Smoothing Hazard Function by using Uniform kernel Epachnikov kernel and Biweight kernel

Rongjie Huang, Xiran Wang, Anja Zgodic
University of South Carolina

3/2020

Section 6.2: Estimating Hazard Function

$$\tilde{H}(t) = \begin{cases} 0, & \text{if } t \leq t_1, \\ \sum_{t_i \leq t} \frac{d_i}{Y_i}, & \text{if } t_1 \leq t. \end{cases}$$

$$\sigma_H^2(t) = \sum_{t_i \leq t} \frac{d_i}{Y_i^2}.$$

Section 6.2: Estimating Hazard Function

- Symmetric Kernels:

The uniform kernel gives equal weight to all deaths in the interval *t* - *b* to *t* + *b*, whereas the other two kernels give progressively heavier weigh points close to *t*.

uniform kernel with

$$K(x) = 1/2 \quad \text{for } -1 \leq x \leq 1,$$

Epanechnikov kernel with

$$K(x) = 0.75(1 - x^2) \quad \text{for } -1 \leq x \leq 1,$$

biweight kernel with

$$K(x) = \frac{15}{16}(1 - x^2)^2 \quad \text{for } -1 \leq x \leq 1.$$

Section 6.2: Estimating Hazard Function

- Asymmetric Kernels

uniform kernel

$$K_q(x) = \frac{4(1+q^3)}{(1+q)^4} + \frac{6(1-q)}{(1+q)^3}x, \quad \text{for } -1 \le x \le q,$$

$$q = \begin{cases} t/b\,, & t < b \\ (t_d - t)/b\,, & t > t_d - b \end{cases}$$

Epanechnikov kernel

$$K_q(x) = K(x)(\alpha_E + \beta_E x), \quad \text{for } -1 \le x \le q,$$

where

$$\alpha_E = \frac{64(2 - 4q + 6q^2 - 3q^3)}{(1+q)^4(19 - 18q + 3q^2)}$$

and

$$\beta_E = \frac{240(1-q)^2}{(1+q)^4(19 - 18q + 3q^2)},$$

Section 6.2: Estimating Hazard Function

- Asymmetric Kernels

biweight kernel

$$K_q(x) = K(x)(\alpha_{BW} + \beta_{BW} x), \quad \text{for } -1 \le x \le q,$$

where

$$\alpha_{BW} = \frac{64(8 - 24q + 48q^2 - 45q^3 + 15q^4)}{(1+q)^5(81 - 168q + 126q^2 - 40q^3 + 5q^4)}$$

and

$$\beta_{BW} = \frac{1120(1-q)^3}{(1+q)^5(81 - 168q + 126q^2 - 40q^3 + 5q^4)}.$$

Section 6.2: Estimating Hazard Function

- Estimating the Hazard Function with Kernels

For time points t for which $b \leq t \leq t_D - b$, the kernel-smoothed estimator of $h(t)$ based on the kernel $K()$ is given by

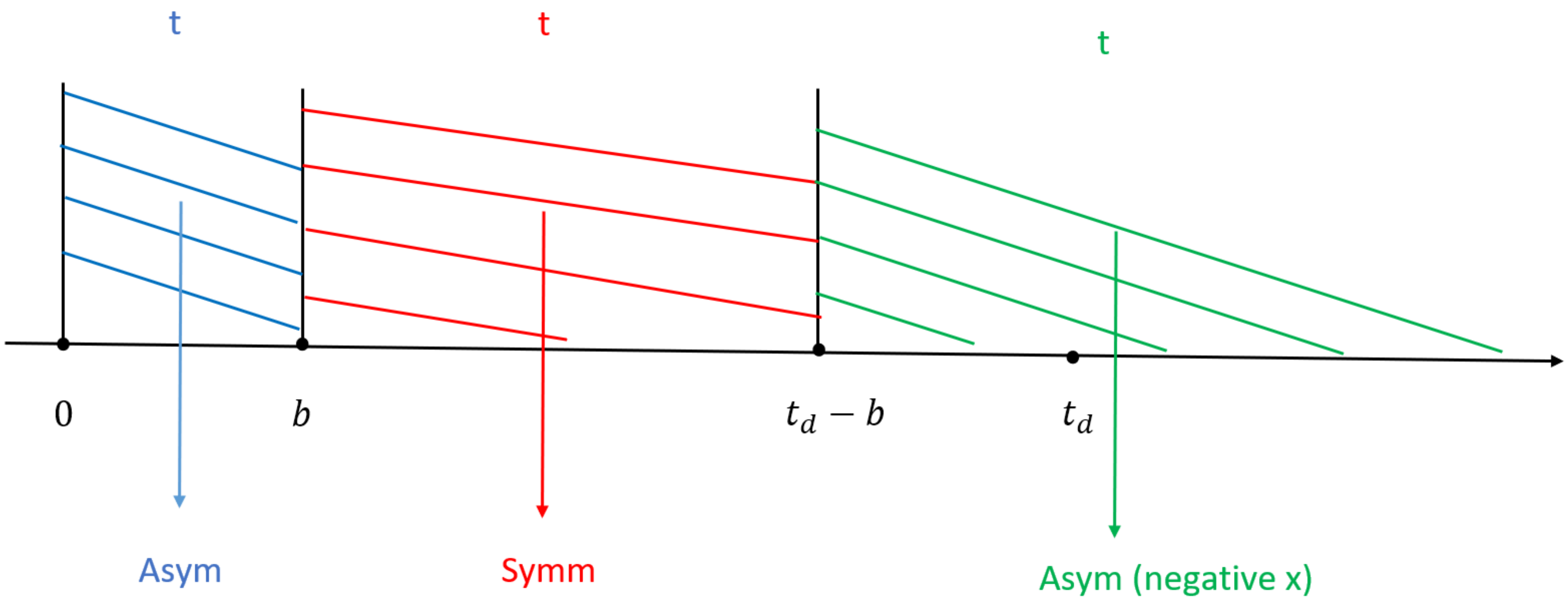
$$\hat{h}(t) = b^{-1} \sum_{i=1}^{D} K\left(\frac{t - t_i}{b}\right) \Delta \tilde{H}(t_i).$$

The variance of $\hat{h}(t)$ is estimated by the quantity

$$\sigma^2[\hat{h}(t)] = b^{-2} \sum_{i=1}^{D} K\left(\frac{t - t_i}{b}\right)^2 \Delta \hat{V}[\tilde{H}(t_i)].$$

$$\hat{h}(t) \exp\left[\pm \frac{Z_{1-\alpha/2}\sigma(\hat{h}(t))}{\hat{h}(t)}\right]$$

Symmetric vs. Asymmetric Kernels



Section 6.2: optimization function for bandwidth b

- The mean integrated squared error

(MISE) of $\hat{h}$ over the range τ_{L} to τ_{U} defined by

$$\begin{aligned} MISE(b) &= E\int_{\tau_{\mathrm{L}}}^{\tau_{\mathrm{U}}} [\hat{h}(u) - h(u)]^2 du \\ &= E\int_{\tau_{\mathrm{L}}}^{\tau_{\mathrm{U}}} \hat{h}^2(u)du - 2E\int_{\tau_{\mathrm{L}}}^{\tau_{\mathrm{U}}} \hat{h}(u)h(u)du + E\int_{\tau_{\mathrm{L}}}^{\tau_{\mathrm{U}}} h^2(u)du. \end{aligned}$$

- ➢ Trapezoidal rule: $$\int_a^b f(x)\,dx \approx \sum_{k=1}^{N} \frac{f(x_{k-1}) + f(x_k)}{2}\Delta x_k$$
- ➢ Cross validation

Section 6.2: Estimating Hazard Function

- Optimization function for bandwidth b

find b which minimizes the function

$$g(b) = \sum_{i=1}^{M-1} \left(\frac{u_{i+1} - u_i}{2} \right) [\hat{h}^2(u_i) + \hat{h}^2(u_{i+1})]$$

$$- 2b^{-1} \sum_{i \neq j} K \left(\frac{t_i - t_j}{b} \right) \Delta \tilde{H}(t_i) \Delta \tilde{H}(t_j).$$

Datasets

- Appendix D: 137 Bone Marrow Transplant Patients Pg.485
  - 137 rows and 22 columns
  - Right censor data
  - 3 groups: All, Low and high

```
> bone$time <- bone$t2
> bone$status <- bone$dfree
> head(bone)
  g   t1   t2 death relapse dfree   ta a  tc c tp p z1 z2 z3 z4 z5 z6   z7 z8 z9 z10 time status
1 1 2081 2081     0       0     0   67 1 121 1 13 1 26 33  1  0  1  1   98  0  1   0 2081      0
2 1 1602 1602     0       0     0 1602 0 139 1 18 1 21 37  1  1  0  0 1720  0  1   0 1602      0
3 1 1496 1496     0       0     0 1496 0 307 1 12 1 26 35  1  1  1  0  127  0  1   0 1496      0
4 1 1462 1462     0       0     0   70 1  95 1 13 1 17 21  0  1  0  0  168  0  1   0 1462      0
5 1 1433 1433     0       0     0 1433 0 236 1 12 1 32 36  1  1  1  1   93  0  1   0 1433      0
6 1 1377 1377     0       0     0 1377 0 123 1 12 1 22 31  1  1  1  1 2187  0  1   0 1377      0
```

Datasets

- Death Times of Kidney Transplant Patients
  - 863 rows and 7 columns
  - Right censor data

```
> kid$time <- kid$time/365
> kid$status <- kid$death
> head(kid )
  obs        time death gender race age status
1   1 0.002739726     0      1    1  46      0
2   2 0.013698630     0      1    1  51      0
3   3 0.019178082     1      1    1  55      1
4   4 0.024657534     0      1    1  57      0
5   5 0.035616438     0      1    1  45      0
6   6 0.035616438     0      1    1  43      0
```

Result comparisons: Kernel function

Uniform, Epanechnikov

and biweight kernel function:

- k.unif(t, ti, td, b){ return(k) }
- k.ep(t, ti, td, b){ return(k) }
- k.biw(t, ti, td, b){ return(k) }

t- interested time

ti- event time from dataset

td- maximum time point in the events

b - bandwidth

TABLE 6.1

Weights Used in Smoothing the Nelson–Aalen Estimator for the ALL Group

t_i	$\Delta\tilde{H}(t_i)]$	$\Delta\hat{V}[\tilde{H}(t_i)]$	$\frac{150-t_i}{100}$	$K\left(\frac{150-t_i}{100}\right)$	$\frac{50-t_i}{100}$	$K\left(\frac{50-t_i}{100}\right)$	$\frac{600-t_i}{100}$	$K\left(\frac{600-t_i}{100}\right)$
1	0.0263	0.00069	1.49	0.0000	0.49	1.0618	5.99	0.0000
55	0.0270	0.00073	0.95	0.0731	−0.05	0.9485	5.45	0.0000
74	0.0278	0.00077	0.76	0.3168	−0.24	0.7482	5.26	0.0000
86	0.0286	0.00082	0.64	0.4428	−0.36	0.6047	5.14	0.0000
104	0.0294	0.00087	0.46	0.5913	−0.54	0.3867	4.96	0.0000
107	0.0303	0.00091	0.43	0.6113	−0.57	0.3518	4.93	0.0000
109	0.0313	0.00099	0.41	0.6239	−0.59	0.3290	4.91	0.0000
110	0.0322	0.00103	0.40	0.6300	−0.60	0.3177	4.90	0.0000
122	0.0667	0.00222	0.28	0.6912	−0.72	0.1913	4.78	0.0000
129	0.0357	0.00128	0.21	0.7169	−0.79	0.1275	4.71	0.0000
172	0.0370	0.00138	−0.22	0.7137	−1.22	0.0000	4.28	0.0000
192	0.0385	0.00147	−0.42	0.6177	−1.42	0.0000	4.08	0.0000
194	0.0400	0.00161	−0.44	0.6048	−1.44	0.0000	4.06	0.0000
230	0.0435	0.00188	−0.80	0.2700	−1.80	0.0000	3.70	0.0000
276	0.0454	0.00207	−1.26	0.0000	−2.26	0.0000	3.24	0.0000
332	0.0476	0.00228	−1.82	0.0000	−2.82	0.0000	2.68	0.0000
383	0.0500	0.00247	−2.33	0.0000	−3.33	0.0000	2.17	0.0000
418	0.0527	0.00277	−2.68	0.0000	−3.68	0.0000	1.82	0.0000
468	0.0555	0.00310	−3.18	0.0000	−4.18	0.0000	1.32	0.0000
487	0.0589	0.00345	−3.37	0.0000	−4.37	0.0000	1.13	0.0000
526	0.0625	0.00391	−3.76	0.0000	−4.76	0.0000	0.74	0.2492
609	0.0714	0.00511	−4.59	0.0000	−5.59	0.0000	−0.09	0.8918
662	0.0769	0.00592	−5.12	0.0000	−6.12	0.0000	−0.62	0.6904

Result comparisons: Kernel function

Uniform, Epanechnikov

and biweight kernel function:

- k.unif(t, ti, td, b){ return(k) }
- k.ep(t, ti, td, b){ return(k) }
- k.biw(t, ti, td, b){ return(k) }

t- interested time

ti- eventime from dataset

td- maimum time point in the events

b - bandwidth

	group	t_i	delta_cumhaz	delta_varcumhar	150-t_i/100	50-t_i/100	600-t_i/100	K(150-t_i/100)	K(50-t_i/100)	K(600-t_i/100)
1	1	1	0.02631579	0.0006925208	1.49	0.49	5.99	0.000000	1.0618964	0.0000000
2	1	55	0.02702703	0.0007304602	0.95	-0.05	5.45	0.073125	0.9485271	0.0000000
3	1	74	0.02777778	0.0007716049	0.76	-0.24	5.26	0.316800	0.7480757	0.0000000
4	1	86	0.02857143	0.0008163265	0.64	-0.36	5.14	0.442800	0.6045569	0.0000000
5	1	104	0.02941176	0.0008650519	0.46	-0.54	4.96	0.591300	0.3865997	0.0000000
6	1	107	0.03030303	0.0009182736	0.43	-0.57	4.93	0.611325	0.3516800	0.0000000
7	1	109	0.03125000	0.0009765625	0.41	-0.59	4.91	0.623925	0.3288136	0.0000000
8	1	110	0.03225806	0.0010405827	0.40	-0.60	4.90	0.630000	0.3175194	0.0000000
9	1	122	0.06666667	0.0022222222	0.28	-0.72	4.78	0.691200	0.1911467	0.0000000
10	1	129	0.03571429	0.0012755102	0.21	-0.79	4.71	0.716925	0.1274369	0.0000000
11	1	172	0.03703704	0.0013717421	-0.22	-1.22	4.28	0.713700	0.0000000	0.0000000
12	1	192	0.03846154	0.0014792899	-0.42	-1.42	4.08	0.617700	0.0000000	0.0000000
13	1	194	0.04000000	0.0016000000	-0.44	-1.44	4.06	0.604800	0.0000000	0.0000000
14	1	230	0.04347826	0.0018903592	-0.80	-1.80	3.70	0.270000	0.0000000	0.0000000
15	1	276	0.04545455	0.0020661157	-1.26	-2.26	3.24	0.000000	0.0000000	0.0000000
16	1	332	0.04761905	0.0022675737	-1.82	-2.82	2.68	0.000000	0.0000000	0.0000000
17	1	383	0.05000000	0.0025000000	-2.33	-3.33	2.17	0.000000	0.0000000	0.0000000
18	1	418	0.05263158	0.0027700831	-2.68	-3.68	1.82	0.000000	0.0000000	0.0000000
19	1	466	0.05555556	0.0030864198	-3.16	-4.16	1.34	0.000000	0.0000000	0.0000000
20	1	487	0.05882353	0.0034602076	-3.37	-4.37	1.13	0.000000	0.0000000	0.0000000
21	1	526	0.06250000	0.0039062500	-3.76	-4.76	0.74	0.000000	0.0000000	0.2491607
22	1	609	0.07142857	0.0051020408	-4.59	-5.59	-0.09	0.000000	0.0000000	0.8917628
23	1	662	0.07692308	0.0059171598	-5.12	-6.12	-0.62	0.000000	0.0000000	0.6903636

Result comparisons: smoothed estimates of hazard rates by groups

- est_haz(data, b, timegrid, group=c(' ', 'g'), kernel=c('unif','ep','biw')) {
  return(estimated hazard rate (by group)) }

data- dataset, **status** and **time** need to be specified by user

b- bandwidth

timegird- time grid specified by user

group - Display output results by group if 'g' is specified

kernel - three kernel smooth methods can be chosen, 'unif' - unifrom method, 'ep' -Epanechnikov method and 'biw'- biweight metod.

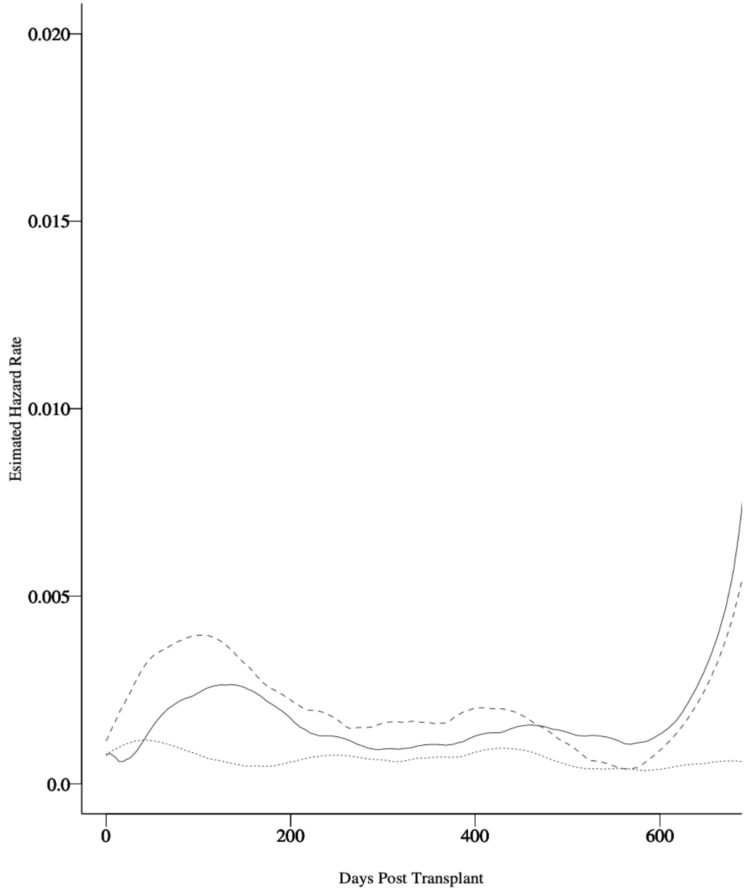


Figure 6.1 *Smoothed estimates of the hazard rates for bone marrow transplant patients based on the Epanechnikov kernel with a bandwidth of 100 days. ALL (———); AML-Low risk (--------); AML-High risk (— — —).*

Result comparisons: smoothed estimates of hazard rates by groups

- est_haz(data, b, timegrid, group=c(' ', 'g'), kernel=c('unif','ep','biw')) {
  return(estimated hazard rate (by group)) }

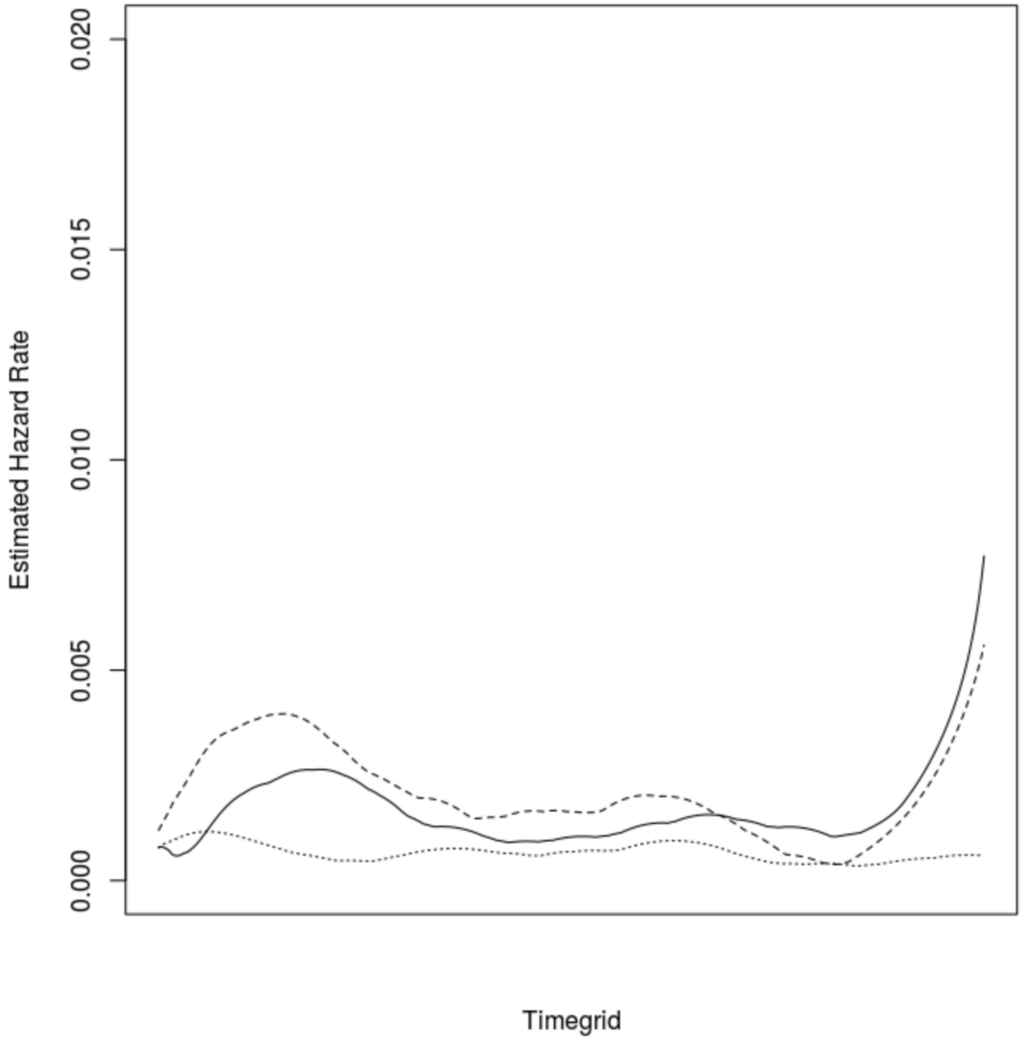
data- dataset, **status** and **time** need to be specified by user

b- bandwidth

timegird- time grid specified by user

group - Display output results by group if 'g' is specified

kernel - three kernel smooth methods can be chosen, 'unif' - unifrom method, 'ep' -Epanechnikov method and 'biw'- biweight metod.



Result comparisons: Estimated cumulative hazard

•cum_haz_plot (data){

```
   fit <- survfit(Surv(time, status) ~ 1, data=data,
ctype=1)
   t_i <- summary(fit)$time
   cumhaz <- summary(fit)$cumha

   plot(...)

return( data.frame( time, cum. hazard) )
}
```

data- Dataset with specified time and status

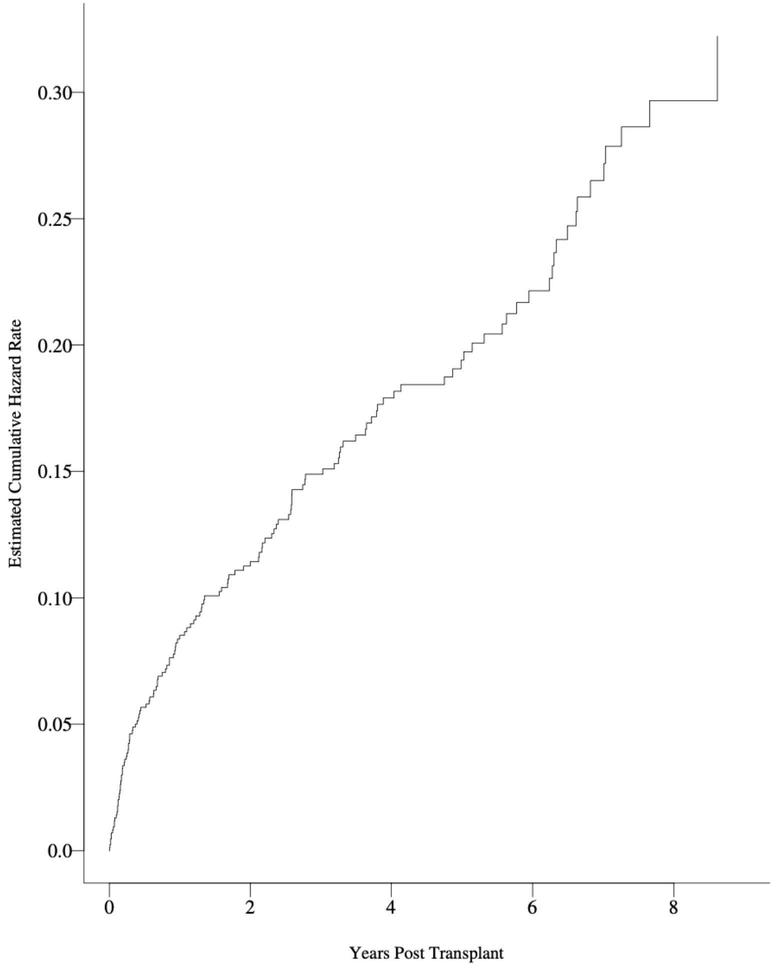
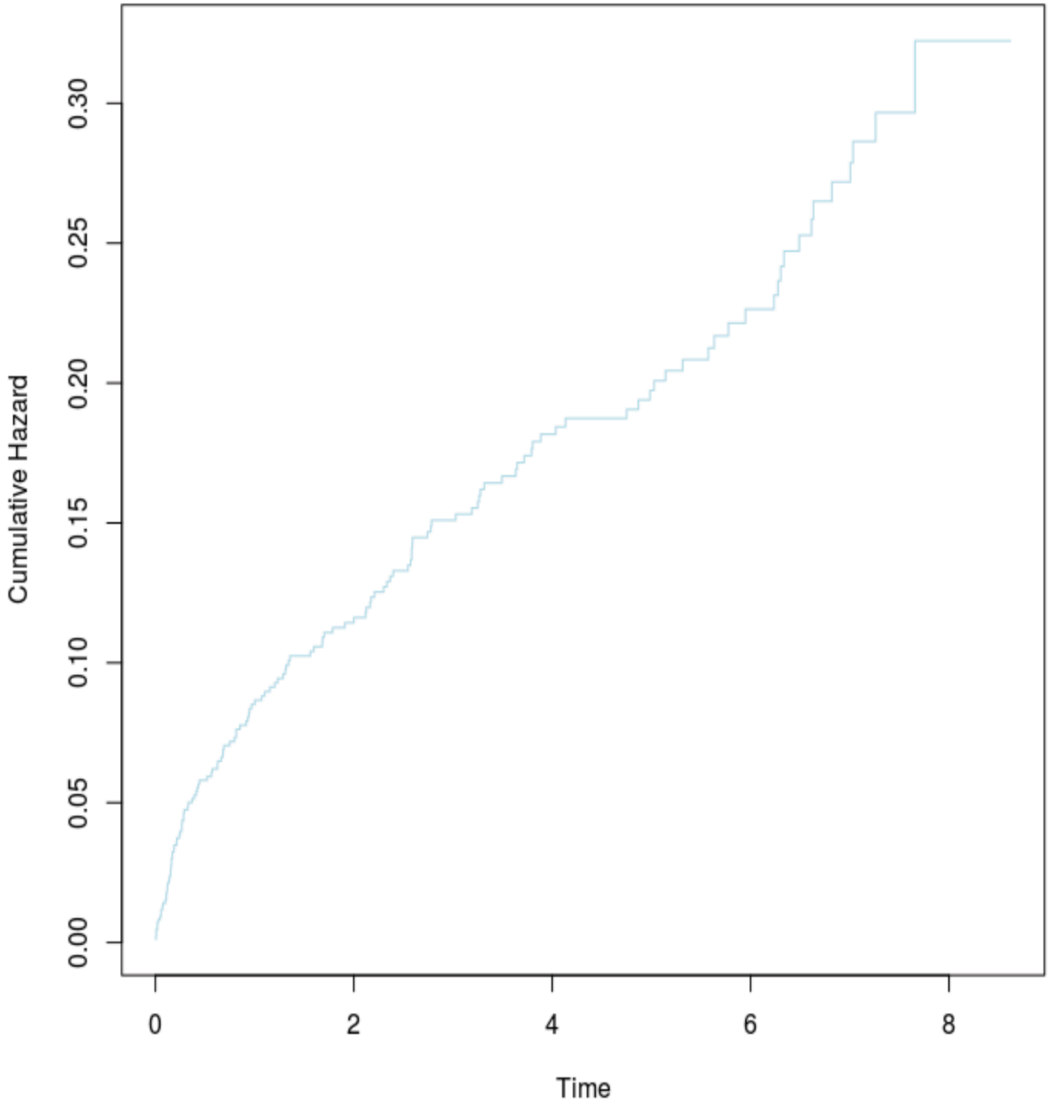


Figure 6.2 *Estimated cumulative hazard rate for kidney transplant patients*

Result comparisons: Estimated cumulative hazard

```
•cum_haz_plot (data){

   fit <- survfit(Surv(time, status) ~ 1, data=data,
ctype=1)
   t_i <- summary(fit)$time
   cumhaz <- summary(fit)$cumha

   plot(...)

return( data.frame( time, cum. hazard) )
}
```

data- Dataset with specified time and status



Result comparisons: Effects of changing the bandwidth on the smoothed hazard rate estimates

```
• est_haz_plot_by_bandwidth(data=data,
  timegrid=timegrid, max_haz = max_haz, kernel='ep'){

   ep_b_0.5 <- est_haz(data=data, b=0.5, timegrid=timegrid,
group='', kernel=kernel)
   ep_b_1 <- est_haz(data=data, b=1, timegrid=timegrid,
group=' ', kernel=kernel)
  ep_b_1.5 <- est_haz(data=data, b=1.5, timegrid=timegrid,
group='', kernel=kernel)
  ep_b_2 <- est_haz(data=data, b=2, timegrid=timegrid,
group=' ', kernel=kernel)

   plot(..); lines(...); lines(...); lines(...)
  out <- data.frame( timegrid, ep_b_0.5$haz, ep_b_1$haz,
ep_b_1.5$haz, ep_b_2$haz)
 return(out)
}
```

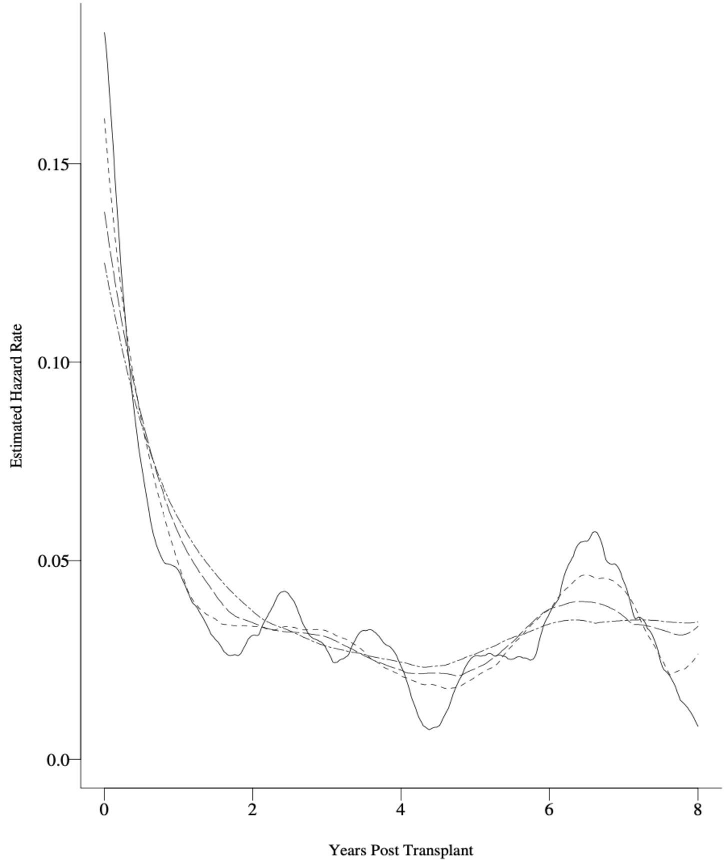
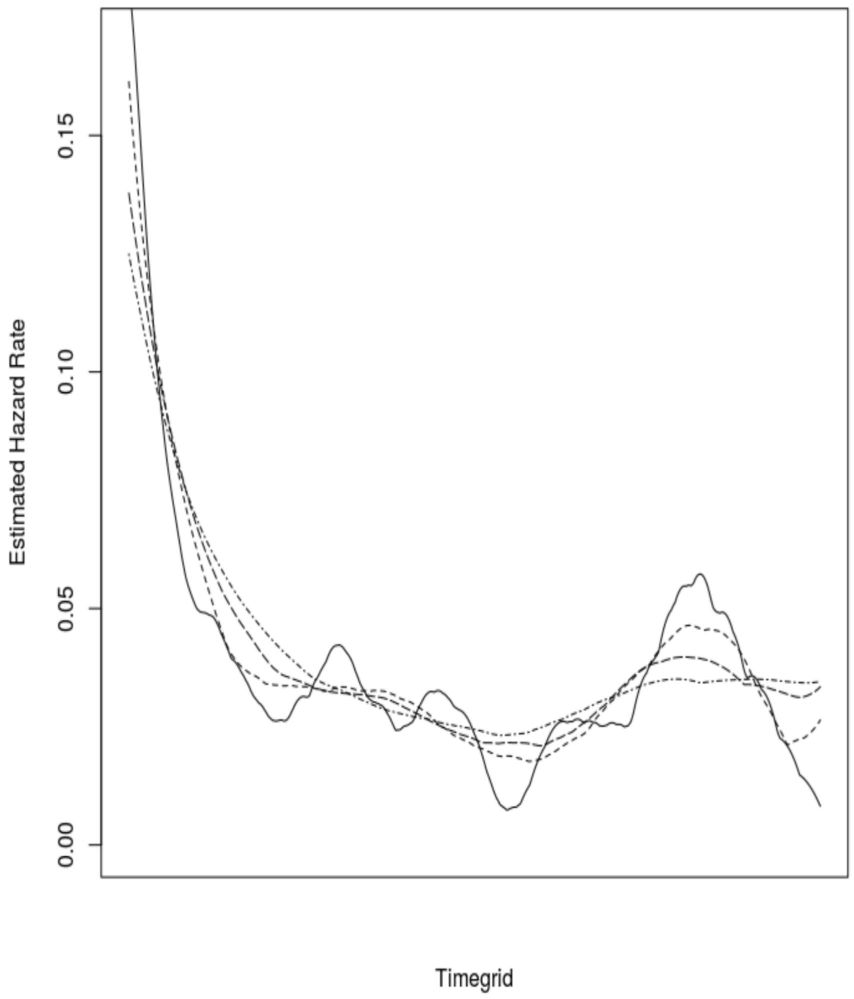


Figure 6.4 *Effects of changing the bandwidth on the smoothed hazard rate estimates for kidney transplant patients using the Epanechnikov kernel. bandwidth = 0.5 years (———) bandwidth = 1.0 years (-------) bandwidth = 1.5 years (— — —) bandwidth = 2.0 years (— · — · —)*

Result comparisons: Effects of changing the bandwidth on the smoothed hazard rate estimates

```
• est_haz_plot_by_bandwidth(data=data,
  timegrid=timegrid, max_haz = max_haz, kernel='ep'){

   ep_b_0.5 <- est_haz(data=data, b=0.5, timegrid=timegrid,
group='', kernel=kernel)
   ep_b_1 <- est_haz(data=data, b=1, timegrid=timegrid,
group=' ', kernel=kernel)
  ep_b_1.5 <- est_haz(data=data, b=1.5, timegrid=timegrid,
group='', kernel=kernel)
  ep_b_2 <- est_haz(data=data, b=2, timegrid=timegrid,
group=' ', kernel=kernel)

   plot(..); lines(...); lines(...); lines(...)
  out <- data.frame( timegrid, ep_b_0.5$haz, ep_b_1$haz,
ep_b_1.5$haz, ep_b_2$haz)
 return(out)
}
```



Result comparisons: Effects of changing the kernel on the smoothed estimates of hazard rates

```
• est_haz_plot_by_kernel(data=data, b=b, timegrid=timegrid,
     max_haz = max_haz){
     ep <- est_haz(data=data, b=b, timegrid=timegrid, group='',
kernel='ep')
     unif <- est_haz(data=data, b=b, timegrid=timegrid, group='',
kernel='unif')
     biw <- est_haz(data=data, b=b, timegrid=timegrid, group='',
kernel='biw')

     Plot (...);  lines(...); lines(...)

  out <- data.frame(timegrid, ep$haz, unif$haz, biw$haz)
  return(out)
}
```

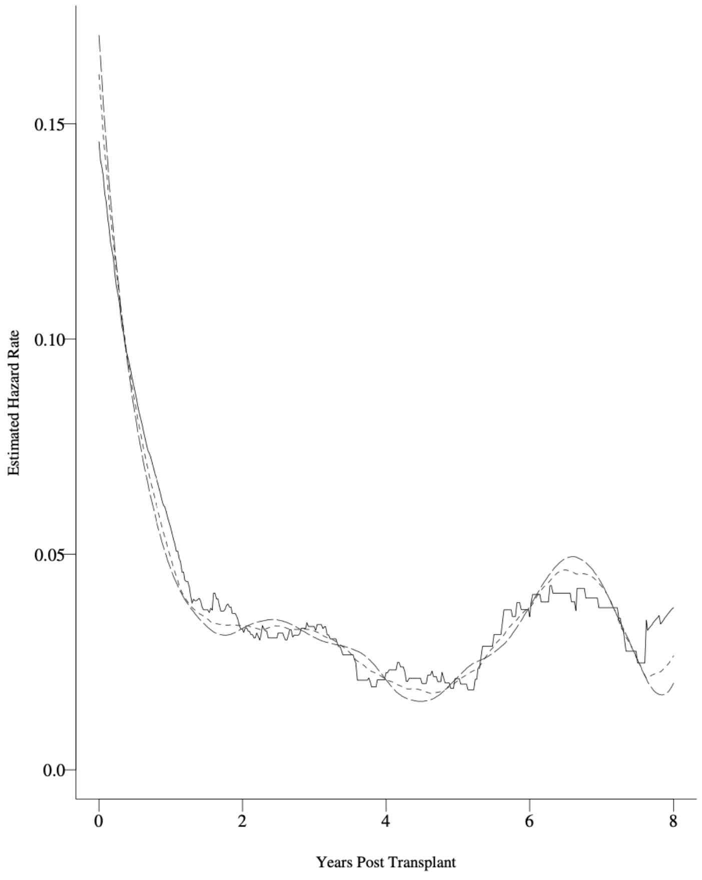
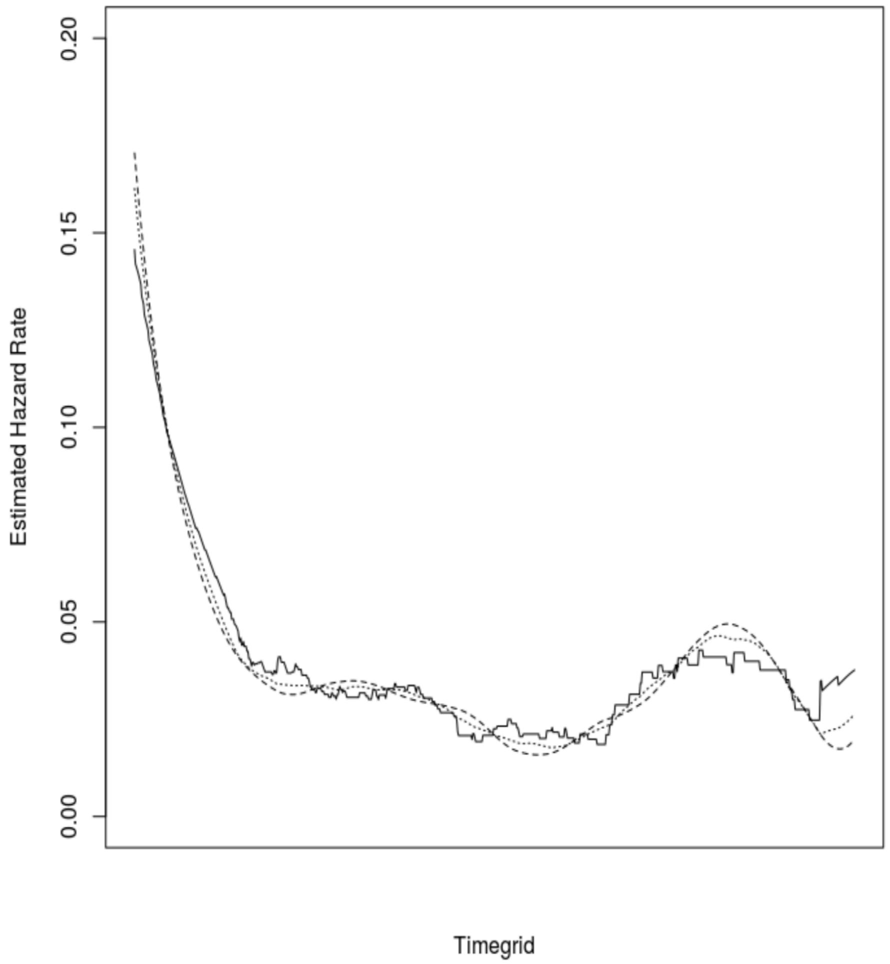


Figure 6.3 *Effects of changing the kernel on the smoothed hazard rate estimates for kidney transplant patients using a bandwidth of 1 year. Uniform kernel (———); Epanechnikov kernel (------) Biweight kernel (— — —)*

Result comparisons: Effects of changing the kernel on the smoothed estimates of hazard rates

```
est_haz_plot_by_kernel(data=data, b=b, timegrid=timegrid,
    max_haz = max_haz){
    ep <- est_haz(data=data, b=b, timegrid=timegrid, group='',
kernel='ep')
    unif <- est_haz(data=data, b=b, timegrid=timegrid, group='',
kernel='unif')
    biw <- est_haz(data=data, b=b, timegrid=timegrid, group='',
kernel='biw')

    Plot (...);  lines(...); lines(...)

  out <- data.frame(timegrid, ep$haz, unif$haz, biw$haz)
  return(out)
}
```



Result comparisons: Estimated risk function

- opti_gb(b, u, data, kernel=c('unif','ep','biw')){

  1st loop for **u** – return vector **gb1**

  If(kernel==kernel) { 2nd loop for t } -return matrix **gb2**

  diag(gb2) <-0

  sum(gb1)+sum(gb2)

  return(gb) }

- Recall g(b) function:

$$g(b) = \sum_{i=1}^{M-1} \left(\frac{u_{i+1} - u_i}{2} \right) [\hat{h}^2(u_i) + \hat{h}^2(u_{i+1})] - 2b^{-1} \sum_{i \neq j} K \left(\frac{t_i - t_j}{b} \right) \Delta \tilde{H}(t_i) \Delta \tilde{H}(t_j).$$

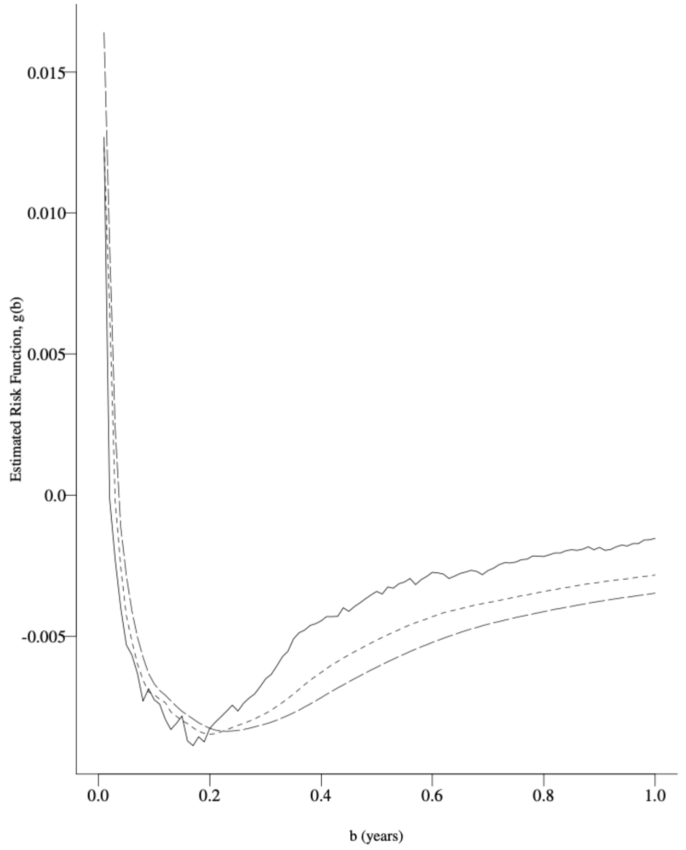
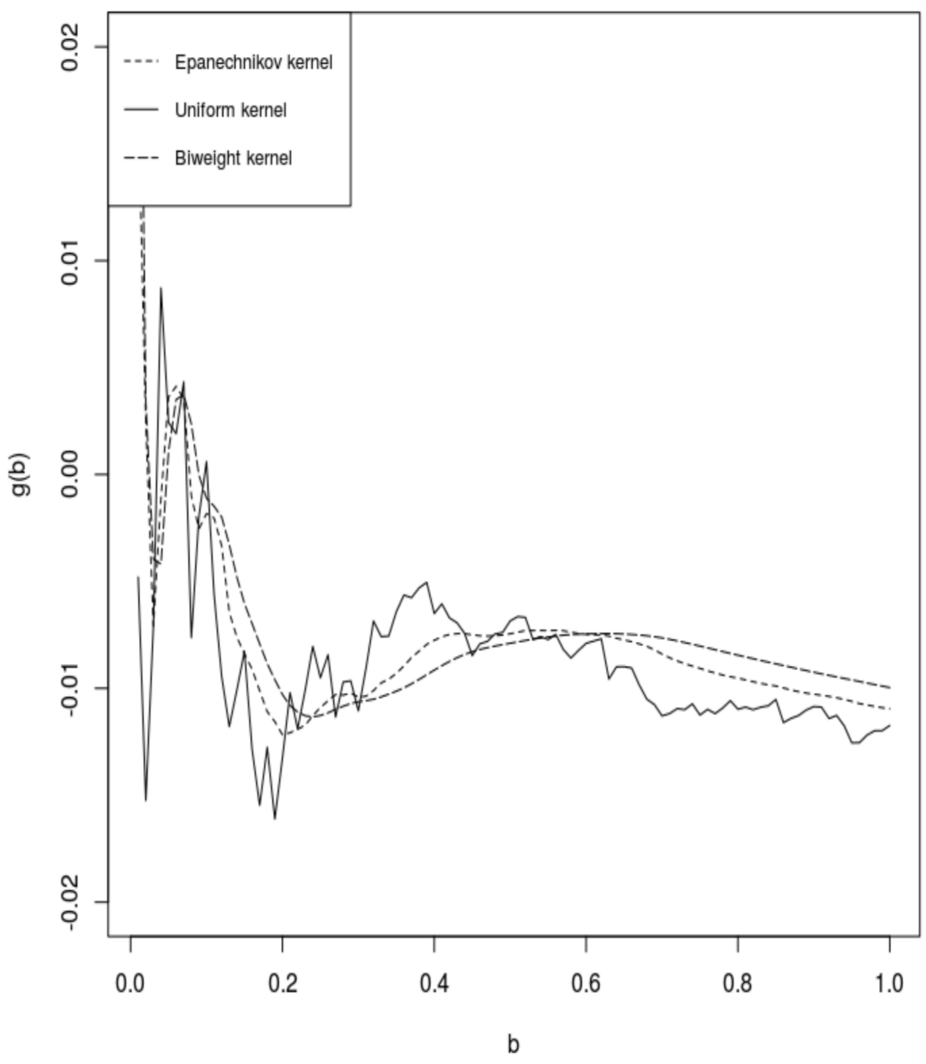


Figure 6.5 *Estimated risk function, g(b), for use in determination of the best bandwidth for the kidney transplant data. Uniform kernel (———); Epanechnikov kernel (------) Biweight kernel (— — —).*

Result comparisons: Estimated risk function

```
optimal_b_plot(u, bgrid,data ){
  for (p in 1:length(bgrid)) {

   pts_ep[p]<-opti_gb(b=bgrid[p],u=u,data=data, kernel='ep')
   pts_unif[p]<-opti_gb(b=bgrid[p],u=u,data=data, kernel='unif')
   pts_biw[p]<-opti_gb(b=bgrid[p],u=u,data=data, kernel='biw')
  }
 ep_optim <- bgrid[(pts_ep==min(pts_ep))]
 unif_optim <-bgrid[(pts_unif==min(pts_unif))]
 biw_optim <- bgrid[(pts_biw==min(pts_biw))]

 out <- data.frame(ep_optim,unif_optim,biw_optim)

 return(out)
}
```

- b=seq(0,6,1)



Result comparisons: Smoothed estimates of the hazard rates for using optimal bandwidths.

- est_haz_optim_group_plot(data, grp='g', timegrid, max_haz = max_haz, kernel=kernel, optimal_b){

```
  haz_1 <- est_haz(data=data[data[,grp]==1,], b=optimal_b[1],
timegrid=timegrid, group='', kernel=kernel)$haz

  haz_2 <- est_haz(data=data[data[,grp]==2,], b=optimal_b[2],
timegrid=timegrid, group='', kernel=kernel)$haz

  haz_3 <- est_haz(data=data[data[,grp]==3,], b=optimal_b[3],
timegrid=timegrid, group='', kernel=kernel)$haz

  plot(...); lines(..);  lines(...)

  return(data.frame(timegrid, haz_1, haz_2, haz_3))

}
```

- optimal_b is c(161, 50, 112)

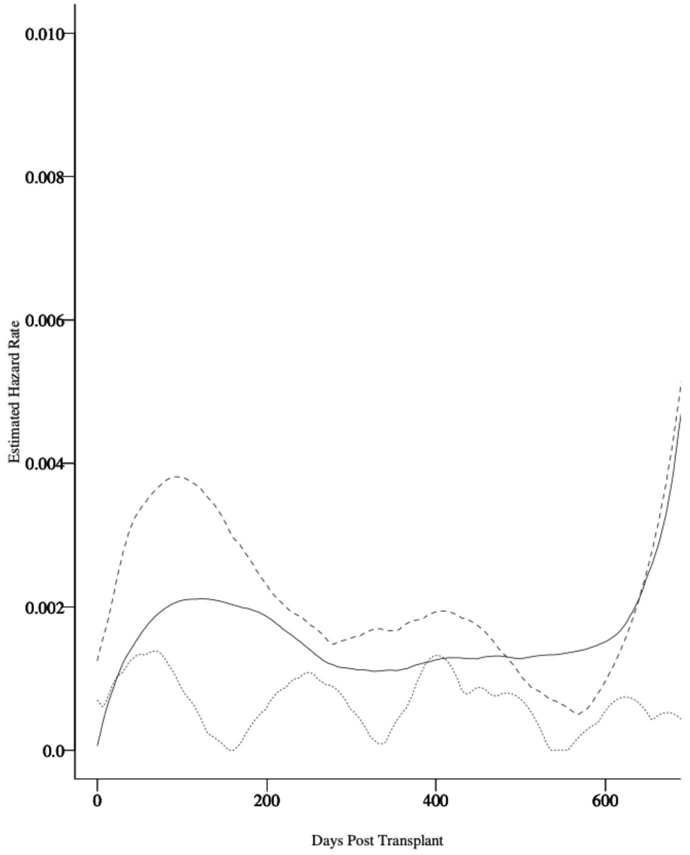


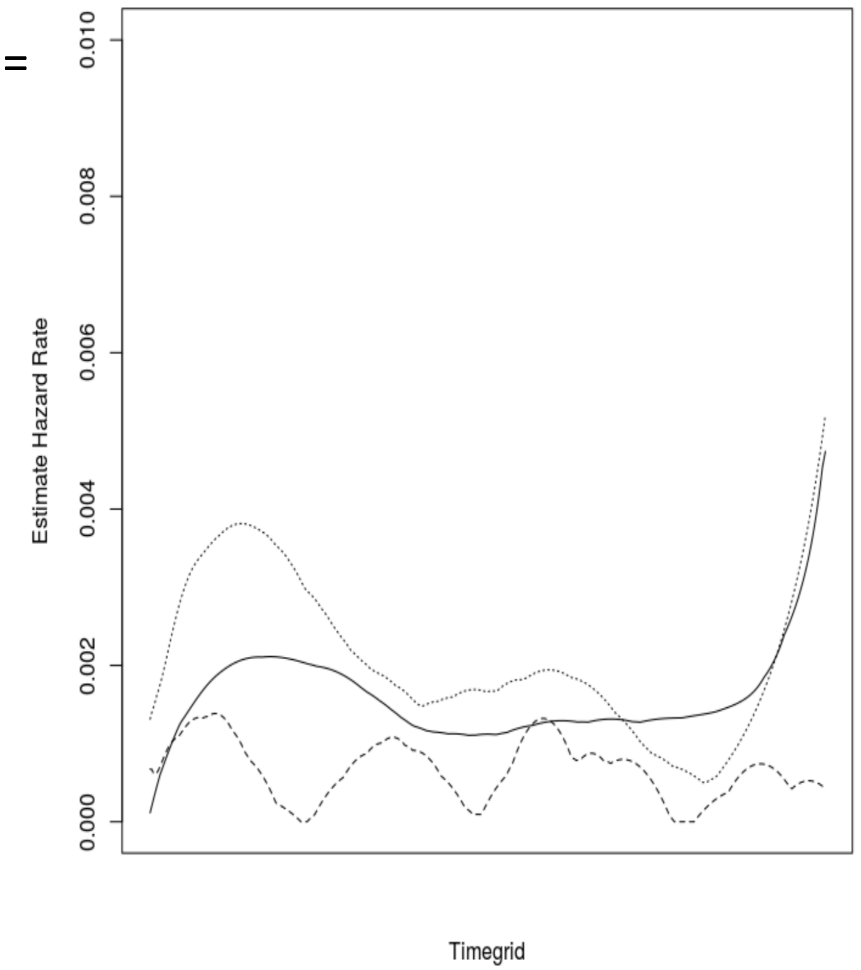
Figure 6.7 *Smoothed estimates of the hazard rates for bone marrow transplant patients based on the Epanechnikov kernel using optimal bandwidths. AML-Low risk (------) AML-High risk (———) ALL (———)*

Result comparisons: Smoothed estimates of the hazard rates for using optimal bandwidths.

- est_haz_optim_group_plot(data, grp='g', timegrid, max_haz = max_haz, kernel=kernel, optimal_b){

```
haz_1 <- est_haz(data=data[data[,grp]==1,], b=optimal_b[1], timegrid=timegrid, group='', kernel=kernel)$haz

haz_2 <- est_haz(data=data[data[,grp]==2,], b=optimal_b[2], timegrid=timegrid, group='', kernel=kernel)$haz

haz_3 <- est_haz(data=data[data[,grp]==3,], b=optimal_b[3], timegrid=timegrid, group='', kernel=kernel)$haz

plot(...); lines(..);  lines(...)

return(data.frame(timegrid, haz_1, haz_2, haz_3))

}
```

- optimal_b is c(161, 50, 112)



Result comparisons: Smoothed estimate of the hazard rate and 95% CI

- var_haz(data, b, timegrid, group=c(' ','g'), kernel=c('unfi', 'ep', 'biw'){

```
if(group==' ')  {return(data.frame( timegrid, var. haz) ) }
if(group=='g ')  {return(data.frame( timegrid, var. haz) ) }
}
```

The variance of $\hat{h}(t)$ is estimated by the quantity

$$\sigma^2[\hat{h}(t)] = b^{-2} \sum_{i=1}^{D} K\left(\frac{t - t_i}{b}\right)^2 \Delta \hat{V}[\tilde{H}(t_i)].$$

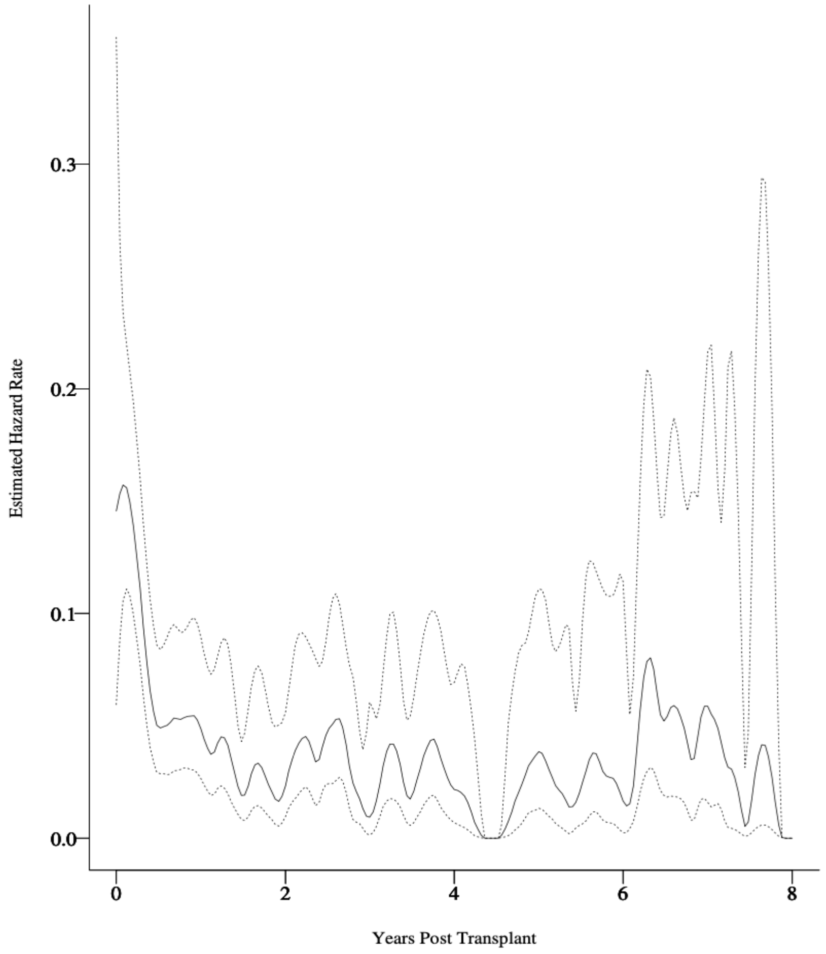


Figure 6.6 *Smoothed estimate of the hazard rate (———) and 95% confidence interval (------) for the time to death following a kidney transplant based on the biweight kernel and the best bandwidth.*

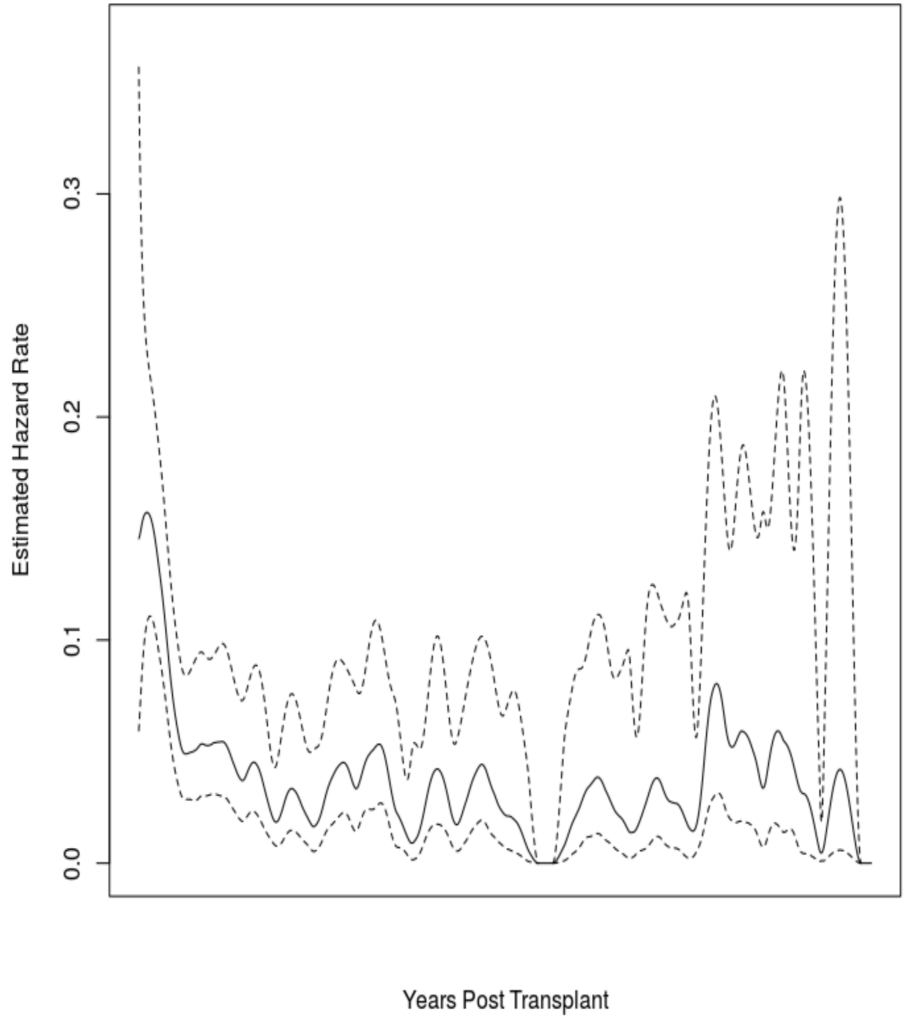
Result comparisons: Smoothed estimate of the hazard rate and 95% CI

- var_haz(data, b, timegrid, group=c(' ','g'), kernel=c('unfi', 'ep', 'biw'){

```
if(group==' ')  {return(data.frame( timegrid, var. haz) ) }
if(group=='g ')  {return(data.frame( timegrid, var. haz) ) }
}
```

The variance of $\hat{h}(t)$ is estimated by the quantity

$$\sigma^2[\hat{h}(t)] = b^{-2} \sum_{i=1}^{D} K\left(\frac{t - t_i}{b}\right)^2 \Delta \hat{V}[\tilde{H}(t_i)].$$



Thank you

Q & A