","right")
diary <- diaryInformation

##########################################
# x is the domain vector
##########################################
x <- seq(0,endOfStudy, length = domx)

##########################################
# tau will be the ordered time grid from all the unique end
#   points of L_i and R_i in the data
##########################################
```

```
cri.tau <- function(data){
  l <- data$left
  r <- data$right
  tau1 <- sort(unique(c(l,r[is.finite(r)],endOfStudy)))
  return(tau1)
}
tau <- cri.tau(data)

##########################################
# Alpha matrix, rows are the observation (intervals) and the columns
#   are the tau values.
# Each entry will hold a 1 or a 0 depending if tau sits inside the
#   interval of L_i,R_i
##########################################
cria.alpha <- function(data,tau){
  tau12 <- cbind(tau[-length(tau)],tau[-1])
  interv <- function(t,inf,sup)
    ifelse(t[1]>=inf & t[2]<=sup,1,0)
  A1 <- apply(tau12,1,interv,inf=data$left,sup=data$right)
  return(A1)
}
A <- cria.alpha(data,tau)

##########################################
# tauindex will give the index of L_i in the tau vector
##########################################
tauadjindex <- function(tau,data)
{
  p <- rep(0, times = nrow(data))
```

```
  for(i in 1:nrow(data))
  {
    index <- 1
    while(tau[index] <= data[i,1])
    {
      p[i] <- index
      index <- index + 1
      if(index == length(tau)+1){break}
    }
  }
  return(p)
}
tauindex <- tauadjindex(tau,data)

##########################################
# tauDomainIndex will give the index location of each tau in the
#   domain vector.
##########################################
tauAdjDomainIndex <- function(x,tau)
{
  temp <- rep(0, times = length(tau))
  for(i in 1:length(tau))
  {
    temp[i] <- ceiling(tau[i]/x[length(x)]*length(x))
  }
  return(temp)
}
tauDomainIndex <- tauAdjDomainIndex(x,tau)
```

```
##########################################
# intindex (interval index) will give the index location for each
#   interval for L_i and R_i.
##########################################
intervaladjindex <- function(data,x)
{
  p <- matrix(1, nrow = nrow(data), ncol = 2)
  for(k in 1:nrow(data))
  {
    i <- 1
    while((x[i] <= data[k,2] || data[k,2] == "Inf") && i <= length(x)){
      if(x[i] <= data[k,1]){
        p[k,1] <- i
      }
      if(data[k,2] == "Inf"){
        p[k,2] <- length(x)
      }
      else{
        p[k,2] <- i
      }
      i <- i+1
    }
  }
  return(p)
}
intindex <- intervaladjindex(data,x)

#####################################
# diaryAlpha will be a matrix where the rows are the individual
```

```
#   observation and the columns are the sequence of taus
# Each entry will contain the C_ij coefficient, the number of diary
#   events that occur in the interval ( tau_(j-1),tau_j ]
#######################################
diaryalpha <- function(tau,diary,A,tauindex,data)
{
  p <- matrix(0, nrow = nrow(A), ncol = ncol(A) + 1)
  for(i in 1:nrow(A))
  {
    sum <- 0
    j <- 1
    taui <- tauindex[i] + 1


    while(!(is.na(diary[i,j])))
    {
      if(diary[i,j] >= data[i,1] && diary[i,j] <= data[i,2]){


        if(diary[i,j] <= tau[taui] && diary[i,j] != data[i,1])
        {
          sum <- sum + 1
          p[i,taui] <- p[i,taui] + 1
        }
        if(j+1 == ncol(diary)+1){break;}
        if(!(is.na(diary[i,j+1])))
        {
          if(diary[i,j] > tau[taui])
          {
```

```
            j <- j - 1
          }
          if(diary[i,j+1] > tau[taui])
          {
            taui <- taui+1
          }
        }
      }
      if(j+1 == ncol(diary)+1){break;}

      j <- j + 1
    }
    p[i,] <- p[i,]/sum

  }
  return(p)
}
diaryA <- diaryalpha(tau,diaryInfo,A,tauindex,data)

########################################
# initf (initial function guess) will contain the initial estimate
#   of the probability density function f for the iterative procedure
########################################
initialguess <- (1/length(x))
guess <- rep(initialguess, times = length(x))

########################################
# function that produces a vector sum of the conditional contribution of
#   the diaryinformation for each (L_i,R_i]
```

```
########################################
sumConditional <- function(diaryA,fit,x){
  sumVector <- rep(0, times = nrow(diaryA))
  for(i in 1:nrow(diaryA))
  {
    sum <- 0
    taui <- tauindex[i] + 1
    for(j in intindex[i,1]:intindex[i,2])
    {
      if((rowSums(diaryA)[i] != "NaN" || !(is.na(rowSums(diaryA)[i])))
&& taui <= ncol(diaryA)){
        if(diaryA[i,taui] != 0){
          sum <- sum + fit[j]*diaryA[i,taui]*(tau[taui]-tau[taui-1])
/(tauDomainIndex[taui]-tauDomainIndex[taui-1])
        }}
      if(j != length(x)){
        if(x[j+1] > tau[taui] && taui < ncol(diaryA)){taui <-taui+1}
      }
    }
    sumVector[i] <- sum
  }
  return(sumVector)
}

########################################
# function that shifts the kernel values to fit the corresponding
#   previous iteration function for the integrated value in the
#   estimate
########################################
```

```
kernelfunc <- function(k,kern,itf,domx)
{
  temp <- kern
  for(i in 1:length(itf))
  {
    temp[domx+1 - k + i] <- kern[domx+1 - k + i]*itf[i]
  }
  return(temp)
}

##########################################
# the estimate function that takes in each necessary parameter to
#   produce the most updated iteration of the estimate f
##########################################
hat <- function(x,fit,diaryA,data,intindex,tauindex,
    tau,bw,tauDomainIndex,kernelFunc){
  Kern <- rep(0, times = 2*length(x) + 1)
  x1 <- seq(0,2*endOfStudy, length = c(2*length(x) + 1))
  for(i in 1:(2*length(x)+1))
  {
    Kern[i] <- kernelFunc((x1[length(x) + 1] - x1[i])/bw)
  }
  sumVect <- sumConditional(diaryA,fit,x)
  estimate <- rep(0, times = length(x))
  for(k in 1:length(x)){
    kernel <- kernelfunc(k,Kern,fit,length(x))
    for(t in 1:(length(tau)-1)){
      integralSum <- 0
      left <- abs(tauDomainIndex[t] - k)
```

```
      right <- abs(tauDomainIndex[t+1] - k)
      if(x[k] > tau[t] && x[k] <= tau[t+1]){
        integralSum <- sum(kernel[(length(x) + 1-left):
(length(x) + 1+right)])}
      else if(x[k] >= tau[t+1]){
        integralSum <- sum(kernel[(length(x) + 1-left):
(length(x) + 1-right)])}
      else if(x[k] < tau[t]){
        integralSum <- sum(kernel[(length(x) + 1+left):
(length(x) + 1+right)])}
      pvalue <- 0
      for(i in 1:nrow(diaryA))
      {
        if(k > intindex[i,1] && k <= intindex[i,2]){
          if(rowSums(diaryA)[i] != "NaN")
          {
            if(diaryA[i,t+1] != 0){
              pvalue <- pvalue + diaryA[i,t+1]/sumVect[i]
            }
          }
          else
          {
            pvalue <- pvalue + 1
          }
        }
      }
      estimate[k] <- estimate[k] + integralSum*pvalue*(tau[t+1] - tau[t])
/(tauDomainIndex[t+1]- tauDomainIndex[t])
    }
```

```
  }
  estimate <- estimate/(bw*nrow(data))
  return(estimate)
newestimate
<- hat(x,guess,diaryA,data,intindex,tauindex,tau,bw,tauDomainIndex,kernelFunc)

################################################
# The iterative process for estimating f
################################################
for(i in 1:iter)
{
  newestimate
<- hat(x,newestimate,diaryA,data,intindex,tauindex,tau,bw,
tauDomainIndex,kernelFunc)
}
return(list(x=x,y=newestimate))
}
```

A.4 Simulation Algorithm

Data simulation algorithm produced in R with the following parameters:

numObservations <- takes in the total number of observations (patients).

endOfStudy <- takes in the last xvalue of the domain.

probability <- takes in the probability of infection at any given sexual event. Parameter used to generate the true survival function.

lambda <- parameter used to control the average number of sexual events b_i per patient. Parameter used to generate the true survival function.

trialLength <- used to give distances per visit of $\{v_1, ..., v_N\}$ for N visits.

itervalSize <- used to control the average size of the intervals B for the interval-censored data.

sexEvents <- will contain a vector where each entry corresponds to the number of sexual encounters per patient.

diaryInfo <- contains a matrix where each row corresponds to a patient and each entry contains a time value of a sexual encounter.

condomInfo <- contains a matrix where each entry corresponds to a patient and whether or not a condom was used during the sexual encounter.

Further details of the simulation algorithm are described in the simulation chapter.

```
#DATA SAMPLING ALGORITHM


numObservations <- 100
endOfStudy <- 300
lambda <- .05
probability <- .1
maxSexEvents <- 300
trialLength <- 15
```

```
intervalSize <- 60
sexEvents <- rpois(numObservations,lambda*endOfStudy)
diaryInfo <- matrix(, nrow = numObservations, ncol = max(sexEvents))
condomInfo <- matrix(, nrow = numObservations, ncol = max(sexEvents))

for(i in 1:numObservations)
{
  diaryInfoTemp <- sort(sample(1:endOfStudy,sexEvents[i]))
  for(j in 1:max(sexEvents))
  {
    diaryInfo[i,j] <- diaryInfoTemp[j]
    condomInfo[i,j] <- round(runif(1,0,1))
  }
}
#infectionTimes <- function(diaryInfo, condomInfo)
#{
trueInfection <- matrix(0,nrow = nrow(diaryInfo), ncol = 1)
for(i in 1:nrow(diaryInfo))
{
  j <- 1
  while(!(is.na(diaryInfo[i,j])))
  {
    if(rbinom(1,1,probability) == 1)
    {trueInfection[i] <- diaryInfo[i,j]
    break}
    else{j <- j+1}
  }
  if(trueInfection[i] == 0)
  {trueInfection[i] <- sample(diaryInfo[i,j-1],1)
```

```
  #Useless Patient Count Occurs Here
  uselessPatients <- uselessPatients + 1
  }
  if(is.na(diaryInfo[i,1]))
  {
    trueInfection[i] <- round(runif(1,1,endOfStudy))
  }
}
#  return(trueInfection)
#}
#trueInfection <- infectionTimes(diaryInfo,condomInfo)
#intervalCensoring <- function(trueInfection,endOfStudy,trialLength,intervalSize
#{
data <- matrix(0, nrow = nrow(trueInfection), ncol = 2)
partitionSize <- round(endOfStudy/trialLength)
c <- matrix(0, nrow = partitionSize)
for(i in 1:partitionSize)
{c[i+1] <- c[i] + trialLength}
for(i in 1:nrow(trueInfection))
{
  ci <- sample(c,1)
  B <- intervalSize/trialLength
  if(ci >= trueInfection[i]){
    intervalCenter<-ceiling(trueInfection[i]/15) + 1
    li <- intervalCenter - sample(1:min(B,intervalCenter-1),1)
    ri <- intervalCenter + sample(0:min(B,partitionSize-intervalCenter),1)
    #while( li <= 0 || ri > partitionSize)
    #{
    #  li <- intervalCenter - sample(1:B,1)
```

```
  #  ri <- intervalCenter + sample(0:B,1)
  #}
  data[i,1] <- c[li]
  data[i,2] <- c[ri]
}
else{
  intervalCenter<-ceiling(trueInfection[i]/15) + 1
  li <- intervalCenter - sample(1:min(B,intervalCenter-1),1)
  data[i,1] <- c[li]
  data[i,2] <- Inf
}
while(data[i,1] ==0 && data[i,2] == Inf)
{
  diaryInfoTemp <- sort(sample(1:endOfStudy,sexEvents[i]))
  for(j in 1:max(sexEvents))
  {
    diaryInfo[i,j] <- diaryInfoTemp[j]
    condomInfo[i,j] <- round(runif(1,0,1))
  }
  j <- 1
  while(!(is.na(diaryInfo[i,j])))
  {
    if(rbinom(1,1,probability) == 1)
    {trueInfection[i] <- diaryInfo[i,j]
    break}
    else{j <- j+1}
  }
  if(trueInfection[i] == 0)
  {trueInfection[i] <- sample(diaryInfo[i,j-1],1)
```

```
    }
    if(is.na(diaryInfo[i,1]))
    {
      trueInfection[i] <- round(runif(1,1,endOfStudy))
    }
    ci <- sample(c,1)
    B <- intervalSize/trialLength
    if(ci >= trueInfection[i]){
      intervalCenter<-ceiling(trueInfection[i]/15) + 1
      li <- intervalCenter - sample(1:min(B,intervalCenter-1),1)
      ri <- intervalCenter + sample(0:min(B,partitionSize-intervalCenter),1)
      #while( li <= 0 || ri > partitionSize)
      #{
      #  li <- intervalCenter - sample(1:B,1)
      #  ri <- intervalCenter + sample(0:B,1)
      #}
      data[i,1] <- c[li]
      data[i,2] <- c[ri]
    }
    else{
      intervalCenter<-ceiling(trueInfection[i]/15) + 1
      li <- intervalCenter - sample(1:min(B,intervalCenter-1),1)
      data[i,1] <- c[li]
      data[i,2] <- Inf
    }
  }
}
#  return(data)
#}
```

```
#data <- intervalCensoring(trueInfection,endOfStudy,trialLength,intervalSize)
```

REFERENCES

[1] Braun J, Duchesne T, and Stafford J (March, 2005), "Local Likelihood Density Estimation for Interval Censored Data". *The Canadian Journal of Statistics* 33, no. 1 , pp. 39-60.

[2] Harezlak J and Tu W (2006), "Estimation of survival functions in interval and right censored data using STD behavioural diaries". *Statistics in Medicine* 25, pp. 4053-4064.

[3] Kaplan EL and Meier P (1958) Nonparametric estimation from incomplete observations. *Journal of the American Statistical Association.*

[4] Katz BP, Fortenberry JD, Tu W, Harezlak J and Orr DP (2001), " Sexual behavior among adolescent women at high risk for sexually transmitted infections". *Sexually Transmitted Diseases* 28 (5), pp. 247-251.

[5] Kleinbaum D.G., (1996), *Survival Analysis: A self-Learning Text.* New York: Springer+Business Media, Inc.

[6] Rubin DB (1987) Multiple imputation for nonresponse in surveys. *Wiley: New York.*

[7] Turnbull BW (1976), "The empirical distribution function with arbitrarily grouped censored and truncated data". *Journal of the Royal Statistical Society* B38, pp. 290-295.

[8] Zhao Q and Sun J (2004), "Generalized log-rank test for mixed interval-censored failure time data". *Statistics in Medicine* 23, pp. 1621-1629.