Interval Censored Data Analysis

Michael P. Fay

National Institute of Allergy and Infectious Diseases

Tutorial:Interval Censored Data Analysis
useR! 2010
July 20, 2010

Outline

- Part 1:
  - Description of Interval Censoring
  - Nonparametric Maximum Likelihood Estimation of Distribution
- Part 2:
  - Testing
  - Parametric Regression
  - Semi-parametric Regression

Notation

- X_i is the time from start of the study until the event of interest for the ith subject.

Notation

- X_i is the time from start of the study until the event of interest for the ith subject.
- $\mathbf{G}_i = [G_{i0}, G_{i1}, \ldots, G_{im_i}]$ is the vector of assessment times.

Notation

- X_i is the time from start of the study until the event of interest for the ith subject.
- $\mathbf{G}_i = [G_{i0}, G_{i1}, \ldots, G_{im_i}]$ is the vector of assessment times.
  - $G_{i0} = 0$, initial assessment at start of study

Notation

- X_i is the time from start of the study until the event of interest for the ith subject.
- $\mathbf{G}_i = [G_{i0}, G_{i1}, \ldots, G_{im_i}]$ is the vector of assessment times.
  - $G_{i0} = 0$, initial assessment at start of study
  - G_{i1}, first assessment after study begins

Notation

- X_i is the time from start of the study until the event of interest for the ith subject.
- $\mathbf{G}_i = [G_{i0}, G_{i1}, \ldots, G_{im_i}]$ is the vector of assessment times.
  - $G_{i0} = 0$, initial assessment at start of study
  - G_{i1}, first assessment after study begins
  - $\vdots$
  - G_{im_i}, last assessment done on ith subject.

Notation

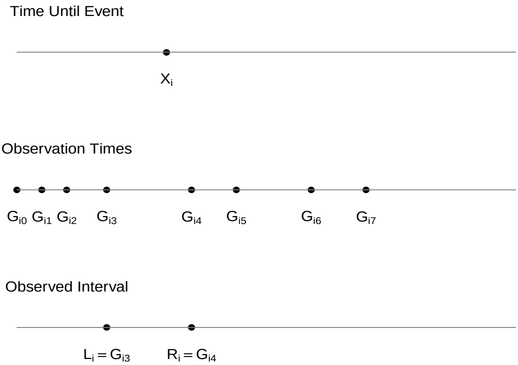
- X_i is the time from start of the study until the event of interest for the ith subject.
- $\mathbf{G}_i = [G_{i0}, G_{i1}, \ldots, G_{im_i}]$ is the vector of assessment times.
  - $G_{i0} = 0$, initial assessment at start of study
  - G_{i1}, first assessment after study begins
  - $\vdots$
  - G_{im_i}, last assessment done on ith subject.
- $(L_i, R_i]$ interval in which event is known to occur.

Notation

- X_i is the time from start of the study until the event of interest for the ith subject.
- $\mathbf{G}_i = [G_{i0}, G_{i1}, \ldots, G_{im_i}]$ is the vector of assessment times.
  - $G_{i0} = 0$, initial assessment at start of study
  - G_{i1}, first assessment after study begins
  - $\vdots$
  - G_{im_i}, last assessment done on ith subject.
- $(L_i, R_i]$ interval in which event is known to occur.
  - L_i left endpoint (may be 0)
  - R_i right endpoint (may be ∞)

Notation

- X_i is the time from start of the study until the event of interest for the ith subject.
- $\mathbf{G}_i = [G_{i0}, G_{i1}, \ldots, G_{im_i}]$ is the vector of assessment times.
  - $G_{i0} = 0$, initial assessment at start of study
  - G_{i1}, first assessment after study begins
  - $\vdots$
  - G_{im_i}, last assessment done on ith subject.
- $(L_i, R_i]$ interval in which event is known to occur.
  - L_i left endpoint (may be 0)
  - R_i right endpoint (may be ∞)
  - Usually only observe $(L_i, R_i]$ not $\mathbf{G}_i$.



Time Until Event

X_i

Observation Times

G_{i0} G_{i1} G_{i2} G_{i3} G_{i4} G_{i5} G_{i6} G_{i7}

Observed Interval

$L_i = G_{i3}$ $R_i = G_{i4}$

Some Assumptions

- Non-informative Censoring

Some Assumptions

- Non-informative Censoring
  - Example: $\mathbf{G}_i$ (assessment times) independent of X_i (event times)
    - Regularly scheduled assessment, changes in schedule not related to event time. For example, must get blood draw to know if disease has occurred, but follow regular blood draw schedule regardless of true disease status.

Some Assumptions

- Non-informative Censoring
  - Example: $\mathbf{G}_i$ (assessment times) independent of X_i (event times)
    - Regularly scheduled assessment, changes in schedule not related to event time. For example, must get blood draw to know if disease has occurred, but follow regular blood draw schedule regardless of true disease status.
    - with Progression-Free Survival (time to first of disease progression or death) this assumption is not likely to be met. Deaths will change assessment schedule, because assess death in near-continuous time not at next scheduled appointment (more on that later).

Some Assumptions

- Non-informative Censoring
  - Example: $\mathbf{G}_i$ (assessment times) independent of X_i (event times)
    - Regularly scheduled assessment, changes in schedule not related to event time. For example, must get blood draw to know if disease has occurred, but follow regular blood draw schedule regardless of true disease status.
    - with Progression-Free Survival (time to first of disease progression or death) this assumption is not likely to be met. Deaths will change assessment schedule, because assess death in near-continuous time not at next scheduled appointment (more on that later).
    - We will often assume independent censoring to start. Later talk about how bad the violation of the assumption can be.

Some Assumptions

- Non-informative Censoring
  - Example: $\mathbf{G}_i$ (assessment times) independent of X_i (event times)
    - Regularly scheduled assessment, changes in schedule not related to event time. For example, must get blood draw to know if disease has occurred, but follow regular blood draw schedule regardless of true disease status.
    - with Progression-Free Survival (time to first of disease progression or death) this assumption is not likely to be met. Deaths will change assessment schedule, because assess death in near-continuous time not at next scheduled appointment (more on that later).
    - We will often assume independent censoring to start. Later talk about how bad the violation of the assumption can be.
- Informative Censoring
  - Assessment schedule may be related to event.

Types of Independent Interval Censoring:

Case 1: Only 1 observation time. Also called current status data.

Case 2: Only 2 observation times.

Case K: Arbitrary number of observation times.

Types of Independent Interval Censoring:

Case 1: Only 1 observation time. Also called current status data.

Case 2: Only 2 observation times.

- Do not confuse with many observation times, but only keeping the interval, $(L_i, R_i]$.

Case K: Arbitrary number of observation times.

Types of Independent Interval Censoring:

Case 1: Only 1 observation time. Also called current status data.

Case 2: Only 2 observation times.

- Do not confuse with many observation times, but only keeping the interval, $(L_i, R_i]$.
- Rare in Practice.

Case K: Arbitrary number of observation times.

Types of Independent Interval Censoring:

Case 1: Only 1 observation time. Also called current status data.

Case 2: Only 2 observation times.

- Do not confuse with many observation times, but only keeping the interval, $(L_i, R_i]$.
- Rare in Practice.
- Used for theoretical work with continuous time inspection processes

Case K: Arbitrary number of observation times.

Types of Independent Interval Censoring:

Case 1: Only 1 observation time. Also called current status data.

Case 2: Only 2 observation times.

- Do not confuse with many observation times, but only keeping the interval, $(L_i, R_i]$.
- Rare in Practice.
- Used for theoretical work with continuous time inspection processes

Case K: Arbitrary number of observation times.

- Usually will assume Case K interval censoring.

Nonparametric Maximum Likelihood of Survival

Right-Censored Data

- NPMLE is Kaplan-Meier estimate

Nonparametric Maximum Likelihood of Survival

Right-Censored Data

- NPMLE is Kaplan-Meier estimate
- Usually assume event time is measured continuously. In practice it is measured discretely (e.g., nearest day, or minute).

Nonparametric Maximum Likelihood of Survival

Right-Censored Data

- NPMLE is Kaplan-Meier estimate
- Usually assume event time is measured continuously. In practice it is measured discretely (e.g., nearest day, or minute).
- To start we will treat event times as continuous.

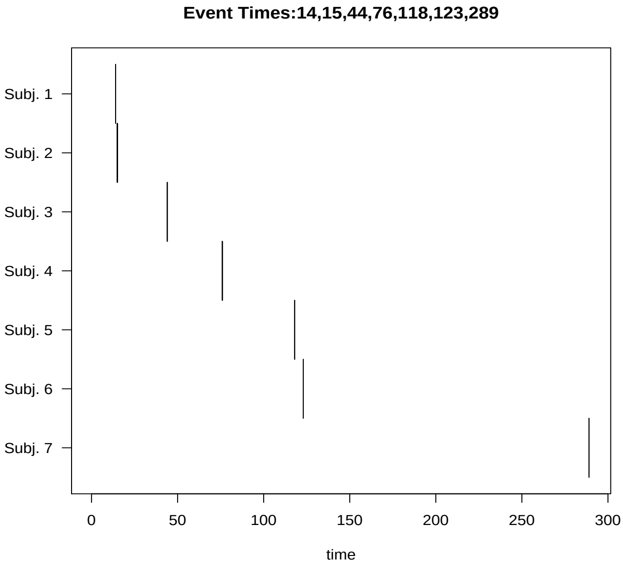
Example: No censoring

Consider 7 observed event times:

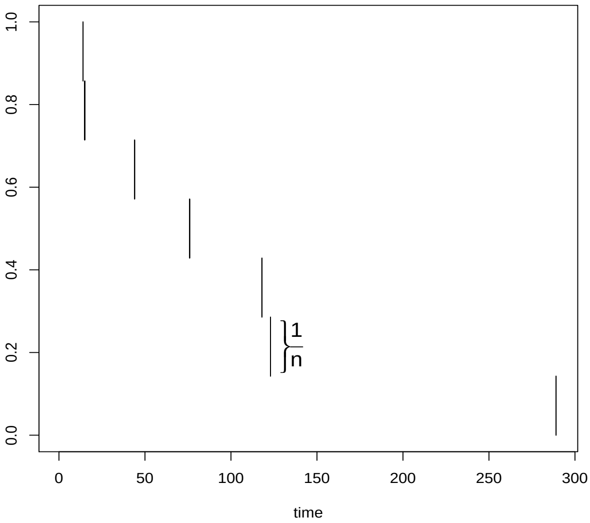
For presentation order data by event times

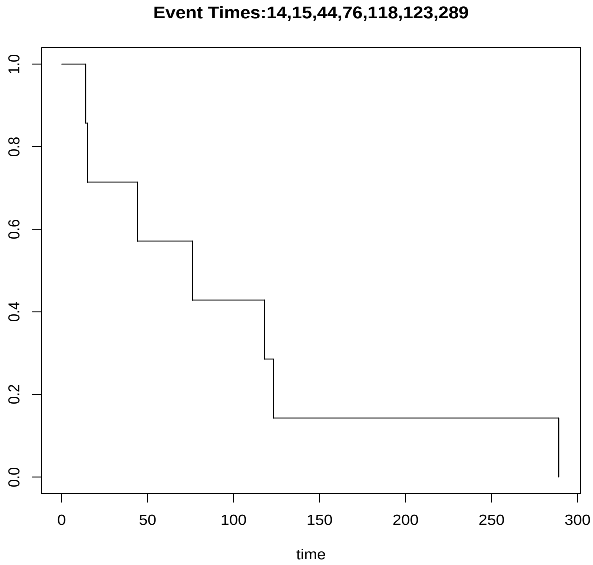
X_1	X_2	X_3	X_4	X_5	X_6	X_7
14	15	44	76	118	123	289

Derive Kaplan-Meier estimate graphically.


Event Times:14,15,44,76,118,123,289
Subj. 1
Subj. 2
Subj. 3
Subj. 4
Subj. 5
Subj. 6
Subj. 7
0
50
100
150
200
250
300
time

Event Times:14,15,44,76,118,123,289





Event Times:14,15,44,76,118,123,289

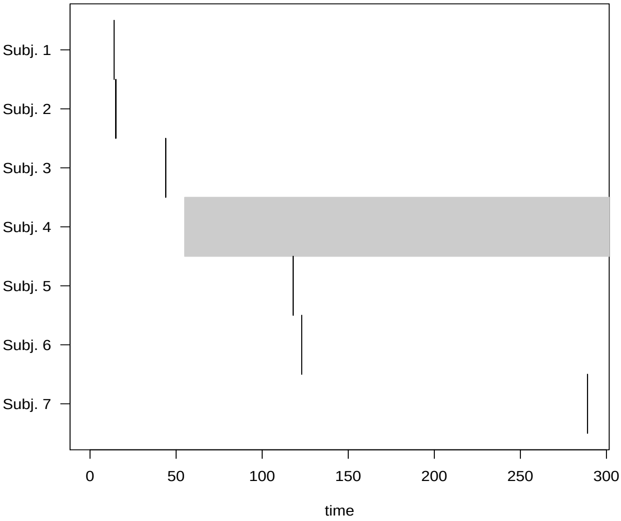
Example: Right-censoring

Original:	14	15	44	76	118	123	289
Modification 1:	14	15	44	55^+	118	123	289

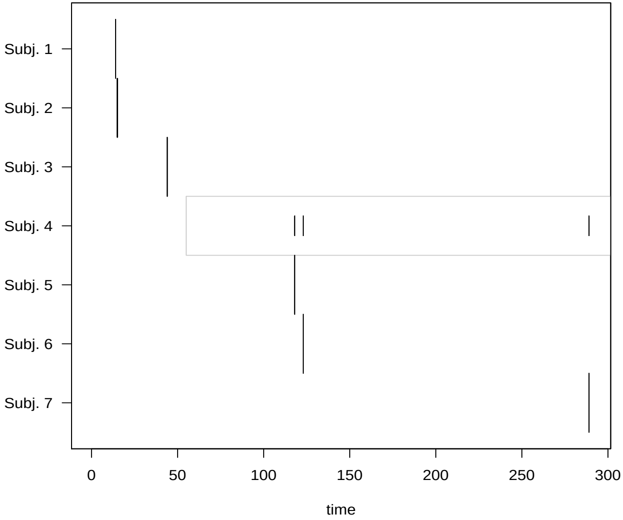
Modification 1: Fourth subject right-censored at 55.

Graphically show Efron's (1967) redistribution-to-the-right algorithm.

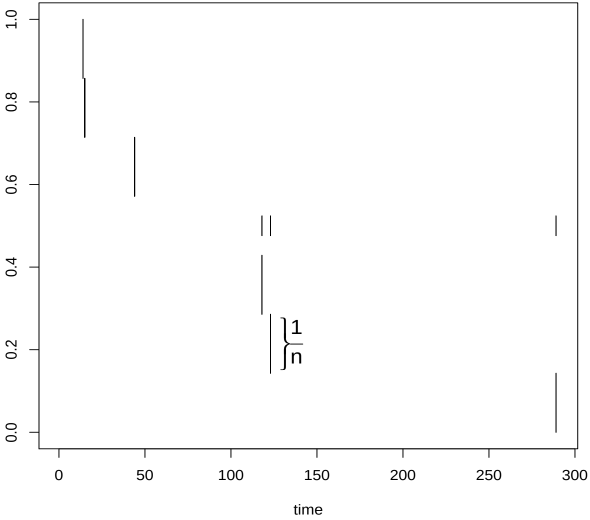
Event Times: 14,15, 44, 55+,118,123,289

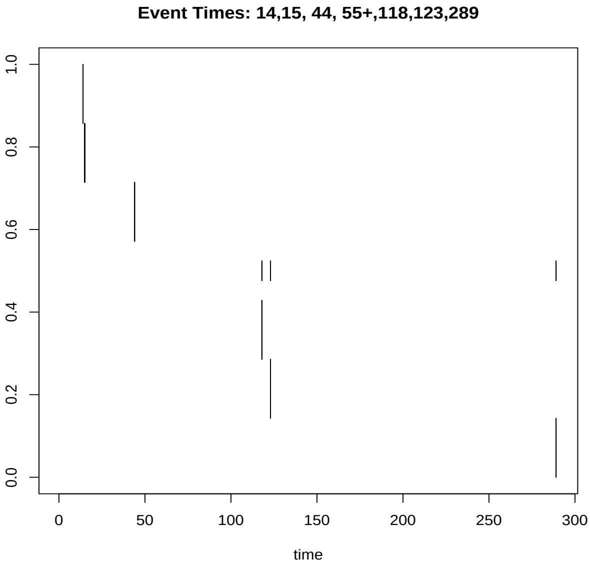


Event Times: 14,15, 44, 55+,118,123,289

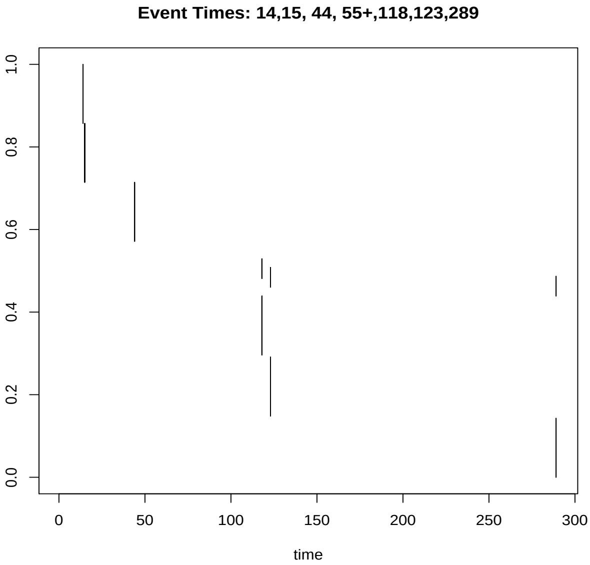


Event Times: 14,15, 44, 55+,118,123,289



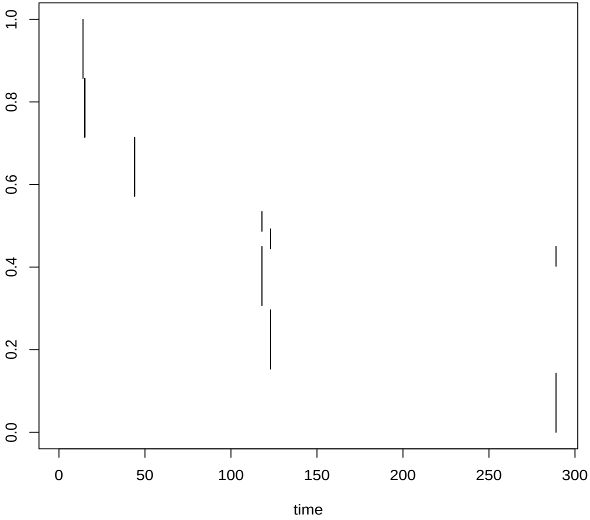


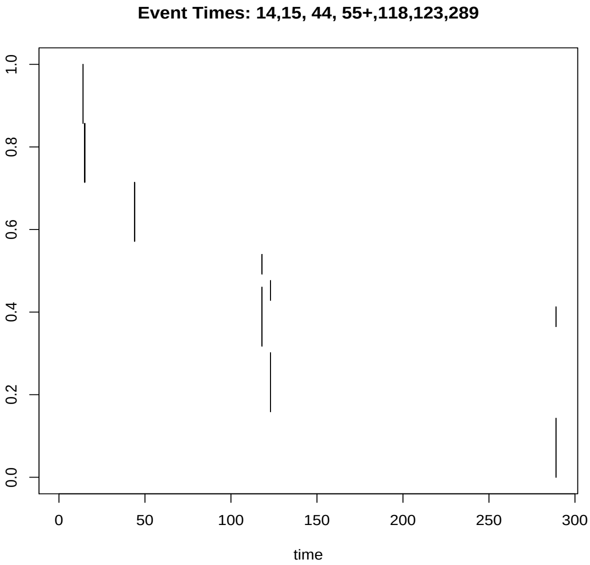
Event Times: 14,15, 44, 55+,118,123,289

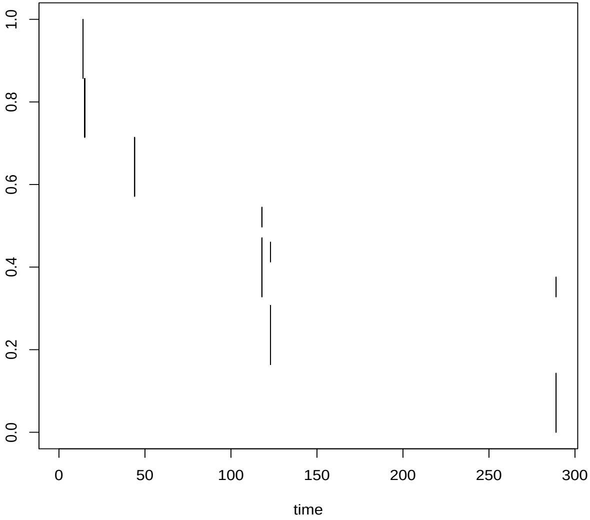


Event Times: 14,15, 44, 55+,118,123,289

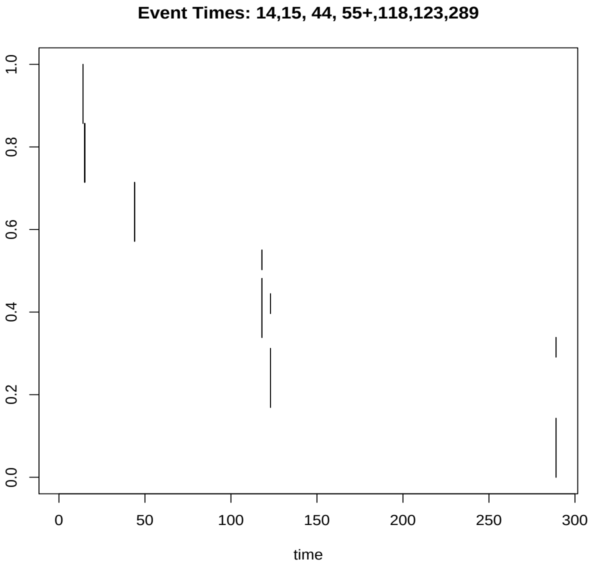
Event Times: 14,15, 44, 55+,118,123,289



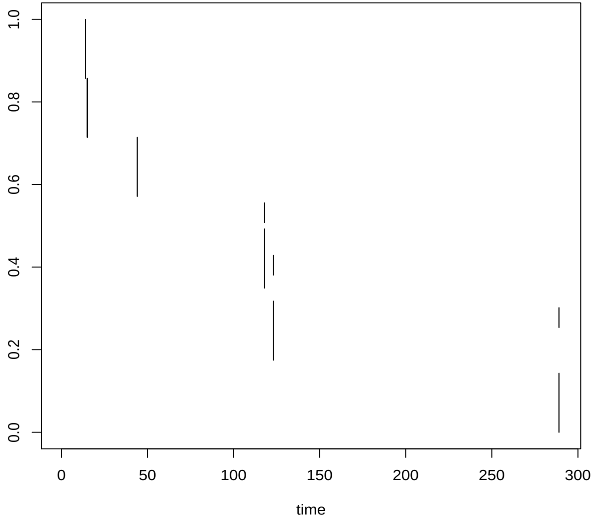

Event Times: 14,15, 44, 55+,118,123,289
0.0
0.2
0.4
0.6
0.8
1.0
0
50
100
150
200
250
300
time



Event Times: 14,15, 44, 55+,118,123,289

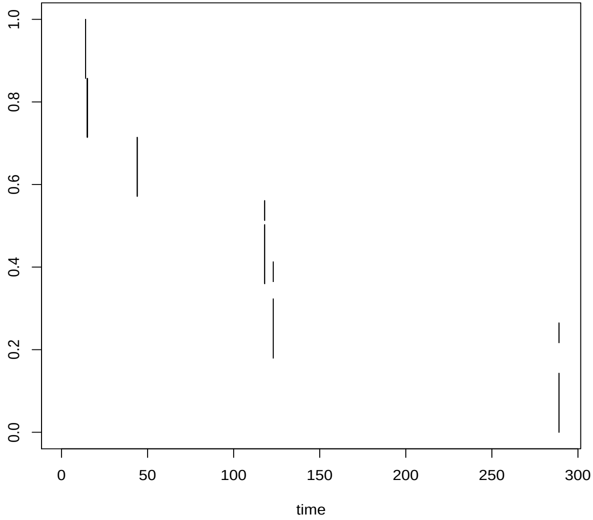


Event Times: 14,15, 44, 55+,118,123,289

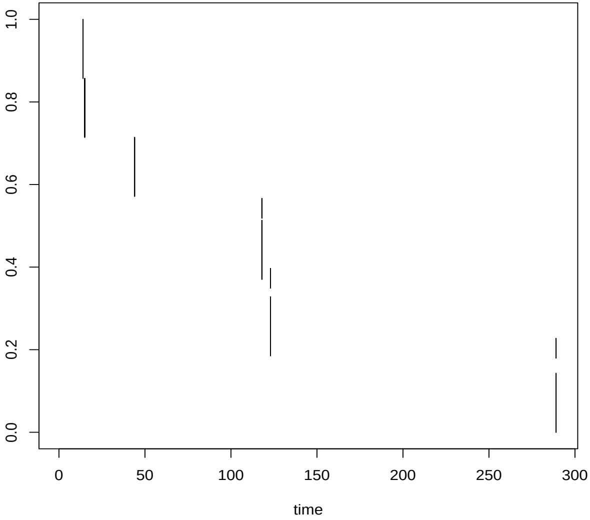


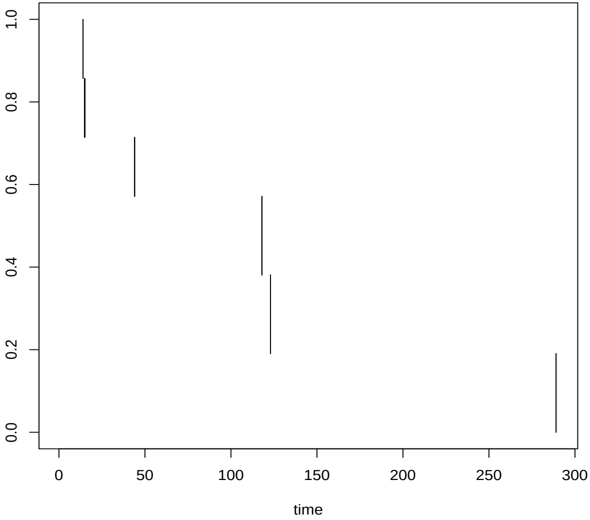
Event Times: 14,15, 44, 55+,118,123,289

Event Times: 14,15, 44, 55+,118,123,289

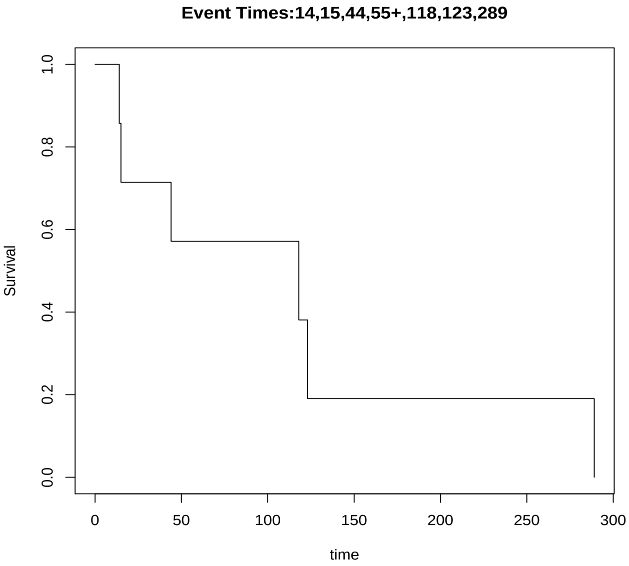


Event Times: 14,15, 44, 55+,118,123,289





Event Times: 14,15, 44, 55+,118,123,289


Event Times:14,15,44,55+,118,123,289

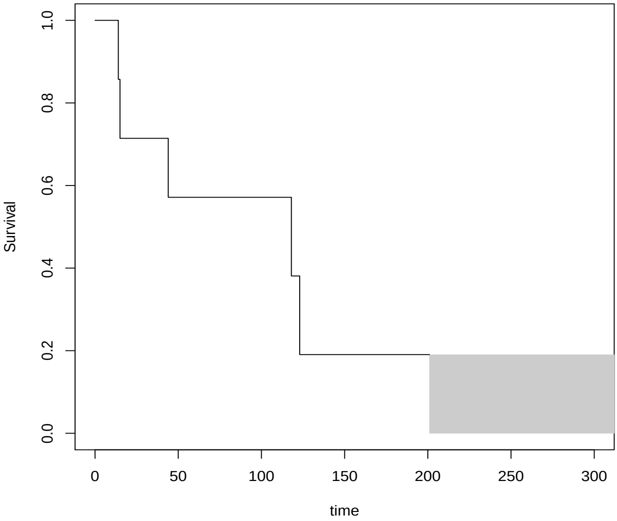
Example: Right-censoring

Consider 5 observed event times, 2 right-censored:

Original:	14	15	44	76	118	123	289
Modification 1:	14	15	44	55^+	118	123	289
Modification 2:	14	15	44	55^+	118	123	201^+

Modification 2: first subject right-censored at 55 and sixth subject right-censored at 201.

Event Times:14,15,44,55+,118,123,201+



Kaplan-Meier (Product-Limit) Estimator

Continuity Assumption:

- For mathematical convenience we assume event times are continuous.

Kaplan-Meier (Product-Limit) Estimator

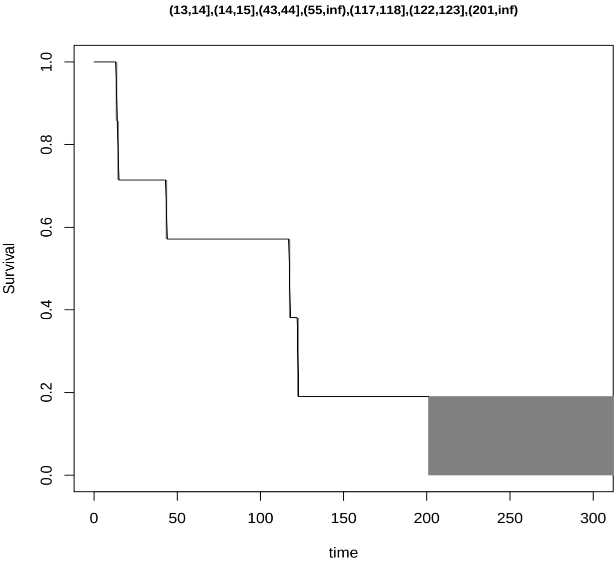
Continuity Assumption:

- For mathematical convenience we assume event times are continuous.
- With continuity assumption, Kaplan-Meier estimate is uniquely defined everywhere except after last observation if censored.

Kaplan-Meier (Product-Limit) Estimator

Continuity Assumption:

- For mathematical convenience we assume event times are continuous.
- With continuity assumption, Kaplan-Meier estimate is uniquely defined everywhere except after last observation if censored.
- Without continuity assumption, NPMLE is undefined also within intervals of unit one.



(13,14],(14,15],(43,44],(55,inf),(117,118],(122,123],(201,inf)

Kaplan-Meier (Product-Limit) Estimator

Bias:

- Assuming non-informative censoring...
- Kaplan-Meier estimate is approximately unbiased.

Kaplan-Meier (Product-Limit) Estimator

Bias:

- Assuming non-informative censoring...
- Kaplan-Meier estimate is approximately unbiased.
  - Asymptotically unbiased in $(0, \tau]$, where τ is largest possible observed event.

Kaplan-Meier (Product-Limit) Estimator

Bias:

- Assuming non-informative censoring...
- Kaplan-Meier estimate is approximately unbiased.
  - Asymptotically unbiased in $(0, \tau]$, where τ is largest possible observed event.
  - Bias comes entirely from how define region of K-M estimator after last observation if censored (see Gillespie, et al, 1992, Biometrika, 149-55).

Kaplan-Meier (Product-Limit) Estimator

Bias:

- Assuming non-informative censoring...
- Kaplan-Meier estimate is approximately unbiased.
  - Asymptotically unbiased in $(0, \tau]$, where τ is largest possible observed event.
  - Bias comes entirely from how define region of K-M estimator after last observation if censored (see Gillespie, et al, 1992, Biometrika, 149-55).
  - If define that region as 0 (Efron, 1967), then negative bias.

Kaplan-Meier (Product-Limit) Estimator

Bias:

- Assuming non-informative censoring...
- Kaplan-Meier estimate is approximately unbiased.
  - Asymptotically unbiased in $(0, \tau]$, where τ is largest possible observed event.
  - Bias comes entirely from how define region of K-M estimator after last observation if censored (see Gillespie, et al, 1992, Biometrika, 149-55).
  - If define that region as 0 (Efron, 1967), then negative bias.
  - If define that region as continuing with K-M unchanged after last observation if censored (Gill, 1980), then positive bias.

Interval Censoring: NPMLE

Likelihood:

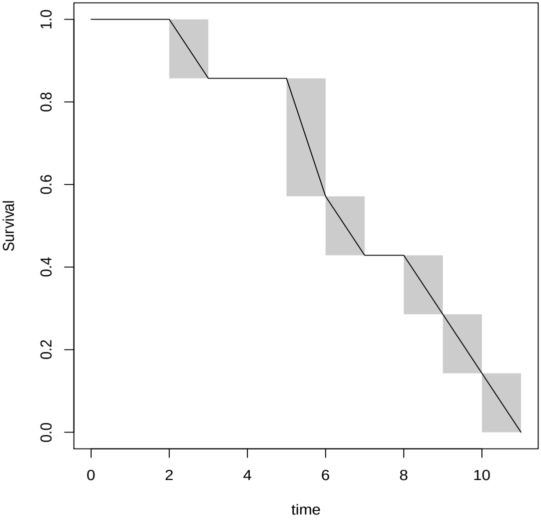
$$\ell = \prod_{i=1}^{n} \{F(R_i) - F(L_i)\}$$

where F is the cumulative distribution function.
Survival distribution is $S(t) = 1 - F(t)$.

Interval Censoring

Example, Regular Observation Times:

Subject Number	L	R
1	2	3
2	5	6
3	9	10
4	10	11
5	5	6
6	6	7
7	8	9


Survival
0.0
0.2
0.4
0.6
0.8
1.0
0
2
4
6
8
10
time

E-M (Expectation-Maximization) Algorithm for NPMLE

Notation:

- X_i event time for ith subject
- $(L_i, R_i]$ observed interval for ith subject
- $t_1, t_2, \ldots, t_m$, set of possible observation times where NPMLE may change. (Describe with pictures).

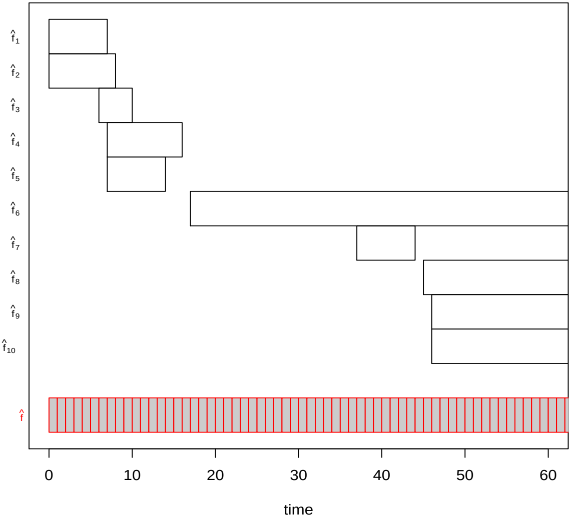
Irregular Observation Times

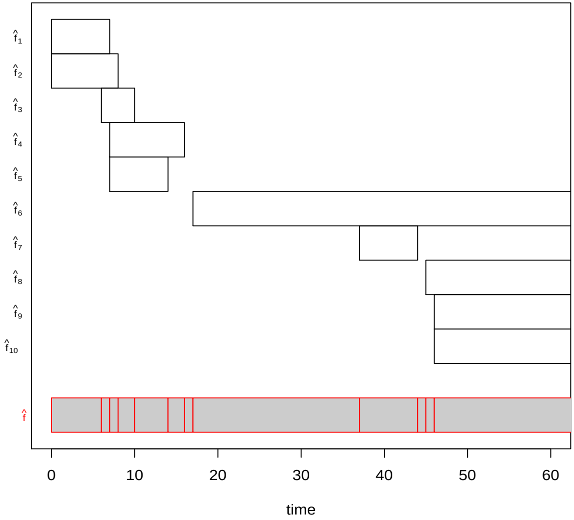
Data (sorted by L_i):

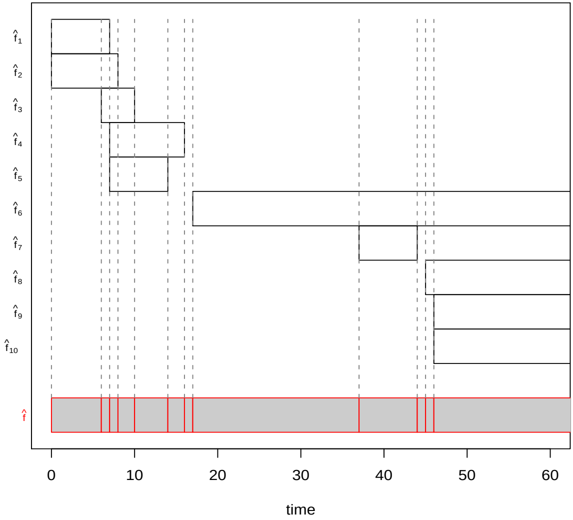
Subject Number	L	R
1	0	7
2	0	8
3	6	10
4	7	16
5	7	14
6	17	∞
7	37	44
8	45	∞
9	46	∞
10	46	∞

Possible change times (later discuss how we do not need all):

$t_0 = 0$, $t_1 = 6$, $t_2 = 7$, $t_3 = 8$, $t_4 = 10$, $t_5 = 14$, $t_6 = 16$, $t_7 = 17$, $t_8 = 37$, $t_9 = 44$, $t_{10} = 45$, $t_{11} = 46$, $t_{12} = \infty$


$\hat{f}_1$
$\hat{f}_2$
$\hat{f}_3$
$\hat{f}_4$
$\hat{f}_5$
$\hat{f}_6$
$\hat{f}_7$
$\hat{f}_8$
$\hat{f}_9$
$\hat{f}_{10}$
$\hat{f}$
0
10
20
30
40
50
60
time


$\hat{f}_1$
$\hat{f}_2$
$\hat{f}_3$
$\hat{f}_4$
$\hat{f}_5$
$\hat{f}_6$
$\hat{f}_7$
$\hat{f}_8$
$\hat{f}_9$
$\hat{f}_{10}$
$\hat{f}$
0
10
20
30
40
50
60
time


$\hat{f}_1$
$\hat{f}_2$
$\hat{f}_3$
$\hat{f}_4$
$\hat{f}_5$
$\hat{f}_6$
$\hat{f}_7$
$\hat{f}_8$
$\hat{f}_9$
$\hat{f}_{10}$
$\hat{f}$
0
10
20
30
40
50
60
time

E-M Algorithm for NPMLE

1. Mathematical Notation (for clarity, precision)
2. Graphically (for intuition)

E-M Algorithm for NPMLE

Notation:

- X_i event time for ith subject
- $(L_i, R_i]$ observed interval for ith subject
- $t_1, t_2, \ldots, t_m$, set of possible observation times where NPMLE may change.

E-M Algorithm for NPMLE

Notation:

- X_i event time for ith subject
- $(L_i, R_i]$ observed interval for ith subject
- $t_1, t_2, \ldots, t_m$, set of possible observation times where NPMLE may change.
- Let $t_0 \equiv 0$ and $t_{m+1} \equiv \infty$.
- $p(t_j) = Pr[t_{j-1} < X \leq t_j]$
- $\mathbf{p} = [p(t_1), p(t_2), \ldots, p(t_{m+1})]$

E-M Algorithm for NPMLE

Notation:

- X_i event time for ith subject
- $(L_i, R_i]$ observed interval for ith subject
- $t_1, t_2, \ldots, t_m$, set of possible observation times where NPMLE may change.
- Let $t_0 \equiv 0$ and $t_{m+1} \equiv \infty$.
- $p(t_j) = Pr[t_{j-1} < X \leq t_j]$
- $\mathbf{p} = [p(t_1), p(t_2), \ldots, p(t_{m+1})]$
- $p_i(t_j) = Pr\left[t_{j-1} < X_i \leq t_j \mid X_i \in (L_i, R_i]\right]$

E-M Algorithm for NPMLE

- start with initial estimate of $\hat{\mathbf{p}}$.
  - Each of $m+1$ elements should be positive, and all should sum to 1.
  - Example: $\hat{\mathbf{p}} = [\frac{1}{m+1}, \frac{1}{m+1}, \ldots, \frac{1}{m+1}]$

E-M Algorithm for NPMLE

- start with initial estimate of $\hat{\mathbf{p}}$.
    - Each of $m+1$ elements should be positive, and all should sum to 1.
    - Example: $\hat{\mathbf{p}} = [\frac{1}{m+1}, \frac{1}{m+1}, \ldots, \frac{1}{m+1}]$
- E-step: for each i
    - $\hat{p}_i(t_j) = \frac{\hat{p}(t_j) I\{t_j \in (L_i, R_i]\}}{\sum_{t_k \in (L_i, R_i]} \hat{p}(t_k)}$

E-M Algorithm for NPMLE

- start with initial estimate of $\hat{\mathbf{p}}$.
  - Each of $m+1$ elements should be positive, and all should sum to 1.
  - Example: $\hat{\mathbf{p}} = [\frac{1}{m+1}, \frac{1}{m+1}, \dots, \frac{1}{m+1}]$
- E-step: for each i
  - $\hat{p}_i(t_j) = \frac{\hat{p}(t_j) I\{t_j \in (L_i, R_i]\}}{\sum_{t_k \in (L_i, R_i]} \hat{p}(t_k)}$
  - Example: $L_i = t_1, R_i = t_3$,

$$\hat{p}_i = \left[0, \frac{\hat{p}(t_2)}{\hat{p}(t_2) + \hat{p}(t_3)}, \frac{\hat{p}(t_3)}{\hat{p}(t_2) + \hat{p}(t_3)}, 0, \dots, 0\right]$$

E-M Algorithm for NPMLE

- start with initial estimate of $\hat{\mathbf{p}}$.
    - Each of $m+1$ elements should be positive, and all should sum to 1.
    - Example: $\hat{\mathbf{p}} = [\frac{1}{m+1}, \frac{1}{m+1}, \dots, \frac{1}{m+1}]$
- E-step: for each i
    - $\hat{p}_i(t_j) = \frac{\hat{p}(t_j) I\{t_j \in (L_i, R_i]\}}{\sum_{t_k \in (L_i, R_i]} \hat{p}(t_k)}$
    - Example: $L_i = t_1, R_i = t_3$,

$$\hat{p}_i = \left[0, \frac{\hat{p}(t_2)}{\hat{p}(t_2) + \hat{p}(t_3)}, \frac{\hat{p}(t_3)}{\hat{p}(t_2) + \hat{p}(t_3)}, 0, \dots, 0\right]$$

- M-step: update $\hat{\mathbf{p}}$. For each j
    - $\hat{p}(t_j) = \frac{1}{n} \sum_{i=1}^{n} \hat{p}_i(t_j)$

E-M Algorithm for NPMLE

- start with initial estimate of $\hat{\mathbf{p}}$.
  - Each of $m+1$ elements should be positive, and all should sum to 1.
  - Example: $\hat{\mathbf{p}} = [\frac{1}{m+1}, \frac{1}{m+1}, \dots, \frac{1}{m+1}]$
- E-step: for each i
  - $\hat{p}_i(t_j) = \frac{\hat{p}(t_j) I\{t_j \in (L_i, R_i]\}}{\sum_{t_k \in (L_i, R_i]} \hat{p}(t_k)}$
  - Example: $L_i = t_1, R_i = t_3$,

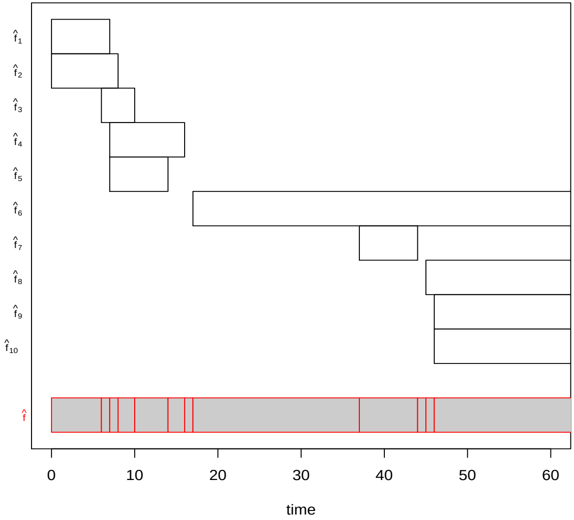
$$\hat{p}_i = \left[0, \frac{\hat{p}(t_2)}{\hat{p}(t_2) + \hat{p}(t_3)}, \frac{\hat{p}(t_3)}{\hat{p}(t_2) + \hat{p}(t_3)}, 0, \dots, 0\right]$$

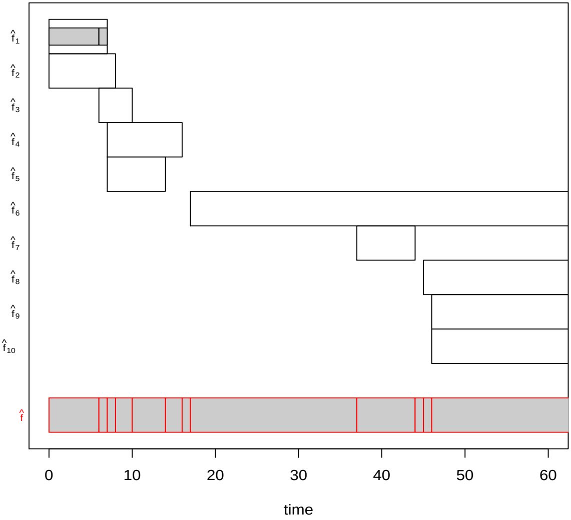
- M-step: update $\hat{\mathbf{p}}$. For each j
  - $\hat{p}(t_j) = \frac{1}{n} \sum_{i=1}^{n} \hat{p}_i(t_j)$
- Iterate until convergence.

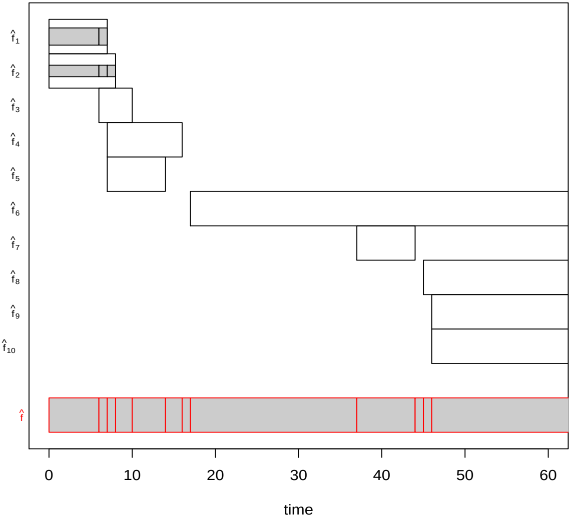
Irregular Observation Times

Data (sorted by L_i):

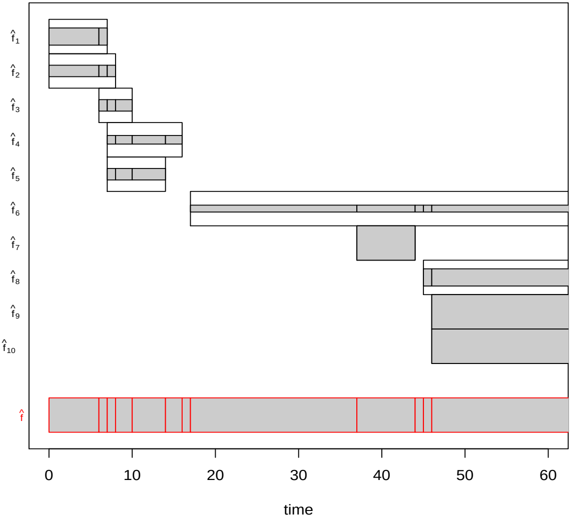
Subject Number	L	R
1	0	7
2	0	8
3	6	10
4	7	16
5	7	14
6	17	∞
7	37	44
8	45	∞
9	46	∞
10	46	∞


$\hat{f}_1$
$\hat{f}_2$
$\hat{f}_3$
$\hat{f}_4$
$\hat{f}_5$
$\hat{f}_6$
$\hat{f}_7$
$\hat{f}_8$
$\hat{f}_9$
$\hat{f}_{10}$
$\hat{f}$
0
10
20
30
40
50
60
time

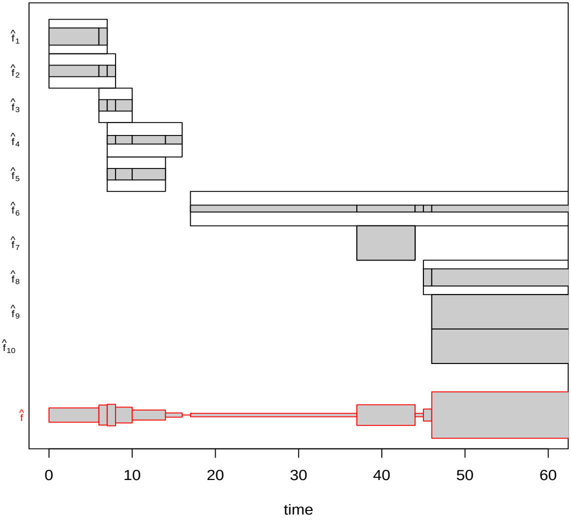

$\hat{f}_1$
$\hat{f}_2$
$\hat{f}_3$
$\hat{f}_4$
$\hat{f}_5$
$\hat{f}_6$
$\hat{f}_7$
$\hat{f}_8$
$\hat{f}_9$
$\hat{f}_{10}$
$\hat{f}$
0
10
20
30
40
50
60
time


$\hat{f}_1$
$\hat{f}_2$
$\hat{f}_3$
$\hat{f}_4$
$\hat{f}_5$
$\hat{f}_6$
$\hat{f}_7$
$\hat{f}_8$
$\hat{f}_9$
$\hat{f}_{10}$
$\hat{f}$
0
10
20
30
40
50
60
time

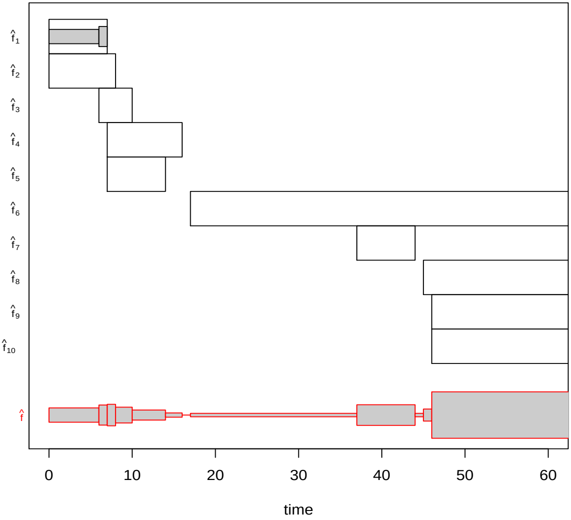
iteration=1, E−step

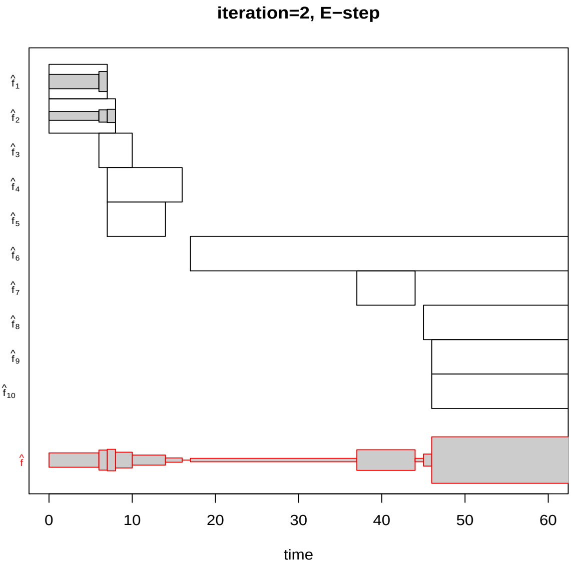


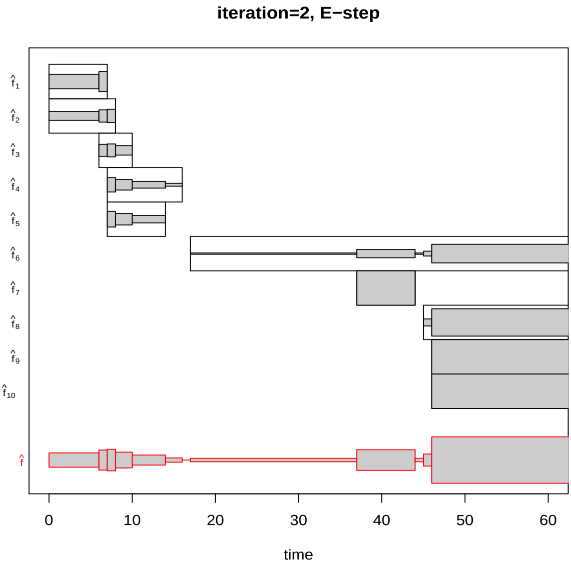
iteration=1, M−step



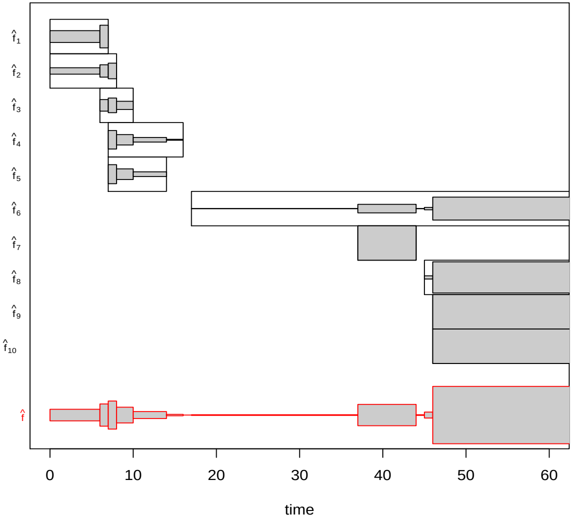
iteration=2, E−step



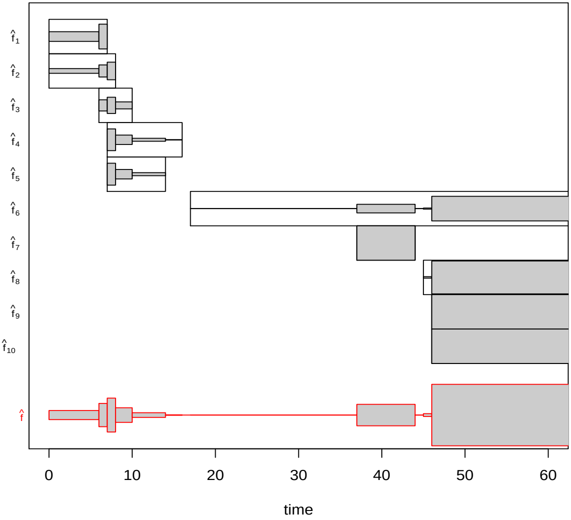

iteration=2, E−step
$\hat{f}_1$
$\hat{f}_2$
$\hat{f}_3$
$\hat{f}_4$
$\hat{f}_5$
$\hat{f}_6$
$\hat{f}_7$
$\hat{f}_8$
$\hat{f}_9$
$\hat{f}_{10}$
$\hat{f}$
0
10
20
30
40
50
60
time


iteration=2, E−step
$\hat{f}_1$
$\hat{f}_2$
$\hat{f}_3$
$\hat{f}_4$
$\hat{f}_5$
$\hat{f}_6$
$\hat{f}_7$
$\hat{f}_8$
$\hat{f}_9$
$\hat{f}_{10}$
$\hat{f}$
0
10
20
30
40
50
60
time

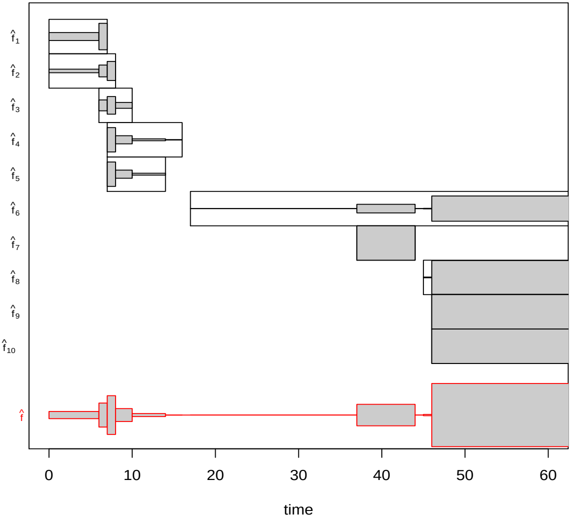
iteration=3, E−step



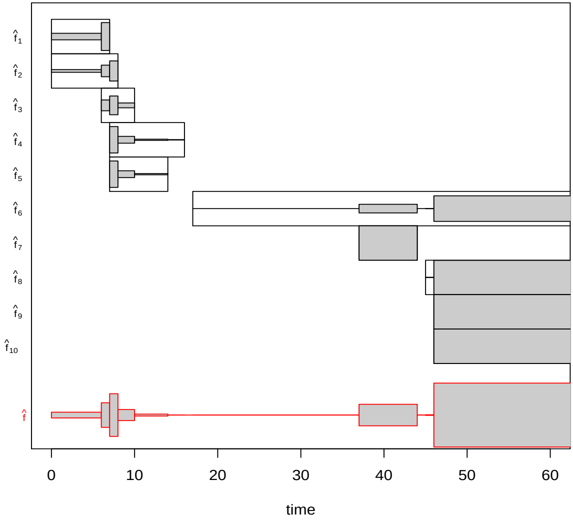
iteration=4, E−step



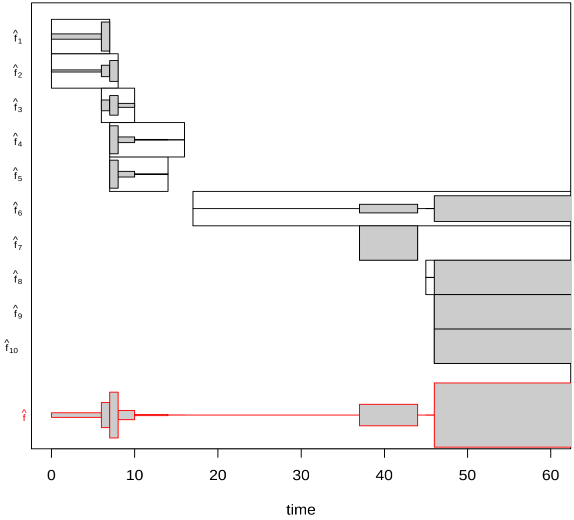
iteration=5, E−step



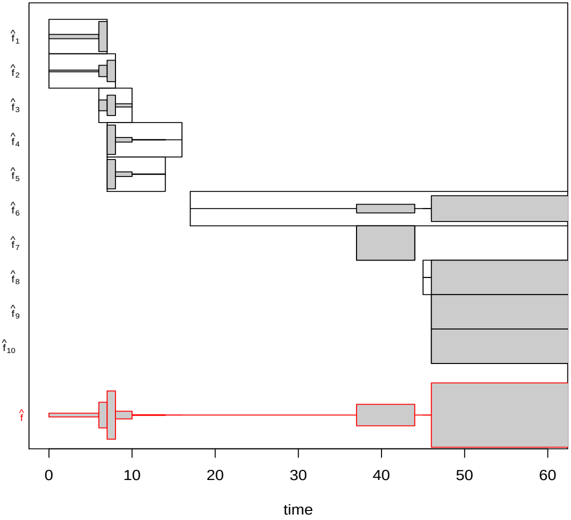
iteration=6, E−step



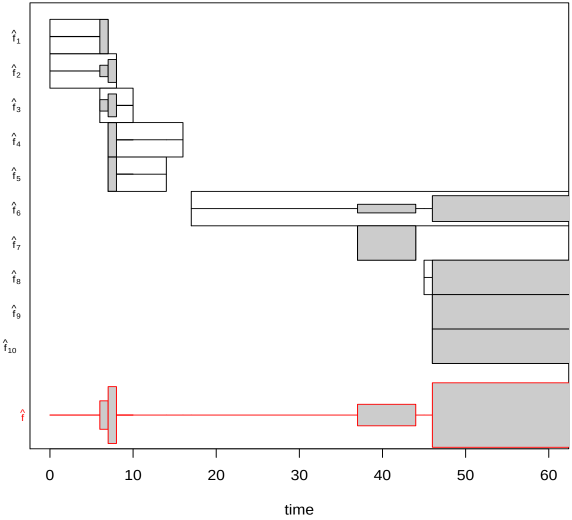
iteration=7, E−step



iteration=8, E−step



iteration=20



Turnbull intervals

Turnbull (1976, JRSS-B, 290-295)

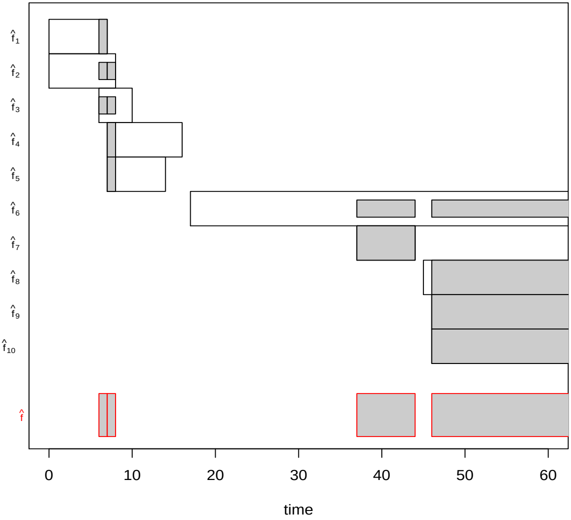
Turnbull intervals:

- Also called innermost intervals and real representations of maximal cliques.
- Set of disjoint intervals whose left endpoints are in $\mathbf{L} = \{L_1, L_2, \ldots, L_n\}$ and right endpoints are in $\mathbf{R} = \{R_1, R_2, \ldots, R_n\}$ but contain no other members of $\mathbf{L}$ or $\mathbf{R}$ except the endpoints.

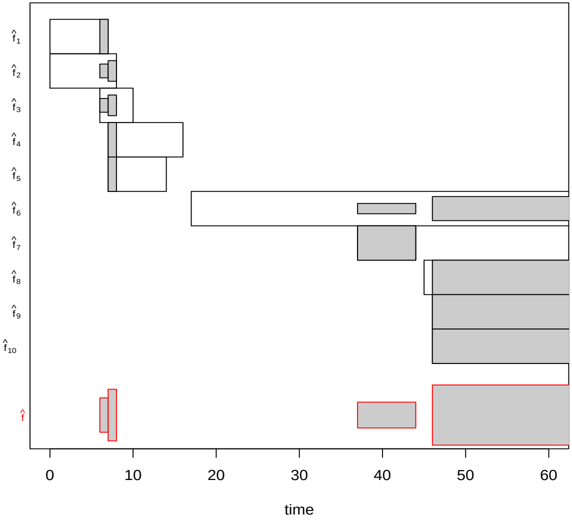
Turnbull intervals

Subj													
#	t_0	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_{11}	t_{12}
1	0		7										
2	0			8									
3		6			10								
4			7				16						
5			7			14							
6								17					∞
7									37	44			
8											45		∞
9												46	∞
10												46	∞
		(6	7](7	8]					(37	44]		(46	∞)

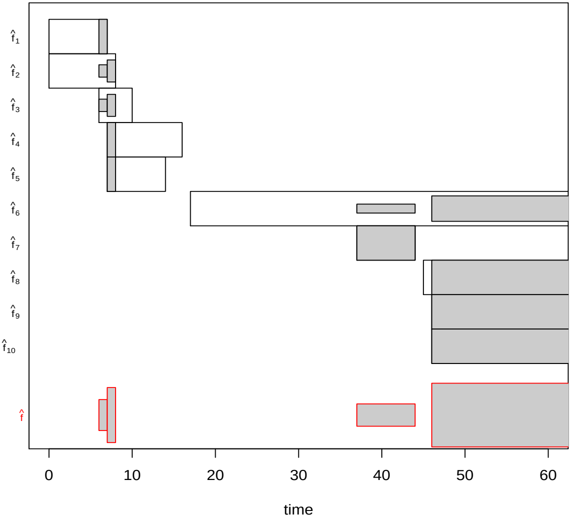
iteration=1, E−step



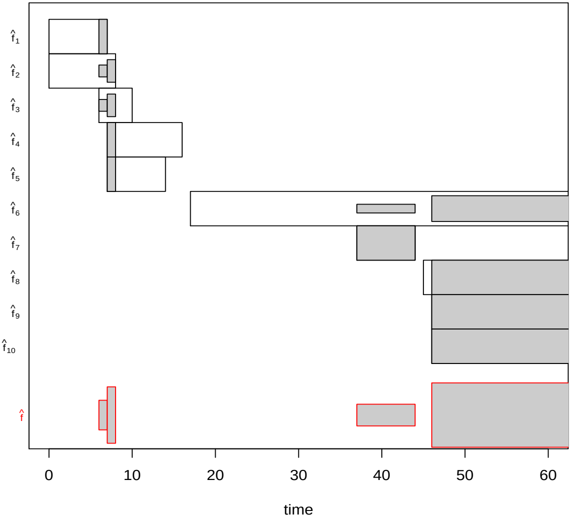
iteration=2, E−step



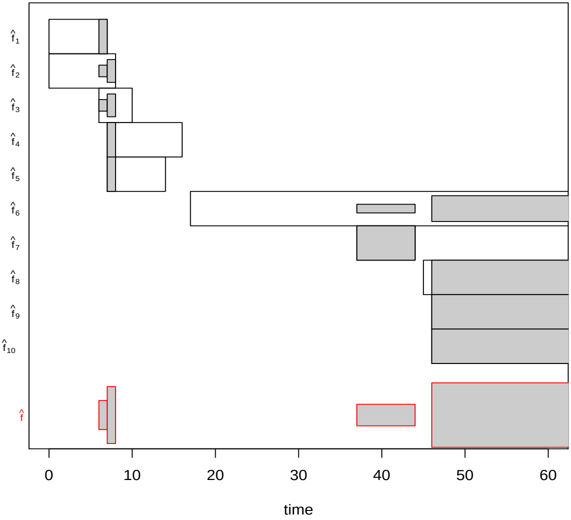
iteration=3, E−step

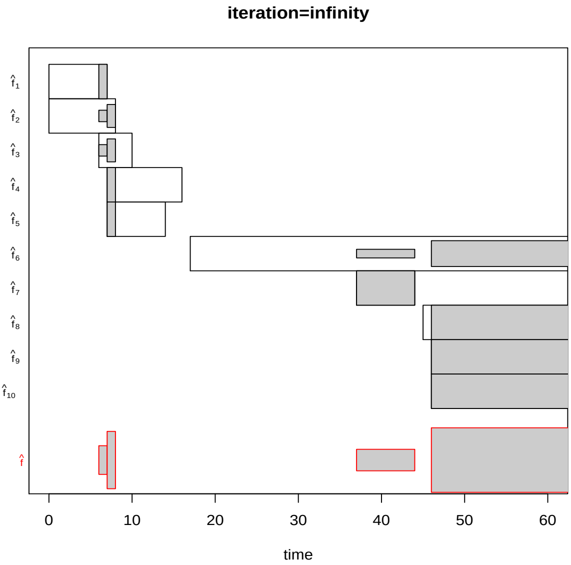


iteration=4, E−step



iteration=5, E−step



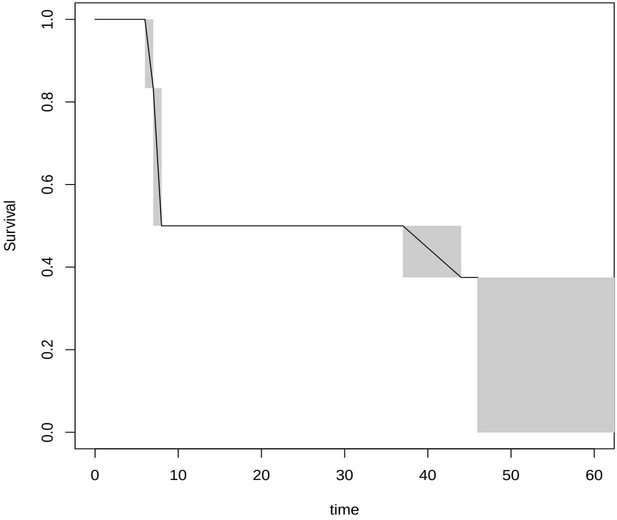


iteration=infinity
$\hat{f}_1$
$\hat{f}_2$
$\hat{f}_3$
$\hat{f}_4$
$\hat{f}_5$
$\hat{f}_6$
$\hat{f}_7$
$\hat{f}_8$
$\hat{f}_9$
$\hat{f}_{10}$
$\hat{f}$
0
10
20
30
40
50
60
time

R code

```
library(interval)
data(bcos)
L<-bcos[1:10,"left"]
R<-bcos[1:10,"right"]

plot(icfit(L,R))
```


Survival
1.0
0.8
0.6
0.4
0.2
0.0
0
10
20
30
40
50
60
time

Continuous Inspection Processes: Theory

- If you have a continuous inspection process, how fast does the NPMLE converge?
- Case 1 Censoring (only 1 inspection for each subject)
- NPMLE converges at $n^{1/3}$ rate.

Continuous Inspection Processes: Theory

- If you have a continuous inspection process, how fast does the NPMLE converge?
- Case 1 Censoring (only 1 inspection for each subject)
- NPMLE converges at $n^{1/3}$ rate.
- Details: As $n \to \infty$

  $$n^{1/3} \frac{\hat{S}(t) - S(t)}{\left(\frac{1}{2} S(t)(1 - S(t)) f(t)/g(t)\right)^{1/3}}$$

  converges in distribution to a (non-normal) random variable. Here $f(t)$ and $g(t)$ are density functions of event time and inspection time respectively.
- Groeneboom and Wellner (1992, Information Bounds and NPMLE, Birkhäuser)

Continuous Inspection Processes: Theory

- If you have a continuous inspection process, how fast does the NPMLE converge?
- Case 1 Censoring (only 1 inspection for each subject)
- NPMLE converges at $n^{1/3}$ rate.
- Details: As $n \to \infty$

  $$n^{1/3} \frac{\hat{S}(t) - S(t)}{\left(\frac{1}{2} S(t)(1 - S(t)) f(t)/g(t)\right)^{1/3}}$$

  converges in distribution to a (non-normal) random variable. Here $f(t)$ and $g(t)$ are density functions of event time and inspection time respectively.
- Groeneboom and Wellner (1992, Information Bounds and NPMLE, Birkhäuser)
- Case K Censoring (hard problem, see e.g., Schick and Yu, 2000, Scan. J. Stat 45-55).

Algorithms for NPMLE of Interval Censored Data

- E-M or Self-consistent algorithm (Turnbull, 1976, JRSSB, 290-5, Dempster, Laird, and Rubin, 1977, JRSS, 1-38).

Algorithms for NPMLE of Interval Censored Data

- E-M or Self-consistent algorithm (Turnbull, 1976, JRSSB, 290-5, Dempster, Laird, and Rubin, 1977, JRSS, 1-38).
- E-M, polish using Kuhn-Tucker conditions (Gentleman and Geyer, 1996, Biometrika, 618-23, interval R Package, SAS, Proc Lifereg).

Algorithms for NPMLE of Interval Censored Data

- E-M or Self-consistent algorithm (Turnbull, 1976, JRSSB, 290-5, Dempster, Laird, and Rubin, 1977, JRSS, 1-38).
- E-M, polish using Kuhn-Tucker conditions (Gentleman and Geyer, 1996, Biometrika, 618-23, interval R Package, SAS, Proc Lifereg).
- Gentleman and Vandal (2001, JCGS, 403-421, Icens R package)
  - Iterative Convex Minorant (ICM) Algorithm (Groeneboom and Wellner, 1992, *Information Bounds and Nonparametric Maximum Likelihood Estimation*).
  - Hybrid, E-M and ICM (Wellner and Zhan, 1997, JASA, 945-957).
  - Vector Exchange Algorithm (Böhning, 1986, Metrika, 337-347)
  - Intra-simplex direction (Lesperance and Kalbfleisch, 1992, JASA, 120-6).

Algorithms for NPMLE of Interval Censored Data

- E-M or Self-consistent algorithm (Turnbull, 1976, JRSSB, 290-5, Dempster, Laird, and Rubin, 1977, JRSS, 1-38).
- E-M, polish using Kuhn-Tucker conditions (Gentleman and Geyer, 1996, Biometrika, 618-23, interval R Package, SAS, Proc Lifereg).
- Gentleman and Vandal (2001, JCGS, 403-421, Icens R package)
  - Iterative Convex Minorant (ICM) Algorithm (Groeneboom and Wellner, 1992, *Information Bounds and Nonparametric Maximum Likelihood Estimation*).
  - Hybrid, E-M and ICM (Wellner and Zhan, 1997, JASA, 945-957).
  - Vector Exchange Algorithm (Böhning, 1986, Metrika, 337-347)
  - Intra-simplex direction (Lesperance and Kalbfleisch, 1992, JASA, 120-6).
- Support Reduction Algorithm (Groeneboom, Jongbloed, Wellner, 2008, Scan J Stat, 385-, MLEcens R package)

R package: interval

```
> library(interval)
> data(bcos)
> head(bcos)
```

```
  left right treatment
1   45   Inf       Rad
2    6    10       Rad
3    0     7       Rad
4   46   Inf       Rad
5   46   Inf       Rad
6    7    16       Rad
```

Two treatments for breast cancer, radiation (Rad, n=46) and radiation with chemotherapy (RadChem, n=48). Response is time in months until breast retraction. Finkelstein and Wolfe (1985, Biometrics, 845-).

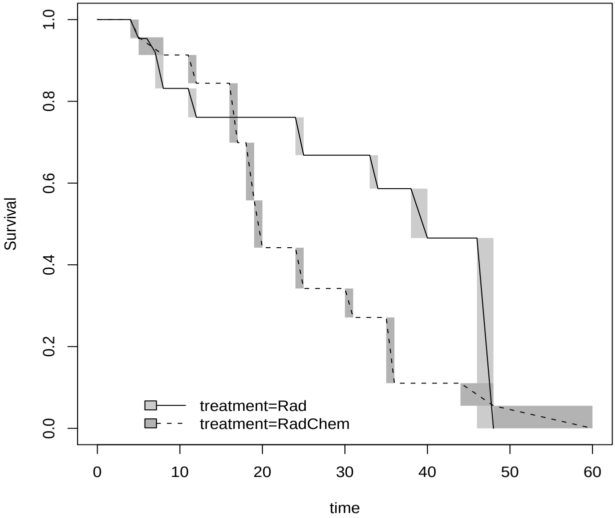
R package interval: icfit function

- icfit function calculates NPMLE by E-M algorithm
- default calls MLEcens package to calculate initial estimate.
  - MLEcens developed for bivariate interval censored data, but can be used in univariate case
  - uses Support Reduction algorithm, written in C (very fast).
- icfit checks the Kuhn-Tucker conditions

R package: interval

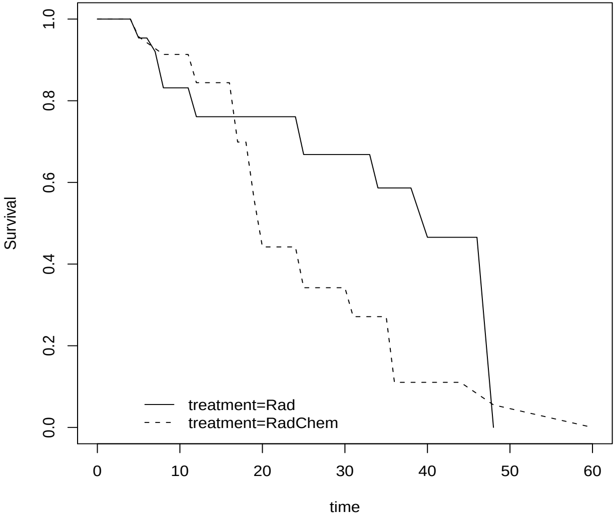
```
> fit<-icfit(Surv(left,right,type="interval2")~treatment,
    data=bcos)
> summary(fit)
```

```
treatment=Rad:
  Interval Probability
1    (4,5]      0.0463
2    (6,7]      0.0334
3    (7,8]      0.0887
4  (11,12]      0.0708
5  (24,25]      0.0926
6  (33,34]      0.0818
7  (38,40]      0.1209
8  (46,48]      0.4656
treatment=RadChem:
   Interval Probability
1     (4,5]      0.0433
2     (5,8]      0.0433
3   (11,12]      0.0692
4   (16,17]      0.1454
5   (18,19]      0.1411
6   (19,20]      0.1157
7   (24,25]      0.0999
8   (30,31]      0.0709
9   (35,36]      0.1608
10  (44,48]      0.0552
11  (48,60]      0.0552
> plot(fit)
```

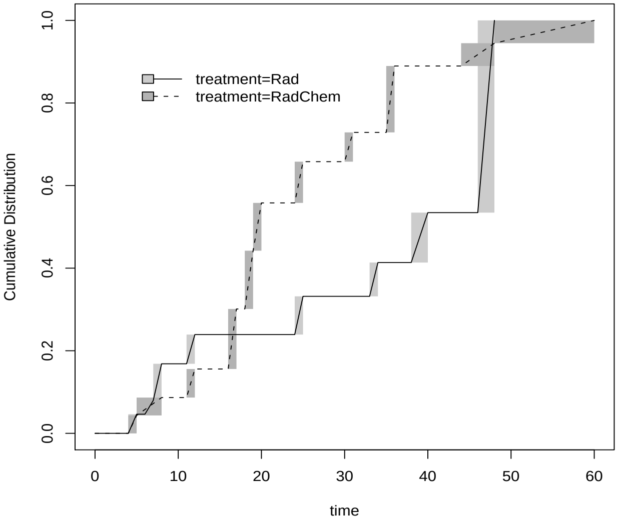


Survival
time
0.0
0.2
0.4
0.6
0.8
1.0
0
10
20
30
40
50
60
treatment=Rad
treatment=RadChem

```
> plot(fit,shade=FALSE)
```



```
> plot(fit, dtype="cdf")
```



```
> fit[1]

$strata
treatment=Rad
            8

$error
[1] 2.067097e-09

$numit
[1] 1

$pf
     (4,5]      (6,7]      (7,8]    (11,12]    (24,25]    (33,34]    (38,40]
0.04634677 0.03336337 0.08866737 0.07075292 0.09264584 0.08178576 0.12087983
   (46,48]
0.46555814

$intmap
     [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]
[1,]    4    6    7   11   24   33   38   46
[2,]    5    7    8   12   25   34   40   48
attr(,"LRin")
      [,1]  [,2]  [,3]  [,4]  [,5]  [,6]  [,7]  [,8]
[1,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[2,]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE

$converge
[1] TRUE

$message
[1] "normal convergence"

...
```

```
> summary(fit[1])
```

```
  Interval Probability
1    (4,5]      0.0463
2    (6,7]      0.0334
3    (7,8]      0.0887
4  (11,12]      0.0708
5  (24,25]      0.0926
6  (33,34]      0.0818
7  (38,40]      0.1209
8  (46,48]      0.4656
```

Using Different Starting Functions

```
> formula<-Surv(left,right,type="interval2")~treatment
> system.time(icfit(formula,data=bcos))
```

```
   user  system elapsed
   0.02    0.00    0.02
```

Warning: default calls MLEcens 0.1-3, can crash R!

```
> system.time(icfit(formula,data=bcos,initfit=NULL))
```

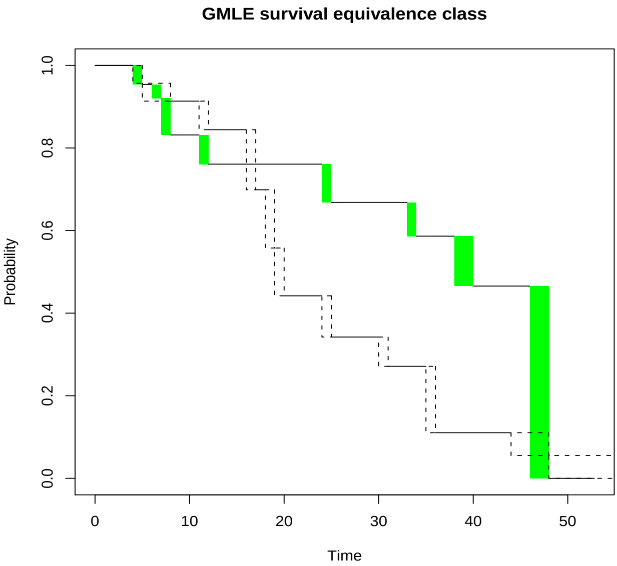
```
   user  system elapsed
   0.10    0.00    0.09
```

```
> system.time(icfit(formula,data=bcos,initfit="initEMICM"))
```

```
   user  system elapsed
   0.17    0.00    0.17
```

Icens package

```
> library(Icens)
> library(interval)
> data(bcos)
> bcosRad<-bcos[bcos$treatment=="Rad",c("left","right")]
> fitRad<-EMICM(bcosRad)
> bcosRadChem<-bcos[bcos$treatment=="RadChem",c("left","right")]
> fitRadChem<-EMICM(bcosRadChem)
> plot(fitRad,surv=TRUE)
> plot(fitRadChem,surv=TRUE,new=FALSE,shade=2)
```



survival package

- Can calculate NPMLE for interval censored data using E-M (i.e., Turnbull) algorithm, **but** ...
- assumes drop in survival occurs at midpoints,
- for confidence intervals of survival uses methods developed for right censored data (not sure of properties for interval censored data).

Distribution Estimation: Other Issues

- For these issues there are no R packages to solve them at this time.

Confidence Intervals on Survival Function

- Vandal, Gentleman, and Liu (2005, Can J. Stat, 71-83)
  - Bootstrap
  - Empirical likelihood
    - When inspection times are fixed, -2 times log-empirical likelihood ratio $\rightarrow$ Chi-square with 1 degree of freedom
    - when number of inspection times grows with sample size, more theory needed.

Interval Censoring with Truncation

- The event time, X, is truncated in B if the researcher would not have been aware of the existence of X had X not been in B.

Interval Censoring with Truncation

- The event time, X, is truncated in B if the researcher would not have been aware of the existence of X had X not been in B.
- Example (left-truncation): If the event time is length of time from diagnosis of cancer until death. Only individuals who live long enough after diagnosis to get in study are observed.

Interval Censoring with Truncation

- The event time, X, is truncated in B if the researcher would not have been aware of the existence of X had X not been in B.
- Example (left-truncation): If the event time is length of time from diagnosis of cancer until death. Only individuals who live long enough after diagnosis to get in study are observed.
- $$\ell = \prod_{i=1}^{n} \frac{F(R_i) - F(L_i)}{F(r_i) - F(l_i)}$$

  where $(l_i, r_i]$ is the truncating interval.

Interval Censoring with Truncation

- The event time, X, is truncated in B if the researcher would not have been aware of the existence of X had X not been in B.
- Example (left-truncation): If the event time is length of time from diagnosis of cancer until death. Only individuals who live long enough after diagnosis to get in study are observed.
- $$\ell = \prod_{i=1}^{n} \frac{F(R_i) - F(L_i)}{F(r_i) - F(l_i)}$$

  where $(l_i, r_i]$ is the truncating interval.
- Frydman (1994, JRSS-B, 71-74) showed Turnbull (1976, JRSSB 290-295) intervals need to be modified for truncation.

Interval Censoring with Truncation

- The event time, X, is truncated in B if the researcher would not have been aware of the existence of X had X not been in B.
- Example (left-truncation): If the event time is length of time from diagnosis of cancer until death. Only individuals who live long enough after diagnosis to get in study are observed.
- $$\ell = \prod_{i=1}^{n} \frac{F(R_i) - F(L_i)}{F(r_i) - F(l_i)}$$

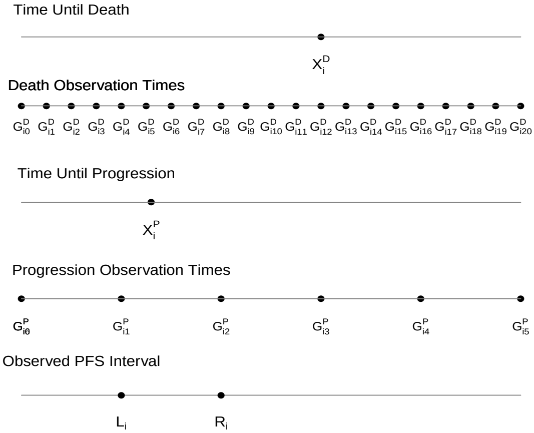
  where $(l_i, r_i]$ is the truncating interval.
- Frydman (1994, JRSS-B, 71-74) showed Turnbull (1976, JRSSB 290-295) intervals need to be modified for truncation.
- See Hudgens (2005, JRSSB 573-587) for existence conditions for NPMLE in presence of left truncation.

Informative Censoring

- Observation times depend on event time
- Strategies
  - Estimate frailty (random effects):
    - from frequency of observation times (Farrington and Gay, 1999, Stat in Med, 1235-48)
    - by modeling (Zhang, Sun, Sun, Finkelstein, 2007, Stat in Med, 2533-46)
  - Nonparametric (when regular observation times, and can observe observation process after event)
    - Finkelstein, Goggins, and Schoenfeld (2002, Biometrics, 298-304).

Why Progression-Free Survival Is Different

- Usually different observation times for death than for progression.
- We do not have one set of observation times that are independent of event time.
- Which observation process we observe depends on event.



Time Until Death

X_i^D

Death Observation Times

G_{i0}^D G_{i1}^D G_{i2}^D G_{i3}^D G_{i4}^D G_{i5}^D G_{i6}^D G_{i7}^D G_{i8}^D G_{i9}^D G_{i10}^D G_{i11}^D G_{i12}^D G_{i13}^D G_{i14}^D G_{i15}^D G_{i16}^D G_{i17}^D G_{i18}^D G_{i19}^D G_{i20}^D

Time Until Progression

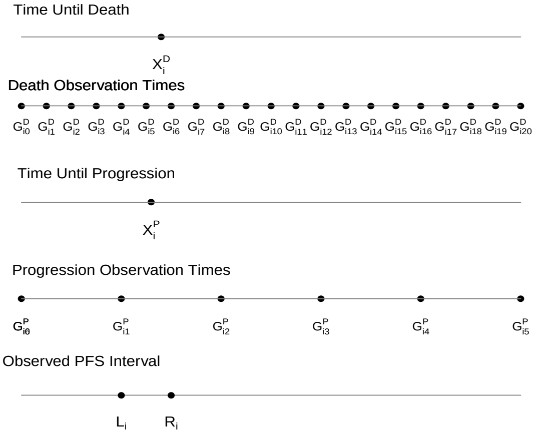
X_i^P

Progression Observation Times

G_{i0}^P G_{i1}^P G_{i2}^P G_{i3}^P G_{i4}^P G_{i5}^P

Observed PFS Interval

L_i R_i



Time Until Death

Death Observation Times

Time Until Progression

Progression Observation Times

Observed PFS Interval

X_i^D
G_{i0}^D G_{i1}^D G_{i2}^D G_{i3}^D G_{i4}^D G_{i5}^D G_{i6}^D G_{i7}^D G_{i8}^D G_{i9}^D G_{i10}^D G_{i11}^D G_{i12}^D G_{i13}^D G_{i14}^D G_{i15}^D G_{i16}^D G_{i17}^D G_{i18}^D G_{i19}^D G_{i20}^D
X_i^P
G_{i0}^P G_{i1}^P G_{i2}^P G_{i3}^P G_{i4}^P G_{i5}^P
L_i
R_i

Is this Dual Inspection Process a Big Problem?

- More work needs to be done to explore problems applying usual interval NPMLE to this problem

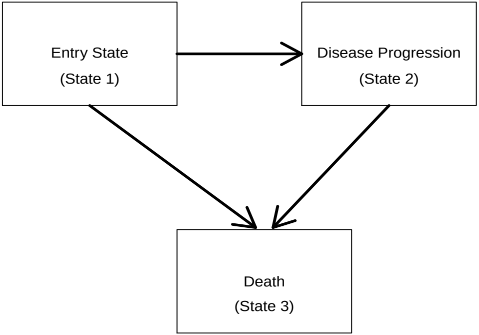
Is this Dual Inspection Process a Big Problem?

- More work needs to be done to explore problems applying usual interval NPMLE to this problem
- simple solution when scheduled progression observation times: if observe a death, use progression observation times to define interval. Lose information.

Is this Dual Inspection Process a Big Problem?

- More work needs to be done to explore problems applying usual interval NPMLE to this problem
- simple solution when scheduled progression observation times: if observe a death, use progression observation times to define interval. Lose information.
- better solution: illness-death model

Illness-Death Model



Entry State
(State 1)

Disease Progression
(State 2)

Death
(State 3)

Illness-Death Model

- Illness-Death Markov model (knowledge of current state [including time since start of study] supplies all prediction information)

Illness-Death Model

- Illness-Death Markov model (knowledge of current state [including time since start of study] supplies all prediction information)
- S is time leave entry state from start of study
- $F_{12}(s) = Pr[S \leq s$ and Progress]
- $F_{13}(s) = Pr[S \leq s$ and Die before Progress]
- $F(s) = F_{12}(s) + F_{13}(s)$ (distribution for progression-free survival)

Illness-Death Model

- Illness-Death Markov model (knowledge of current state [including time since start of study] supplies all prediction information)
- S is time leave entry state from start of study
- $F_{12}(s) = Pr[S \leq s$ and Progress$]$
- $F_{13}(s) = Pr[S \leq s$ and Die before Progress$]$
- $F(s) = F_{12}(s) + F_{13}(s)$ (distribution for progression-free survival)
- Frydman and Szarek (Biometrics, 2009, 143-151)
  - Assumes observation scheme non-informative
  - Allows interval-censoring of progression, right-censoring of death, properly handles death before observed progression
  - Uses self-consistent (i.e., E-M) algorithm
  - Notation very difficult
  - No readily available software

End of Part 1: Questions?

Types of 2-sample Tests

- Common tests, No censoring
  - t-test (difference in means)
  - Wilcoxon-Mann-Whitney test (rank test)

Types of 2-sample Tests

- Common tests, No censoring
  - t-test (difference in means)
  - Wilcoxon-Mann-Whitney test (rank test)
- Common tests, censoring
  - logrank test (rank test)
  - Wilcoxon-type test (rank test)

Types of 2-sample Tests

- Common tests, No censoring
  - t-test (difference in means)
  - Wilcoxon-Mann-Whitney test (rank test)
- Common tests, censoring
  - logrank test (rank test)
  - Wilcoxon-type test (rank test)
- Difference in means tests with censoring
  - Do not usually use difference in means tests because means effected by large tails, and with right censoring cannot estimate observations in tail well
  - Right censored: weighted Kaplan-Meier tests (Pepe and Fleming, 1989, Biometrics, 497-507)
  - Interval censored: integrated weighed difference tests (Petroni and Wolf, 1994, Biometrics, 77-87, Fay and Shih, 1998, JASA, 387-396)

Rank Testing with Censoring: Overview

1. Choose Likelihood/Process
   - Marginal Likelihood of Ranks (integrate over all ranks possible given censoring)
   - Grouped Continuous Model (estimate baseline distribution)
   - Counting Process
   - Partial Likelihood (only useful for right censoring, logrank)
2. Choose model or scores
   - Proportional odds/ Wilcoxon-type
   - Proportional hazards (continuous)/ Logrank-type
   - Approximate Proportional hazards (discrete)/ Logrank-type
3. Choose inference method
   - Permutation test on efficient scores (exact or asymptotic)
   - Score Test
   - Imputation

Cover first

Permutation Tests

- Data:
  - $\mathbf{x}$ responses
  - $\mathbf{z}$ covariates (treatment indicators)

Permutation Tests

- Data:
  - $\mathbf{x}$ responses
  - $\mathbf{z}$ covariates (treatment indicators)
- Choose test statistic, $T(\mathbf{x}, \mathbf{z})$, larger indicates farther from the null hypothesis
  - e.g.: $T(\mathbf{x}, \mathbf{z}) = \hat{P}r[X(1) > X(0)] + \frac{1}{2}\hat{P}r[X(1) = X(0)]$ where $X(a) = X$ when $z = a$

Permutation Tests

- Data:
  - $\mathbf{x}$ responses
  - $\mathbf{z}$ covariates (treatment indicators)
- Choose test statistic, $T(\mathbf{x},\mathbf{z})$, larger indicates farther from the null hypothesis
  - e.g.: $T(\mathbf{x},\mathbf{z}) = \hat{P}r[X(1) > X(0)] + \frac{1}{2}\hat{P}r[X(1) = X(0)]$ where $X(a) = X$ when $z = a$
- Permute
  - $T_0 = T(\mathbf{x},\mathbf{z})$
  - $T_i = T(\mathbf{x},\pi_i(\mathbf{z})), i = 1,\ldots,n!$
- p-value $= \frac{\sum_{i=1}^{n!} I\{T_i \geq T_0\}}{n!}$

Permutation Tests

- Data:
  - $\mathbf{x}$ responses
  - $\mathbf{z}$ covariates (treatment indicators)
- Choose test statistic, $T(\mathbf{x}, \mathbf{z})$, larger indicates farther from the null hypothesis
  - e.g.: $T(\mathbf{x}, \mathbf{z}) = \hat{P}r[X(1) > X(0)] + \frac{1}{2}\hat{P}r[X(1) = X(0)]$ where $X(a) = X$ when $z = a$
- Permute
  - $T_0 = T(\mathbf{x}, \mathbf{z})$
  - $T_i = T(\mathbf{x}, \pi_i(\mathbf{z})), i = 1, \ldots, n!$
- p-value $= \frac{\sum_{i=1}^{n!} I\{T_i \geq T_0\}}{n!}$
- Assumption: strong null (e.g.: treatment labels do not matter)

Permutation Tests

- Permute
  - $T_0 = T(\mathbf{x}, \mathbf{z})$
  - $T_i = T(\mathbf{x}, \pi(\mathbf{z})), i = 1, \ldots, n!$
- p-value $= \frac{\sum_{i=1}^{n!} I\{T_i \geq T_0\}}{n!}$

Permutation Tests

- Permute
    - $T_0 = T(\mathbf{x}, \mathbf{z})$
    - $T_i = T(\mathbf{x}, \pi(\mathbf{z})), i = 1, \ldots, n!$
- p-value $= \frac{\sum_{i=1}^{n!} I\{T_i \geq T_0\}}{n!}$
- p-values invariant to monotonic transformations of T
    - e.g.: $T^*(\mathbf{x}, \mathbf{z}) = b_0 + b_1 T(\mathbf{x}, \mathbf{z})$
      where b_0, b_1 do not change with permutations
    - Then p-values same for T and T^* since,
        - $I\{T_i \geq T_0\} = I\{T_i^* \geq T_0^*\}$ for all i.
    - Call T and T^* equivalent test statistics

- Following equivalent test statistics (all give the Wilcoxon-Mann-Whitney test):

- Following equivalent test statistics (all give the Wilcoxon-Mann-Whitney test):
  - $T =$ Difference in mean ranks

- Following equivalent test statistics (all give the Wilcoxon-Mann-Whitney test):
  - $T =$ Difference in mean ranks
  - $T =$ Sum of ranks from one group

- Following equivalent test statistics (all give the Wilcoxon-Mann-Whitney test):
    - $T =$ Difference in mean ranks
    - $T =$ Sum of ranks from one group
    - $T =$ sum of scores from one group, where ith score represents $\hat{P}r[X_i > X] + \frac{1}{2}\hat{P}r[X_i = X]$
      where X random response regardless of group

- Following equivalent test statistics (all give the Wilcoxon-Mann-Whitney test):
  - $T =$ Difference in mean ranks
  - $T =$ Sum of ranks from one group
  - $T =$ sum of scores from one group, where ith score represents $\hat{P}r[X_i > X] + \frac{1}{2}\hat{P}r[X_i = X]$
    where X random response regardless of group
  - $T =$ sum of scores from one group, where ith score represents $\hat{P}r[X_i \geq X] + \hat{P}r[X_i > X] - 1$
    where X random response regardless of group
    scores sum to 0 in this case

Different scores just linear functions of ranks:

Rank	$\hat{P}r[X_i \geq X]$	$\hat{P}r[X_i > X]$	$\hat{P}r[X_i \geq X] + \hat{P}r[X_i > X] - 1$
1	1/n	0/n	$\frac{1}{n}$ -1
2	2/n	1/n	$\frac{3}{n}$ - 1
⋮	⋮	⋮	
j	j/n	(j-1)/n	$\frac{2j-1}{n}$ - 1
⋮	⋮	⋮	
n	n/n	(n-1)/n	$\frac{2n-1}{n}$ -1

$$\sum c_i = \sum \frac{2i - 1 - n}{n} = 0$$

Generalizing Wilcoxon-Mann-Whitney for Censoring

- Peto and Peto (1972, JRSS-A, 185-207).
  - paper included possibility of interval censoring
  - can also be derived as permutation test on score statistic (more on that later)
- Permutation test on sum of scores, c_i, from one group
  -

$$\begin{aligned} c_i &= \hat{P}r[X_i \geq X] + \hat{P}r[X_i > X] - 1 \\ &= \hat{S}(L_i) + \hat{S}(R_i) - 1 \end{aligned}$$

Generalizing Wilcoxon-Mann-Whitney for Censoring

- Peto and Peto (1972, JRSS-A, 185-207).
  - paper included possibility of interval censoring
  - can also be derived as permutation test on score statistic (more on that later)
- Permutation test on sum of scores, c_i, from one group
  -

$$\begin{aligned} c_i &= \hat{Pr}[X_i \geq X] + \hat{Pr}[X_i > X] - 1 \\ &= \hat{S}(L_i) + \hat{S}(R_i) - 1 \end{aligned}$$

- Peto and Peto (1972) introduced another rank test, the logrank test (more on logrank test later)

Right Censoring

- Turnbull intervals are point masses at observed event times, plus interval from last censored observation to ∞ if last censored observation is larger than largest observed event time.
- $c_i = \hat{S}(L_i) + \hat{S}(R_i) - 1$
  - Observed event time at R_i: let $L_i = R_i-$
    then $c_i = \hat{S}(R_i-) + \hat{S}(R_i) - 1$
  - Right censored at L_i: let $R_i = \infty$
    then $c_i = \hat{S}(L_i) - 1$

Inference from a Permutation Test

- Assumptions:
    - Under null hypothesis treatment (z_i values) are independent of failure time and assessment times.
        - Therefore, for permutation test: No information on treatment (z_i values) used in creating scores.
    - Non-informative censoring

Inference from a Permutation Test

- Assumptions:
  - Under null hypothesis treatment (z_i values) are independent of failure time and assessment times.
    - Therefore, for permutation test: No information on treatment (z_i values) used in creating scores.
  - Non-informative censoring
- Exact tests
  - complete enumeration
  - network algorithm (StatXact)
  - many other algorithms
  - Monte Carlo approximation to exact

Inference from a Permutation Test

- Assumptions:
  - Under null hypothesis treatment (z_i values) are independent of failure time and assessment times.
    - Therefore, for permutation test: No information on treatment (z_i values) used in creating scores.
  - Non-informative censoring
- Exact tests
  - complete enumeration
  - network algorithm (StatXact)
  - many other algorithms
  - Monte Carlo approximation to exact
- Asymptotic tests
  - see "Permutational central limit theorems" (Ency of Stat., Sen, 1985).

Rank Testing with Censoring: Overview

Colors: covered, Next

1. Choose Likelihood/Process
   - Marginal Likelihood of Ranks (integrate over all ranks possible given censoring)
   - Grouped Continuous Model (estimate baseline distribution)
   - Counting Process
   - Partial Likelihood (only useful for right censoring, logrank)
2. Choose model or scores
   - Proportional odds/ Wilcoxon-type
   - Proportional hazards (continuous)/ Logrank-type
   - Approximate Proportional hazards (discrete)/ Logrank-type
3. Choose inference method
   - Permutation test on efficient scores (exact or asymptotic)
   - Score Test
   - Imputation

Choosing a Model

Parametric Model: Accelerated Failure Time Model

$$\log(X_i) = \alpha + z_i\beta + \sigma\epsilon$$

where

- z_i is a vector of covariates
- β is a vector of parameters
- α and σ are location and scale parameters
- ϵ is the error, where $\epsilon \sim F$ where F is a known distribution.

Choosing a Model

Semi-Parametric Model

$$g(X_i) = z_i\beta + \epsilon$$

where

- $g(\cdot)$ is an UNKNOWN monotonic transformation of the failure time
- z_i is a vector of covariates
- β is a vector of parameters
- ϵ is the error, where $\epsilon \sim F$ where F is a known distribution with convenient mean and variance (not necessarily mean=0 and variance=1).

Choosing a Model

Choose F

- F is logistic
  - Proportional Odds Model
  - Let $S(t;\beta,H) = 1 - F(t;\beta,H)$ and $S_0(t) = S(t;0,H)$, where H is a vector of nuisance parameters, and

$$\begin{aligned}\frac{S(t;\beta,H)}{1-S(t;\beta,H)} &= \frac{S_0(t)}{1-S_0(t)}\exp(-z_i'\beta)\\ \Rightarrow S(t;\beta,H) &= \left[1+\left(\frac{1-S_0(t)}{S_0(t)}\right)\exp(z_i\beta)\right]^{-1}\end{aligned}$$

- F is extreme minimum value
  - Proportional Hazards Model
  - $S(t;\beta,H) = S_0(t)^{\exp(z_i\beta)}$
  - $\lambda(t;\beta,H) = \lambda_0(t)\exp(z_i\beta)$

Two Logrank Models

- Finkelstein (1986, Biometrics, 845-854)
  - grouped continuous model on continuous proportional hazards model.
  - $\lambda(t;\beta,H) = \lambda_0(t)\exp(z_i\beta)$
- Sun (1996, Stat in Med, 1387-1395)
  - follow discrete approximation to proportional hazards used in Cox (1972, JRSSB 187-220).
  - $$\frac{\lambda(t;\beta,H)}{1-\lambda(t;\beta,H)} = \frac{\lambda_0(t)}{1-\lambda_0(t)}\exp(z_i\beta)$$
  - $$S(t;\beta,H) = \prod_{\ell=1}^{j}\left[1+\left(\frac{S_0(t_{\ell-1})-S_0(t_\ell)}{S_0(t_\ell)}\right)\exp(z_i\beta)\right]^{-1}$$

Rank Testing with Censoring: Overview

Colors: covered, Next

1. Choose Likelihood/Process
   - Marginal Likelihood of Ranks (integrate over all ranks possible given censoring)
   - Grouped Continuous Model (estimate baseline distribution)
   - Counting Process
   - Partial Likelihood (only useful for right censoring, logrank)
2. Choose model or scores
   - Proportional odds/ Wilcoxon-type
   - Proportional hazards (continuous)/ Logrank-type
   - Approximate Proportional hazards (discrete)/ Logrank-type
3. Choose inference method
   - Permutation test on efficient scores (exact or asymptotic)
   - Score Test
   - Imputation

Choosing Likelihood/Process: Marginal Likelihood of Ranks

Semi-Parametric Model

$$g(X_i) = z_i\beta + \epsilon$$

where $\epsilon \sim F$.

- Since $g(\cdot)$ is unknown and monotonic, ranks have all information. So use marginal likelihood of ranks.
- $$\sum_{r \in \mathbf{R}} \int_{A(r)} \dots \int \prod_{i=1}^{n} f(u_i - z_i\beta) du_i$$

  where $\mathbf{R}$ is set of ranks, r, consistent with censoring
  $A(r)$ is region corresponding to rank vector r

Choosing Likelihood/Process: Marginal Likelihood of Ranks

- Score test: Self and Grossman (1986, Biometrics, 521-30)
  - Theoretically nice and clean
  - Difficult to calculate scores
- marginal likelihood of ranks (MLR) can be used for semi-parametric regression
  - Return to this in regression section.

Choosing Likelihood/Process: Grouped Continuous Model

Semi-Parametric Model

$$g(X_i) = z_i\beta + \epsilon$$

where $\epsilon \sim F$.
Likelihood:

$$\ell = \prod_{i=1}^{n} \{F(g(R_i) - z_i\beta) - F(g(L_i) - z_i\beta)\}$$

$$\begin{aligned} \text{where} \quad z_i &= \text{vector of treatment indicators} \\ \beta &= \text{treatment parameters} \\ g(\cdot) &= \text{infinite dimensional nuisance parameter} \end{aligned}$$

Only need to estimate $g(X)$ at inspection times.

Choosing Likelihood/Process: Grouped Continuous Model

Recall the Likelihood:

$$\ell \;=\; \prod_{i=1}^{n} \{F(g(R_i) - z_i\beta) - F(g(L_i) - z_i\beta)\}$$

Reparametrize:

$$\ell \;=\; \prod_{i=1}^{n} \left\{F\left(F^{-1}\left[H(R_i)\right] - z_i\beta\right) - F\left(F^{-1}\left[H(L_i)\right] - z_i\beta\right)\right\}$$

here $H = 1 - S_0$ is a baseline distribution function.

Choosing Likelihood/Process: Grouped Continuous Model

Efficient Score Vector (ignore Math if you want):

$$
\begin{aligned}
U &= \left[\frac{\partial \log \ell}{\partial \beta}\right]_{\beta=0} \\
&= \frac{\partial}{\partial \beta}\left[\sum_{i=1}^{n} \log\left\{F\left(F^{-1}\left[\hat{H}(R_i)\right] - z_i\beta\right) - \right.\right. \\
&\qquad \left.\left. F\left(F^{-1}\left[\hat{H}(L_i)\right] - z_i\beta\right)\right\}\right]_{\beta=0} \\
&= \sum_{i=1}^{n} \frac{\frac{\partial}{\partial\beta}\left\{F\left(F^{-1}\left[\hat{H}(R_i)\right] - z_i\beta\right) - F\left(F^{-1}\left[\hat{H}(L_i)\right] - z_i\beta\right)\right\}_{\beta=0}}{\hat{H}(R_i) - \hat{H}(L_i)} \\
&= \sum_{i=1}^{n} z_i \frac{-\left\{f\left(F^{-1}\left[\hat{H}(R_i)\right]\right) - f\left(F^{-1}\left[\hat{H}(L_i)\right]\right)\right\}}{\hat{H}(R_i) - \hat{H}(L_i)} \\
&= \sum_{i=1}^{n} z_i c_i
\end{aligned}
$$

Choosing Likelihood/Process: Grouped Continuous Model

Efficient Score Vector (bottom line):

$$U = \sum_{i=1}^{n} z_i c_i$$

where c_i is a function of F and NPMLE of survival function, $\hat{S} = 1 - \hat{H}$

Scores:

- Proportional odds (Wilcoxon-type)
  - $c_i = \hat{S}(L_i) + \hat{S}(R_i) - 1$
- Proportional Hazards (logrank, Finkelstein, 1986)
  - $c_i = \frac{\hat{S}(L_i) \log \hat{S}(L_i) - \hat{S}(R_i) \log \hat{S}(R_i)}{\hat{S}(L_i) - \hat{S}(R_i)}$
- Proportional Hazards (discrete version) (logrank, Sun, 1996)
  - $c_i = \frac{\hat{S}(L_i) \log \tilde{S}(L_i) - \hat{S}(R_i) \log \tilde{S}(R_i)}{\hat{S}(L_i) - \hat{S}(R_i)}$
  - where $\tilde{S}$ is like a Nelson-Aalen estimator,

$$\begin{aligned} \tilde{S}(t_j) &= \exp\left(- \sum_{k=1}^{j} \hat{\lambda}_k \right) \\ &\textit{where} \\ \hat{\lambda}_k &= \frac{\hat{S}(t_{k-1}) - \hat{S}(t_k)}{\hat{S}(t_{k-1})} \end{aligned}$$

Alternate Expression for Weighted Logrank Tests

- Consider ℓth element of score vector, U representing treatment ℓ:

$$U_\ell = \sum_{j=1}^{m} w_j \left(D_{j\ell} - \frac{N_{j\ell} D_j}{N_j} \right)$$

- $D_j =$ expected number of total deaths in $(t_{j-1}, t_j]$
- $D_{j\ell} =$ expected number of deaths in $(t_{j-1}, t_j]$, treatment group ℓ
- N_j and $N_{j\ell}$ expected number at risk in $(t_{j-1}, t_j]$
- Sun's (1996) test: $w_j = 1$
- Finkelstein's (1986) test:

$$w_j = \frac{\hat{S}(t_{j-1}) \left\{ \log \hat{S}(t_{j-1}) - \log \hat{S}(t_j) \right\}}{\hat{S}(t_{j-1}) - \hat{S}(t_j)} \approx 1$$

- Fay (1999, Stat in Med, 273-285) shows equivalence between score tests in weighted logrank form (above) and permutation form ($\sum z_i c_i$).

Alternate Expression for Weighted Logrank Tests

- Consider ℓth element of score vector, U representing treatment ℓ:

$$U_\ell = \sum_{j=1}^{m} w_j \left(D_{j\ell} - \frac{N_{j\ell} D_j}{N_j} \right)$$

- Sun's (1996) test: $w_j = 1$
- Finkelstein's (1986) test:

$$w_j = \frac{\hat{S}(t_{j-1}) \left\{ \log \hat{S}(t_{j-1}) - \log \hat{S}(t_j) \right\}}{\hat{S}(t_{j-1}) - \hat{S}(t_j)} \approx 1$$

- Proportional Odds (Wilcoxon-type test):

$$w_j = \hat{S}(t_{j-1})$$

Rank Testing with Censoring: Overview

Colors: covered, Next

1. Choose Likelihood/Process
   - Marginal Likelihood of Ranks (integrate over all ranks possible given censoring)
   - Grouped Continuous Model (estimate baseline distribution)
   - Counting Process
   - Partial Likelihood (only useful for right censoring, logrank)
2. Choose model or scores
   - Proportional odds/ Wilcoxon-type
   - Proportional hazards (continuous)/ Logrank-type
   - Approximate Proportional hazards (discrete)/ Logrank-type
3. Choose inference method
   - Permutation test on efficient scores (exact or asymptotic)
   - Score Test
   - Imputation

Why not Use MidPoint Imputation then usual Right Censoring Methods?

Before considering formal interval censoring inference methods, show there is a problem with using midpoint imputation on intervals then using usual right censoring methods.

Inferences when Treatment Related to Inspection Process Only

Examples:

- Treatment A scheduled every 2 weeks, treatment B scheduled every 4 weeks.
- Treatment A causes side effects which lead to more frequent inspections, but does not change time to progression or death

Simple Methods When Inspection Times Differ Between Treatments

- Midpoint imputation, then treat data like right censored data
  - Law and Brookmeyer (1992, Stat in Med, 1569-1578) showed type I error for nominal 0.05 can be as large as 0.19
  - See also recent work by Sun and Chen (unpublished manuscript JSM, 2009).

Interval-Censoring Inference Methods

- $H_0 : \beta = 0$
- Recall efficient score vector,

$$U = \sum_{i=1}^{n} z_i c_i$$

where c_i are function of NPMLE of distribution

- Inference options
  - Permutation of treatment vector z_i
    - Require no treatment information used in calculating scores, c_i.
    - Assumption: Non-informative censoring... inspection times independent of event time
    - Assumption: inspection times independent of treatment

Interval-Censoring Inference Methods

- $H_0 : \beta = 0$
- Recall efficient score vector,

$$U = \sum_{i=1}^{n} z_i c_i$$

  where c_i are function of NPMLE of distribution
- Inference options
  - Permutation of treatment vector z_i
    - Require no treatment information used in calculating scores, c_i.
    - Assumption: Non-informative censoring... inspection times independent of event time
    - Assumption: inspection times independent of treatment
  - Score Test
    - Require number of nuisance parameters (dimension of NPMLE) to be fixed as $n \to \infty$.
    - Assumption: Non-informative censoring... inspection times independent of event time
    - Likelihood-based, so inspection times may depend on treatment

Exponential Example

- $X \sim Exponential(1)$
- Inspection Schedule for Treatment A: $0, 1, 2$
- Inspection Schedule for Treatment B: $0, 2$
- $n = 1000$ in each group
- Treatment A:
  - $(0, 1]$: expect n=632, expected logrank score 0.368
  - $(1, 2]$: expect n=233, expected logrank score -0.265
  - $(2, \infty)$: expect n=135, expected logrank score -1.265
- Treatment B:
  - $(0, 2]$: expect n=632+233, expected logrank score

$$\frac{632 * 0.368 + 233 * (-.265)}{632 + 233} = .197$$

  - $(2, \infty)$: expect n=135, expected logrank score -1.265

Exponential Example

Problem with Midpoint Imputation:

- $X \sim \mathit{Exponential}(1)$
- Inspection Schedule for Treatment A: $0, 1, 2$
- Inspection Schedule for Treatment B: $0, 2$
- $n = 1000$ in each group
- Treatment A:
  - $(0, 1] \Rightarrow .5$: expect n=632, expected logrank score 0.648
  - $(1, 2] \Rightarrow 1.5$: expect n=233, expected logrank score -0.411
  - $(2, \infty)$: expect n=135, expected logrank score -1.411
- Treatment B:
  - $(0, 2] \Rightarrow 1$: expect n=632+233, expected logrank score

$$0.0517 \neq \frac{632 * 0.648 + 233 * (-.411)}{632 + 233} = 0.389$$

  - $(2, \infty)$: expect n=135, expected logrank score -1.411

Rewriting Scores as Weighted Sums

Notation:

- $t_1, t_2, \ldots, t_m$, set of possible observation times where NPMLE (all groups combined) may change.
- Let $t_0 \equiv 0$ and $t_{m+1} \equiv \infty$.
- $p(t_j) = Pr[t_{j-1} < X \leq t_j]$
- NPMLE of probability mass function: $\hat{\mathbf{p}} = [\hat{p}(t_1), \hat{p}(t_2), \ldots, \hat{p}(t_{m+1})]$
- $\hat{p}_i(t_j) = \hat{Pr}\left[t_{j-1} < X_i \leq t_j \mid X_i \in (L_i, R_i]\right]$

Rewrite scores:

$$\begin{aligned} c_i &= c(L_i, R_i, \hat{\mathbf{p}}) \\ &= \sum_{j=1}^{m+1} \hat{p}_i(t_j) c(t_{j-1}, t_j, \hat{\mathbf{p}}) \end{aligned}$$

Permutation Method when Inspection Process Different Between Treatment Groups

- Although variances of scores, c_i, may be different by changing the inspection process, expected value of score c_i is zero regardless of inspection process.

Permutation Method when Inspection Process Different Between Treatment Groups

- Although variances of scores, c_i, may be different by changing the inspection process, expected value of score c_i is zero regardless of inspection process.
- So it is hard to think of two inspection processes (one for each treatment group) that would give a bad type I error.

Permutation Method when Inspection Process Different Between Treatment Groups

- Although variances of scores, c_i, may be different by changing the inspection process, expected value of score c_i is zero regardless of inspection process.
- So it is hard to think of two inspection processes (one for each treatment group) that would give a bad type I error.
- That is why generally, permutation test does not perform badly even when the inspection processes are different between groups.

Score Method when Inspection Process Different Between Treatment Groups

- Theoretically, score method does not require that the two treatment groups have the same inspection process.

Score Method when Inspection Process Different Between Treatment Groups

- Theoretically, score method does not require that the two treatment groups have the same inspection process.
- The problem with score method is that the number of nuisance parameters may grow with n. Further, nuisance parameters may be on boundary of parameter space.

Score Method when Inspection Process Different Between Treatment Groups

- Theoretically, score method does not require that the two treatment groups have the same inspection process.
- The problem with score method is that the number of nuisance parameters may grow with n. Further, nuisance parameters may be on boundary of parameter space.
- Fay (1996, Biometrics, 811-22) redefines nuisance parameters, any part of distribution with estimated mass of 0 is set to 0. Then do usual score test. This is how *interval* R package does score test.

Score Method when Inspection Process Different Between Treatment Groups

- Theoretically, score method does not require that the two treatment groups have the same inspection process.
- The problem with score method is that the number of nuisance parameters may grow with n. Further, nuisance parameters may be on boundary of parameter space.
- Fay (1996, Biometrics, 811-22) redefines nuisance parameters, any part of distribution with estimated mass of 0 is set to 0. Then do usual score test. This is how *interval* R package does score test.
- Because of ad hoc nature, prefer permutation methods for inferences when very many nuisance parameters (i.e., almost continuous inspection processes).

Rank Testing with Censoring: Overview

Colors: covered, Next

1. Choose Likelihood/Process
   - Marginal Likelihood of Ranks (integrate over all ranks possible given censoring)
   - Grouped Continuous Model (estimate baseline distribution)
   - Counting Process
   - Partial Likelihood (only useful for right censoring, logrank)
2. Choose model or scores
   - Proportional odds/ Wilcoxon-type
   - Proportional hazards (continuous)/ Logrank-type
   - Approximate Proportional hazards (discrete)/ Logrank-type
3. Choose inference method
   - Permutation test on efficient scores (exact or asymptotic)
   - Score Test
   - Imputation

Testing: Imputation

- Imputation
  - Calculate Non-parametric MLE of survival function
  - Sample from NPMLE, impute into usual right-censored weighted logrank equations m times, average m score vectors, U_j, but correct average of m variances by within cluster resampling method:

$$V = \frac{\sum_{j=1}^{m} V_j}{m} - \frac{\sum_{j=1}^{m} (U_j - \bar{U})(U_j - \bar{U})^T}{m-1}$$

  - Motivation, letting D be data

$$Var\left[E(U|D)\right] = Var(U) - E\left[Var(U|D)\right]$$

  - Huang, Lee, and Yu (2008, Statistics in Medicine, 3217-3226).
  - Variance estimation similar to within cluster resampling (Hoffman, Sen, Weinberg, Biometrika, 2001, 1121-1134) also called Multiple Outputation (Follmann, Proschan, and Leifer, Biometrics, 2003, 420-429).

Overview of Type I Error Problems

- Data: Inspection process independent of treatment and response.
  - No problems for any methods.

Overview of Type I Error Problems

- Data: Inspection process independent of treatment and response.
  - No problems for any methods.
- Data: Inspection process related to treatment not response
  - Type I error problem: Midpoint imputation
  - No type I error problem: score test
  - Probably not much type I error problem: permutation test

Overview of Type I Error Problems

- Data: Inspection process independent of treatment and response.
  - No problems for any methods.
- Data: Inspection process related to treatment not response
  - Type I error problem: Midpoint imputation
  - No type I error problem: score test
  - Probably not much type I error problem: permutation test
- Data: Informative censoring, but same for all groups
  - Permutation method no type I error problem
  - Score test and midpoint imputation ?

Overview of Type I Error Problems

- Data: Inspection process independent of treatment and response.
  - No problems for any methods.
- Data: Inspection process related to treatment not response
  - Type I error problem: Midpoint imputation
  - No type I error problem: score test
  - Probably not much type I error problem: permutation test
- Data: Informative censoring, but same for all groups
  - Permutation method no type I error problem
  - Score test and midpoint imputation ?
- Data: Informative censoring and differential between treatment groups.
  - Problem with all methods.

Overview of Type I Error Problems

- Data: Inspection process independent of treatment and response.
  - No problems for any methods.
- Data: Inspection process related to treatment not response
  - Type I error problem: Midpoint imputation
  - No type I error problem: score test
  - Probably not much type I error problem: permutation test
- Data: Informative censoring, but same for all groups
  - Permutation method no type I error problem
  - Score test and midpoint imputation ?
- Data: Informative censoring and differential between treatment groups.
  - Problem with all methods.
- Data: Continuous inspection process but same for all treatment groups
  - score test (theory not known, perhaps type I error problem?)
  - permutation test (OK)

Testing: Stratification

Three methods all maintain type I error when strata effects present:

1. Weighted logrank test ignoring strata
   - when small strata effects, this is more powerful than method 2

Testing: Stratification

Three methods all maintain type I error when strata effects present:

1. Weighted logrank test ignoring strata
   - when small strata effects, this is more powerful than method 2
2. Separate ranking (i.e., separate baseline hazard) within each strata, then either permute accounting for strata or combine separate weighted logrank tests (e.g. stratified logrank test)
   - when large strata effects this is more powerful than method 1

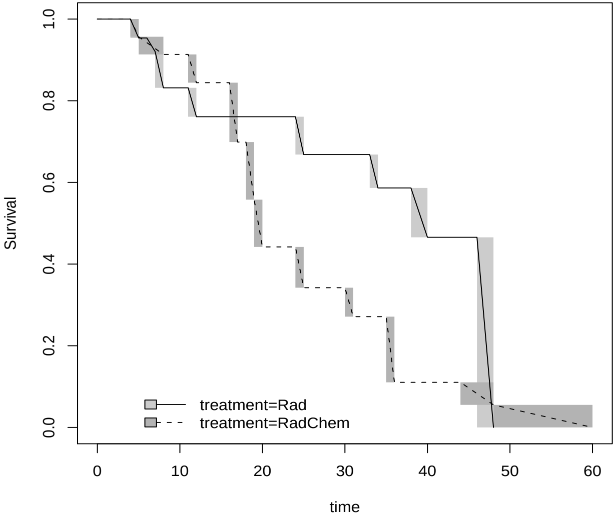
Testing: Stratification

Three methods all maintain type I error when strata effects present:

1. Weighted logrank test ignoring strata
   - when small strata effects, this is more powerful than method 2
2. Separate ranking (i.e., separate baseline hazard) within each strata, then either permute accounting for strata or combine separate weighted logrank tests (e.g. stratified logrank test)
   - when large strata effects this is more powerful than method 1
3. Automatic adjustment
   - measure within versus between variance using ranks only
   - shrinkage estimator of distribution for each strata, if within variance large shrink a lot, if between variance large shrink little
   - rank subjects based on shrinkage estimator
   - automatically gives good power regardless of whether strong or weak strata effect
   - Shih and Fay (1999, Biometrics, 1156-1161)

Recall breast cosmesis data:

```
> library(interval)
> data(bcos)
> fit<-icfit(Surv(left,right,type="interval2")~treatment,
+     data=bcos)
> plot(fit)
> plot(fit)
```



Survival
1.0
0.8
0.6
0.4
0.2
0.0
time
0
10
20
30
40
50
60
treatment=Rad
treatment=RadChem

logrank: Sun's scores

```
> test<-ictest(Surv(left,right,type="interval2")~treatment,data=bcos)
> test


        Asymptotic Logrank two-sample test (permutation form), Sun's scores

data:  Surv(left, right, type = "interval2") by treatment
Z = -2.6684, p-value = 0.007622
alternative hypothesis: survival distributions not equal

                   n Score Statistic*
treatment=Rad     46        -9.141846
treatment=RadChem 48         9.141846
* like Obs-Exp, positive implies earlier failures than expected
```

logrank: Finkelstein's scores

```
> ictest(Surv(left,right,type="interval2")~treatment,data=bcos,
+     scores="logrank2")
```

```
        Asymptotic Logrank two-sample test (permutation form), Finkelstein's
        scores

data:  Surv(left, right, type = "interval2") by treatment
Z = -2.6839, p-value = 0.007277
alternative hypothesis: survival distributions not equal

                   n Score Statistic*
treatment=Rad     46        -9.944182
treatment=RadChem 48         9.944182
* like Obs-Exp, positive implies earlier failures than expected
```

Wilcoxon-type tests

```
> ictest(Surv(left,right,type="interval2")~treatment,data=bcos,
+     scores="wmw")
```

```
        Asymptotic Wilcoxon two-sample test (permutation form)

data:  Surv(left, right, type = "interval2") by treatment
Z = -2.1672, p-value = 0.03022
alternative hypothesis: survival distributions not equal

                   n Score Statistic*
treatment=Rad     46       -5.656724
treatment=RadChem 48        5.656724
* like Obs-Exp, positive implies earlier failures than expected
```

exact Monte Carlo

```
> ictest(Surv(left,right,type="interval2")~treatment,data=bcos,
+     scores="wmw", exact=TRUE)
```

```
        Exact Wilcoxon two-sample test (permutation form)

data:  Surv(left, right, type = "interval2") by treatment
p-value = 0.026
alternative hypothesis: survival distributions not equal

                   n Score Statistic*
treatment=Rad     46        -5.656724
treatment=RadChem 48         5.656724
* like Obs-Exp, positive implies earlier failures than expected
p-value estimated from 999 Monte Carlo replications
99 percent confidence interval on p-value:
 0.009926283 0.048044749
```

(took 0.08 seconds on my desktop).

exact Monte Carlo

```
> test<-ictest(Surv(left,right,type="interval2")~treatment,data=bcos,
+     scores="wmw", exact=TRUE, mcontrol=mControl(nmc=10^6-1))
```

```
        Exact Wilcoxon two-sample test (permutation form)

data:  Surv(left, right, type = "interval2") by treatment
p-value = 0.02967
alternative hypothesis: survival distributions not equal

                   n Score Statistic*
treatment=Rad     46       -5.656724
treatment=RadChem 48        5.656724
* like Obs-Exp, positive implies earlier failures than expected
p-value estimated from 999999 Monte Carlo replications
99 percent confidence interval on p-value:
 0.02905283 0.03030048
```

(took 32 seconds on my desktop).

exact network algorithm

```
test<-ictest(Surv(left,right,type="interval2")~treatment,data=bcos,
    scores="wmw", exact=TRUE, method="exact.network")
```

```
Error: cannot allocate vector of size 244.7 Mb
```

So, network algorithm is only feasible for quite small sample sizes (about 10 per group).

score method

```
> ictest(Surv(left,right,type="interval2")~treatment,data=bcos,
+     method="scoretest")
```

```
        Asymptotic Logrank two-sample test (score form), Sun's scores

data:  Surv(left, right, type = "interval2") by treatment
Chi Square = 7.6177, p-value = 0.00578
alternative hypothesis: survival distributions not equal

                   n Score Statistic*
treatment=Rad     46       -9.141846
treatment=RadChem 48        9.141846
* like Obs-Exp, positive implies earlier failures than expected
```

imputation: within subject resampling

```
> ictest(Surv(left,right,type="interval2")~treatment,data=bcos,
+     method="wsr.HLY")
```

```
        Asymptotic Logrank 2-sample test(WSR HLY), Sun's scores

data:  Surv(left, right, type = "interval2") by treatment
Chi Square = 7.1912, p-value = 0.007326
alternative hypothesis: survival distributions not equal

                   n Score Statistic*
treatment=Rad     46        -9.141846
treatment=RadChem 48         9.141846
* like Obs-Exp, positive implies earlier failures than expected
p-value estimated from  Monte Carlo replications
```

Regression: Semi-parametric approaches

- Marginal likelihood of ranks
    - can be solved by stochastic integration (Markov Chain Monte Carlo)

Regression: Semi-parametric approaches

- Marginal likelihood of ranks
    - can be solved by stochastic integration (Markov Chain Monte Carlo)
        - Proportional Hazards: Satten (1996, Biometrika, 355-370)

Regression: Semi-parametric approaches

- Marginal likelihood of ranks
  - can be solved by stochastic integration (Markov Chain Monte Carlo)
    - Proportional Hazards: Satten (1996, Biometrika, 355-370)
    - Proportional odds: Gu, Sun, Zuo (2005, Lifetime Data Analysis, 473-488)

Regression: Semi-parametric approaches

- Marginal likelihood of ranks
  - can be solved by stochastic integration (Markov Chain Monte Carlo)
    - Proportional Hazards: Satten (1996, Biometrika, 355-370)
    - Proportional odds: Gu, Sun, Zuo (2005, Lifetime Data Analysis, 473-488)
  - Monte Carlo E-M for Proportional Hazards (Goggins, Finkelstein, Schoenfeld, Zaslavsky, 1998, Biometrics, 1498-1507) S program at http://hedwig.mgh.harvard.edu/biostatistics/software

Regression: Semi-parametric approaches

- Marginal likelihood of ranks
  - can be solved by stochastic integration (Markov Chain Monte Carlo)
    - Proportional Hazards: Satten (1996, Biometrika, 355-370)
    - Proportional odds: Gu, Sun, Zuo (2005, Lifetime Data Analysis, 473-488)
  - Monte Carlo E-M for Proportional Hazards (Goggins, Finkelstein, Schoenfeld, Zaslavsky, 1998, Biometrics, 1498-1507) S program at http://hedwig.mgh.harvard.edu/biostatistics/software
  - E-M on Proportional Hazards (Goetghebeur, Ryan, 2000, Biometrics, 1139-1144)

Regression: Semi-parametric approaches

- Marginal likelihood of ranks
  - can be solved by stochastic integration (Markov Chain Monte Carlo)
    - Proportional Hazards: Satten (1996, Biometrika, 355-370)
    - Proportional odds: Gu, Sun, Zuo (2005, Lifetime Data Analysis, 473-488)
  - Monte Carlo E-M for Proportional Hazards (Goggins, Finkelstein, Schoenfeld, Zaslavsky, 1998, Biometrics, 1498-1507) S program at http://hedwig.mgh.harvard.edu/biostatistics/software
  - E-M on Proportional Hazards (Goetghebeur, Ryan, 2000, Biometrics, 1139-1144)
  - Iterative Convex Minorant algorithm for Proportional Hazards (Pan, 1999, J. Comp Graph Stat, 109-120) intcox R package (package not written by Pan, I could not get results to match coxph when data right censored).

Regression: Semi-parametric approaches (continued)

- Estimate non-parametric part with many parameters
  - Piecewise constant intensity model, then use GLM methods for Bernoulli (Farrington, 1996, Stat in Med, 283-292, Carstensen, 1996, Stat in Med, 2177-2189)
  - Icens function in Epi R package

Regression: Semi-parametric approaches (continued)

- Estimate non-parametric part with many parameters
  - Piecewise constant intensity model, then use GLM methods for Bernoulli (Farrington, 1996, Stat in Med, 283-292, Carstensen, 1996, Stat in Med, 2177-2189)
  - Icens function in Epi R package
- Likelihood Smoothing using Kernels (Betensky, Lindsey, Ryan and Wand, 2002, Stat in Med, 263-275)

Parametric Models

Accelerated Failure Time Models

$$\log(X_i) = \alpha + z_i\beta + \sigma\epsilon$$

where

- z_i is a vector of covariates
- β is a vector of parameters
- α and σ are location and scale parameters
- ϵ is the error, where $\epsilon \sim F$ where F is a known distribution.

Parametric Models: AFT

$$\log(X_i) = \alpha + z_i\beta + \sigma\epsilon_i$$

where ϵ_i is the error, where $\epsilon_i \sim F$ and F is known.
Creating Likelihood:

$$\epsilon_i = \frac{\log(X_i) - (\alpha + z_i\beta)}{\sigma}$$

so

$$Pr[X_i \leq x] = F\left[\frac{\log(X_i) - (\alpha + z_i\beta)}{\sigma}\right]$$

but $X_i \in (L_i, R_i]$ so

$$L = \prod_{i=1}^{n}\left\{F\left[\frac{\log(R_i) - (\alpha + z_i\beta)}{\sigma}\right] - F\left[\frac{\log(L_i) - (\alpha + z_i\beta)}{\sigma}\right]\right\}$$

Parametric Models: AFT

$X_i \in (L_i, R_i]$ so

$$L = \prod_{i=1}^{n} \left\{ F\left[\frac{\log(R_i) - (\alpha + z_i\beta)}{\sigma}\right] - F\left[\frac{\log(L_i) - (\alpha + z_i\beta)}{\sigma}\right] \right\}$$

Special cases:

- Right censoring, $R_i = \infty$ and since $F(\infty) = 1$,

$$L = \prod_{i=1}^{n} \left\{ 1 - F\left[\frac{\log(L_i) - (\alpha + z_i\beta)}{\sigma}\right] \right\}$$

- Left censoring, $L_i = 0$ and since $F(0) = 0$,

$$L = \prod_{i=1}^{n} \left\{ F\left[\frac{\log(R_i) - (\alpha + z_i\beta)}{\sigma}\right] \right\}$$

Inference from Parametric Model

- Usual asymptotic methods
  - Wald test
  - Score test
  - Likelihood ratio test
- Regularity Conditions
  - Number of parameters does not increase as n increases
  - Nuisance parameters not on boundary of parameter space. For parametric models this is not a problem.

Frailty (Random effects)

We can handle frailty in interval censored data:

- Parametric
  - Weibull model with Normal frailty. SAS code in appendix. Bellamy, et al (2004, Stat in Med, 3607-3621)
  - Gamma frailty. Goethals, et al (2009, JABES 1-14). Have SAS and R code.
- Semi-parametric
  - Proportional hazards with frailty. No software. Hougaard, et al (1994, Biometrics, 1178-88).

R Software for Interval Censoring: Overview

- NPMLE
  - Icens package: many algorithms
  - survival package: survfit (E-M algorithm)
  - interval package: icfit (E-M, polish using Kuhn-Tucker)
  - MLEcens package: does bivariate NPMLE, can be used for univariate

R Software for Interval Censoring: Overview

- NPMLE
  - Icens package: many algorithms
  - survival package: survfit (E-M algorithm)
  - interval package: icfit (E-M, polish using Kuhn-Tucker)
  - MLEcens package: does bivariate NPMLE, can be used for univariate
- Smoothing of Distribution
  - ICE package: uses Kernel smooth (reduces to NPMLE as bandwidth gets small)
  - polspline package: oldlogspline (spline estimator of density)

R Software for Interval Censoring: Overview

- NPMLE
  - Icens package: many algorithms
  - survival package: survfit (E-M algorithm)
  - interval package: icfit (E-M, polish using Kuhn-Tucker)
  - MLEcens package: does bivariate NPMLE, can be used for univariate
- Smoothing of Distribution
  - ICE package: uses Kernel smooth (reduces to NPMLE as bandwidth gets small)
  - polspline package: oldlogspline (spline estimator of density)
- Testing
  - interval package: ictest (weighted logrank tests, using permutation [exact, asymptotic], score test, or imputation)

R Software for Interval Censoring: Overview

- NPMLE
  - Icens package: many algorithms
  - survival package: survfit (E-M algorithm)
  - interval package: icfit (E-M, polish using Kuhn-Tucker)
  - MLEcens package: does bivariate NPMLE, can be used for univariate
- Smoothing of Distribution
  - ICE package: uses Kernel smooth (reduces to NPMLE as bandwidth gets small)
  - polspline package: oldlogspline (spline estimator of density)
- Testing
  - interval package: ictest (weighted logrank tests, using permutation [exact, asymptotic], score test, or imputation)
- Regression
  - survival package: survreg (parametric survival models)
  - Epi package: Icens (Piecewise constant intensity model)

Example Data Set for Regression

```
> library(Epi)
> data(hivDK)
```

R examples

Data Set: HIV in Danish men from 1983 to 1989 (see Carstensen, 1996, Stat in Med, 2177-2189)

- Data in Epi R package: hivDK
- **Modify Data: Treat as if known to be well (HIV antibody negative) at start of study. This mimics most clinical trials.**
- Time to event: event is HIV antibody positive. Start of study, 12/31/1980
- Time measured in days since start of study
- Sample size: n=297
  - Right censored: n=232
  - Left censored: n=26
  - interval censored: n=39
- Covariate: traveled to U.S. ? (No, n= 190), (Yes, n= 107)

hivDK data: Original form

```
> hivDK[1:10,]

    id      entry       well        ill bth pyr us
1  101 1980-12-31 1987-03-15       <NA>   0   0  0
2  104 1980-12-31 1989-05-15       <NA>   9  20  0
3  105 1980-12-31 1981-11-15       <NA>   8   3  0
4  106 1980-12-31 1989-05-15       <NA>  -5   8  0
5  107 1980-12-31 1981-11-15 1987-03-15  -2   5  0
6  108 1980-12-31 1987-03-15       <NA>  -9  25  0
7  109 1980-12-31 1984-08-15       <NA>  -8   3  0
8  110 1980-12-31 1987-03-15       <NA>  -3   6  0
9  111 1980-12-31 1984-08-15       <NA>  -3   4  0
10 112 1980-12-31 1981-11-15       <NA>   8   7  0
```

R code: setup/modify hivDK data

```
> library(Epi)
> data(hivDK)
> d<-hivDK
> left<-as.numeric(d$well-d$entry)
> left.na<-left
> left[is.na(left)]<-0
> right<- as.numeric(d$ill - d$entry)
> right.na<-right
> right[is.na(right)]<-Inf
> d<-data.frame(left,left.na,right,
+    right.na,us=d$us,
+    year.of.birth=d$bth+1950,
+    age.at.entry=d$bth+30,
+    partners.per.year=d$pyr)
```

hivDK data

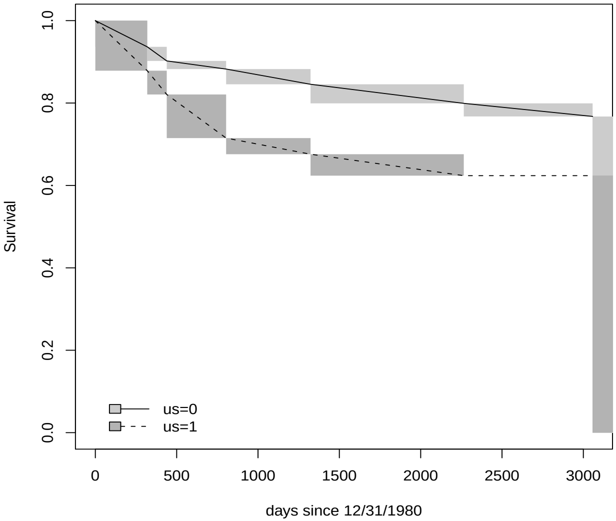
```
## for survreg, need left and right in different format
## created variables left.na and right.na such that
## left censoring has left.na=NA
## right censoring has right.na=NA
d[1:3,]
```

```
  left left.na right right.na year.of.birth
1 2265    2265   Inf       NA          1950
2 3057    3057   Inf       NA          1959
3  319     319   Inf       NA          1958
  age.at.entry partners.per.year
1           30                 0
2           39                20
3           38                 3
```

```
fit<-icfit(Surv(left,right,type="interval2")~us, data=d)
summary(fit)
```

```
us=0:
     Interval Probability
1      (0,319]      0.0636
2    (319,439]      0.0342
3    (439,804]      0.0197
4   (804,1323]      0.0370
5 (1323,2265]      0.0463
6 (2265,3057]      0.0317
7  (3057,Inf)      0.7675
us=1:
     Interval Probability
1      (0,319]      0.1215
2    (319,439]      0.0577
3    (439,804]      0.1058
4   (804,1323]      0.0391
5 (1323,2265]      0.0521
6  (3057,Inf)      0.6237
```

```
> plot(fit)
```



1.0
0.8
0.6
0.4
0.2
0.0
Survival
0
500
1000
1500
2000
2500
3000
days since 12/31/1980
us=0
us=1

```
> test<-ictest(Surv(left,right,type="interval2")~us, data=d)
> test


        Asymptotic Logrank two-sample test
        (permutation form), Sun's scores

data:  Surv(left, right, type = "interval2") by us
Z = -2.7393, p-value = 0.006156
alternative hypothesis: survival distributions not equal

    n Score Statistic*
0 190        -10.26357
1 107         10.26357
* like Obs-Exp, positive implies earlier failures
than expected
```

Rank Permutation Test for Trend

```
> ictest(Surv(left,right,type="interval2")~partners.per.year,
+     score="wmw",data=d)
```

```
        Asymptotic Wilcoxon trend test(permutation form)

data:  Surv(left, right, type = "interval2") by partners.per.year
Z = 3.0424, p-value = 0.002347
alternative hypothesis: survival distributions not equal

       n Score Statistic*
[1,] 297         514.0171
* postive so larger covariate values give earlier failures than expected
```

interval R package: Other capabilities

Other details outlined in vignette of interval software. In R, after loading interval package type:
vignette(”intervalPaper”)

- You can output scores, $c_1, \ldots, c_n$, that may then be used in other permutation software (e.g., coin R package, StatXact).
- You can estimate the NPMLE from another package (e.g., Icens) and input the results into ictest through the initfit option. Recall Icens has many options for calculating the NPMLE (e.g., E-M, Iterative Convex Minorant, Vector Exchange Algorithm).

```
## first try Weibull
## for survreg, need left and right in different format
## created variables left.na and right.na such that
## left censoring has left=NA
## right censoring has right=NA
> sreg<-survreg(Surv(left.na,right.na,type="interval2")~us,
+     data=d)
```

summary(sreg)

```
Call:
survreg(formula = Surv(left.na, right.na, type = "interval2") ~
    us, data = d)
            Value Std. Error     z        p
(Intercept) 10.36      0.507 20.43 8.74e-93
us          -1.17      0.463 -2.54 1.11e-02
Log(scale)   0.56      0.141  3.98 6.79e-05

Scale= 1.75

Weibull distribution
Loglik(model)= -214.7   Loglik(intercept only)= -218.3
        Chisq= 7.2 on 1 degrees of freedom, p= 0.0073
Number of Newton-Raphson Iterations: 8
n= 297
```

Interpretation of Weibull parameter

Fold-change in Time: $e^{\beta} = e^{-1.17} = .309$
95% Confidence Interval: $e^{\beta \pm 1.96 * se}$
The time to HIV seroconversion in Danish men who went to US is about 0.309 fold shorter than the time to HIV seroconversion of those who did not go to US.

```
> exp(cbind(sreg$coef,confint(sreg))["us",])


             2.5 %    97.5 %
0.3088391 0.1246499 0.7651960
```

Or opposite, time to HIV seroconversion is about 3.24 fold longer for those who never when to US compared to those who did:

```
> exp(-cbind(sreg$coef,confint(sreg))["us",])


            2.5 %   97.5 %
3.237932 8.022471 1.306855
```

Interpretation of Weibull parameter

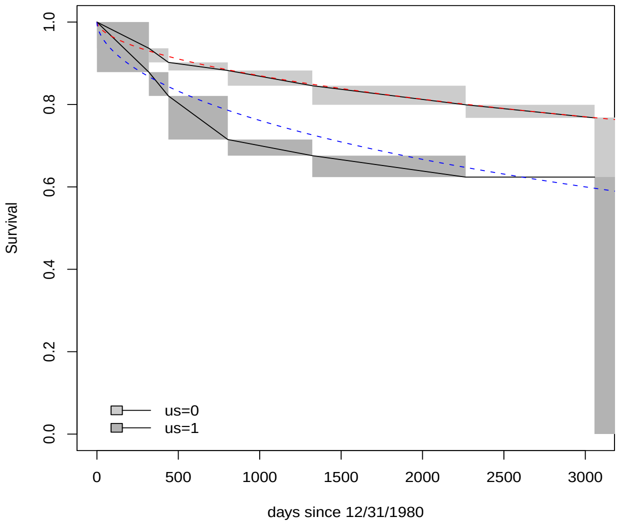
For Weibull, also interpret as a hazard ratio. The hazard of HIV seroconversion in Danish men who went to US is about 0.309 fold smaller than the hazard of HIV seroconversion in those who did not go to US.

```
exp(cbind(sreg$coef,confint(sreg))["us",])
```

```
              2.5 %    97.5 %
0.3088391 0.1246499 0.7651960
```

R code for plotting results

```
> fit<-icfit(Surv(left,right,type="interval2")~us,
+     data=d)
> plot(fit,LTY=c(1,1),XLAB="days since 12/31/1980",
+    main="NPMLEs with Weibull fit in color")
> pct<-1:999/1000
> ptime<-predict(sreg,newdata=data.frame(us=0),
+    type='quantile',p=pct)
> lines(ptime,1-pct,col="red",lty=2)
> ptime<-predict(sreg,newdata=data.frame(us=1),
+    type='quantile',p=pct)
> lines(ptime,1-pct,col="blue",lty=2)
```



NPMLEs with Weibull fit in color

```
sreg2<-survreg(Surv(left.na,right.na,type="interval2")~us,
dist="loglogistic",data=d)
summary(sreg2)
```

```
Call:
survreg(formula = Surv(left.na, right.na, type = "interval2") ~
    us, data = d, dist = "loglogistic")
             Value Std. Error     z         p
(Intercept)  9.862      0.457 21.56 4.35e-103
us          -1.229      0.466 -2.64  8.31e-03
Log(scale)   0.439      0.138  3.17  1.52e-03

Scale= 1.55

Log logistic distribution
Loglik(model)= -214.2   Loglik(intercept only)= -217.9
        Chisq= 7.56 on 1 degrees of freedom, p= 0.006
Number of Newton-Raphson Iterations: 4
n= 297
```

Interpretation of Log-logistic Model

Fold-change in Time: $e^{\beta} = e^{-1.23} = .293$
95% Confidence Interval: $e^{\beta \pm 1.96 * se}$
The time to HIV seroconversion in Danish men who went to US is about 0.293 fold shorter than the time to HIV seroconversion of those who did not go to US.

```
exp(cbind(sreg2$coef,confint(sreg2))["us",])
```

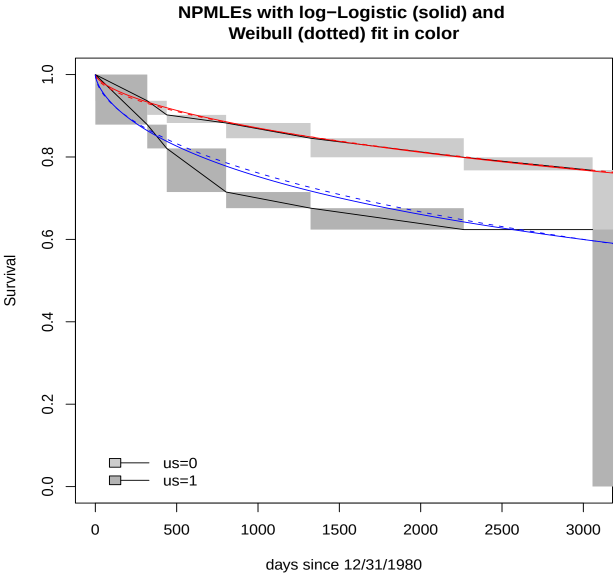
```
              2.5 %    97.5 %
0.2927120 0.1175371 0.7289639
```

Or opposite, those who did not go to US compared to those who did:

```
exp(-cbind(sreg2$coef,confint(sreg2))["us",])
```

```
            2.5 %   97.5 %
3.416327 8.507951 1.371810
```

```
> plot(fit,LTY=c(1,1),XLAB="days since 12/31/1980",
+    main="NPMLEs with log-Logistic (solid) and
+    Weibull (dotted) fit in color")
> pct<-1:999/1000
> ptime<-predict(sreg,newdata=data.frame(us=0),
+    type='quantile',p=pct)
> lines(ptime,1-pct,col="red",lty=2)
> ptime<-predict(sreg,newdata=data.frame(us=1),
+     type='quantile',p=pct)
> lines(ptime,1-pct,col="blue",lty=2)
> ptime2<-predict(sreg2,newdata=data.frame(us=0),
+    type='quantile',p=pct)
> lines(ptime2,1-pct,col="red",lty=1)
> ptime2<-predict(sreg2,newdata=data.frame(us=1),
+    type='quantile',p=pct)
> lines(ptime2,1-pct,col="blue",lty=1)
```


NPMLEs with log−Logistic (solid) and
Weibull (dotted) fit in color
Survival
0.0
0.2
0.4
0.6
0.8
1.0
0
500
1000
1500
2000
2500
3000
days since 12/31/1980
us=0
us=1

Weibull Regression with More than One Term

```
sreg.both<-survreg(
  Surv(left.na,right.na,type="interval2")~us+
  partners.per.year,data=d)
summary(sreg.both)
```

```
Call:
survreg(formula = Surv(left.na, right.na, type = "interval2") ~
    us + partners.per.year, data = d)
                    Value Std. Error     z        p
(Intercept)       10.5018    0.52097 20.16 2.29e-90
us                -0.8557    0.45988 -1.86 6.28e-02
partners.per.year -0.0183    0.00741 -2.47 1.33e-02
Log(scale)         0.5356    0.13974  3.83 1.27e-04

Scale= 1.71

Weibull distribution
Loglik(model)= -212   Loglik(intercept only)= -218.3
        Chisq= 12.58 on 2 degrees of freedom, p= 0.0019
Number of Newton-Raphson Iterations: 8
n= 297
```

Weibull Regression: Likelihood Ratio Test

```
> sreg.ppy<-survreg(
+  Surv(left.na,right.na,type="interval2")~partners.per.year,
+    data=d)
> sreg.both<-survreg(
+  Surv(left.na,right.na,type="interval2")~us+
+    partners.per.year,data=d)
> anova(sreg.ppy,sreg.both)
```

```
                   Terms Resid. Df    -2*LL Test Df
1      partners.per.year       294 427.7760      NA
2 us + partners.per.year       293 424.0726  +us  1
  Deviance   P(>|Chi|)
1       NA          NA
2 3.703415 0.05430123
```

End of Tutorial: Questions?

Getting R and R Packages

- R is freeware (available for PC, Mac, Unix): go to www.r-project.org
- To install packages for Windows version: Packages -> Install Packages (select CRAN mirror) (select package)
- Load in R program (for example, interval package): library(interval)

SAS Software for Interval Censoring: Overview

- NPMLE
  - Proc LIFEREG (E-M, polish using Kuhn-Tucker conditions)
- Regression
  - Proc LIFEREG: parametric survival models

What is Nelson-Aalen Estimator?

Review:

- Survival function:

$$S(t) = Pr[T > t]$$

- Density function:

$$f(t) = \frac{-\partial S(t)}{\partial t}$$

- hazard rate:

$$\begin{aligned} \lambda(t) &= \lim_{\Delta \to 0} Pr[t \leq T \leq t + \Delta \mid T \geq t] \\ &= \frac{f(t)}{S(t-)} \end{aligned}$$

- Solve differential equation:

$$\frac{-\partial S(t)}{\partial t} = \lambda(t) S(t-)$$

$$\Rightarrow \qquad S(t-) = \exp\left(-\int_0^t \lambda(t)\right)$$

What is Nelson-Aalen Estimator?

- $$S(t-) = \exp\left(-\int_0^t \lambda(t)\right)$$

- Nelson-Aalen estimator (right censored data):

$$\begin{aligned}\tilde{S}(t-) &= \exp\left(-\int_0^t \hat{\lambda}(t)\right)\\ &= \exp\left(-\sum_{j:t_j<t} \hat{\lambda}(t_j)\right)\end{aligned}$$

- Kaplan-Meier estimator (right censored data):

$$\begin{aligned}\hat{S}(t-) &= \mathcal{P}_{s<t}\left(1 - d\hat{\Lambda}(s)\right)\\ &= \prod_{j:t_j<t}\left(1 - \hat{\lambda}(t_j)\right)\end{aligned}$$