Solutions: Chapter 4

Exercise 1

Verify equation (4.17).

Proof. The increment at $X_i = T_i \wedge C_i$ is:

$$\begin{aligned}
&1 - \hat{\mathrm{P}}(T > X_i) - (1 - \hat{\mathrm{P}}(T > X_i-)) \\
=&1 - \prod_{X_i \le t} \left(1 - \frac{\Delta N_{0\cdot}(X_i)}{Y_0(X_i)}\right) - 1 + \prod_{X_i < t} \left(1 - \frac{\Delta N_{0\cdot}(X_i)}{Y_0(X_i)}\right) \\
=& \prod_{X_i < t} \left(1 - \frac{\Delta N_{0\cdot}(X_i)}{Y_0(X_i)}\right) \cdot \left(-1 + \frac{\Delta N_{0\cdot}(X_i)}{Y_0(X_i)} + 1\right) \\
=&\hat{\mathrm{P}}(T > X_i-)\frac{\Delta N_{0\cdot}(X_i)}{Y_0(X_i)}
\end{aligned}$$

□

Exercise 2

Show that

$$\hat{\mathrm{P}}(T \le t, X_T = 1) \le 1 - \prod_{T_i \wedge C_i \le t} \left(1 - \frac{\Delta N_{01}(T_i \wedge C_i)}{Y_0(T_i \wedge C_i)}\right)$$

Proof.

$$\begin{aligned}
\hat{\mathrm{P}}(T \le t, X_T = 1) &= \sum_{T_i \wedge C_i \le t} \hat{\mathrm{P}}(T > (T_i \wedge C_i)-) \cdot \frac{\Delta N_{01}(T_i \wedge C_i)}{Y_0(T_i \wedge C_i)} \\
&\le \sum_{T_i \wedge C_i \le t} \hat{\mathrm{P}}(T > (T_i \wedge C_i)-) \cdot \frac{\Delta N_{0\cdot}(T_i \wedge C_i)}{Y_0(T_i \wedge C_i)} \\
&= 1 - \prod_{T_i \wedge C_i \le t} 1 - \left(\frac{\Delta N_{0\cdot}(T_i \wedge C_i)}{Y_0(T_i \wedge C_i)}\right),
\end{aligned}$$

because of Exercise 1. □

Remark

For the following exercises you will need:

```
> require(etm); require(mvna)
> data(fourD)
```

Exercise 3

a) Matrix of logical defining the possible transitions:

```
> tra.fD <- matrix(FALSE, ncol = 3, nrow = 3)
> tra.fD[1, 2:3] <- TRUE
```

Modification of the data set to use `mvna`

```
> to <- with(fourD, ifelse(status == 0, "cens", status))
> fourD.na <- data.frame(id = fourD$id, from = rep(0, length(to)),
+                        to = to, time = fourD$time)
```

Cumulative CSH for the untreated:

```
> na.untreated <- mvna(fourD.na, c("0", "1", "2"), tra.fD, "cens")
```

We need common event times for both outcomes (only if there is an event).

```
> times <- sort(unique(na.untreated$time))[apply(na.untreated$n.event, 3, sum) != 0]
> na.unt <- predict(na.untreated, times = times)
```

A function to get the increments of the Nelson-Aalen estimator
Input:

`x:` A list. Possibly coming from a call to `predict.mvna`, with possible transformation (see Exercise 4)

Output:
A list of `data.frame` (one per CIF) containing:

`dna:` The increments

`time:` time points at which they are calculated

```
> dna <- function(x) {
+     ## if (!inherits(x, "mvna"))
+     ##     stop("'x' must be of class 'mvna'")
+
+     zzz <- lapply(x, function(ll) {
+         aa <- diff(c(0, ll$na))
+         data.frame(dna = aa, time = ll$time)
+     })
```

```
+
+     names(zzz) <- names(x)
+     zzz
+ }
```

Function to compute Kaplan-Meier estimates from the increments of the Nelson-Aalen estimates
Input:

`x:` An object coming from the `dna` function.

`sminus:` Logical. Whether to return S-. Useful for computing the cumulative incidence functions

Output:
A data.frame with two or three columns:

`time:` time points

`S:` Kaplan-Meier estimates

`Sminus:` Estimates of $S(t-)$

```
> prodint <- function(x, sminus = TRUE) {
+
+     if (!is.list(x))
+         stop("Might not be the good input")
+
+     dna.tot <- apply(do.call(cbind, lapply(x, "[", "dna")), 1, sum)
+     S <- cumprod(1 - dna.tot)
+     zzz <- data.frame(time = x[[1]]$time, S = S)
+     if (sminus) {
+         zzz$Sminus <- c(1, S[-length(S)])
+     }
+     zzz
+ }
```

Compute Kaplan-Meier estimate:

```
> km.unt <- prodint(dna(na.unt), sminus = FALSE)
```

Kaplan-Meier estimate of the censoring distribution `km.cens`:

```
> cens.times <- unique(fourD.na$time[fourD.na$to == "cens"])
> km.cens <- c(1, summary(survfit(Surv(time, to == "cens") ~ 1,
+                                 data = fourD.na))$surv)
```

b) Function for generating a data set **Input**:

`n:` sample size

`km`: Kaplan-meier estimates (coming from the `prodint` function)

`times`: event times

`dna1` **and** `dna2`: Increments of the Nelson-Aalen estimates (coming from the `dna` function)

`km.cens`: Kalan-meier estimates of the censoring distribution

`cens.times`: censoring times

`a`: a logical. Whether the Kaplan-Meier estimator spends 100% of the probability mass. TRUE if not.

Output: A data frame (`time, from, to`).

```
> gen.data <- function(n, km, times, dna1, dna2,
+                      km.cens, cens.times, a) {
+
+     ## Create simulated times
+     times <- sort(times)
+     km <- c(1, km)
+     wp <- -diff(km)
+     wp <- if (a) c(wp, km[length(km)]) else wp
+     T <- sample(times, n, replace = TRUE, prob = wp)
+
+     ## Event times
+     tmp <- findInterval(T, times)
+     probs <- dna1[tmp] / (dna1[tmp] + dna2[tmp])
+     probs[is.na(probs)] <- 0
+     ev <- rbinom(n, size = 1, prob = probs)
+     ev <- ifelse(ev == 0, 2, 1)
+
+     ## censoring
+     cens.times <- sort(cens.times)
+     wp.cens <- -diff(km.cens)
+     wp.cens <- if (!a) c(wp.cens, km.cens[length(km.cens)]) else wp.cens
+     T.cens <- sample(cens.times, n, replace = TRUE, prob = wp.cens)
+
+     ## final step
+     time <- pmin(T.cens, T)
+     to <- ifelse(time == T.cens, "cens", ev)
+
+     data.frame(id = 1:n, time = time, from = rep(0, n), to = to)
+ }
```

Now, we generate data for the placebo group.

```
> epsilon = 0.01
> dna.unt <- dna(na.unt)
> set.seed(1345534)
> data.untreated <- gen.data(500, km.unt$S, c(times, max(times) + epsilon),
+                            dna.unt[[1]]$dna, dna.unt[[2]]$dna,
+                            km.cens, cens.times, TRUE)
```

c) Simulation of a treatment group

```
> na.t <- na.unt
> na.t[["0 1"]]$na <- na.t[["0 1"]]$na * exp(-0.1)
> dna.t <- dna(na.t)
> km.t <- prodint(dna.t, sminus = FALSE)
> set.seed(49668)
> data.treated <- gen.data(500, km.t$S, c(times, max(times) + epsilon),
+                          dna.t[[1]]$dna, dna.t[[2]]$dna,
+                          km.cens, cens.times, TRUE)
```

d) Nelson-Aalen estimates

```
> mvna.unt <- mvna(data.untreated, c("0", "1", "2"), tra.fD, "cens")
> mvna.t <- mvna(data.treated, c("0", "1", "2"), tra.fD, "cens")
> ## Plots of the estimated cumulative hazards
> olpar <- par(mfrow = c(1, 2))
> plot(mvna.t, tr.choice = "0 1", legend = FALSE, col = 2,
+      main = "Interest", xlim = c(0, 6), ylim = c(0, 1.5))
> lines(mvna.unt, tr.choice = "0 1")
> legend("topleft", c("Control", "Treated"), lty = 1, col = c(1, 2),
+        cex = 1.3, bty = "n")
> plot(mvna.t, tr.choice = "0 2", legend = FALSE, col = 2,
+      main = "Competing", xlim = c(0, 6), ylim = c(0, 1.5))
> lines(mvna.unt, tr.choice = "0 2")
> par(olpar)
```

Kaplan-Meier and CIFs using `etm`

```
> etm.unt <- etm(data.untreated, c("0", "1", "2"), tra.fD, "cens", 0)
> etm.t <- etm(data.treated, c("0", "1", "2"), tra.fD, "cens", 0)
```

Plot of the Kaplan-Meier estimator:

```
> plot(etm.unt, tr.choice = "0 0", col = 1,
+      legend = FALSE)
> lines(etm.t, tr.choice = "0 0", col = 2)
> legend("topright", c("Control", "Treated"), lty = 1, col = c(1, 2),
+        cex = 1.3, bty = "n")
```

Plot of the CIFs:

```
> olpar <- par(mfrow = c(1, 2))
> plot(etm.t, tr.choice = "0 1", legend = FALSE, col = 2,
+      main = "Interest", xlim = c(0, 6), ylim = c(0, 0.6))
> lines(etm.unt, tr.choice = "0 1")
> legend("topleft", c("Control", "Treated"), lty = 1, col = c(1, 2),
+        cex = 1.3, bty = "n")
> plot(etm.t, tr.choice = "0 2", legend = FALSE, col = 2,
+      main = "Competing", xlim = c(0, 6), ylim = c(0, 0.6))
> lines(etm.unt, tr.choice = "0 2")
> par(olpar)
```

Exercise 4

Scenario 1

Untreated:

```
> tna.unt <- na.unt
> dna.unt <- dna(tna.unt)
> km.unt <- prodint(dna.unt)
> data.exo4.untreated <- gen.data(500, km.unt$S, c(times, max(times) + epsilon),
+                            dna.unt[[1]]$dna, dna.unt[[2]]$dna,
+                            km.cens, cens.times, TRUE)
```

Treated:

```
> tna.t <- na.unt
> tna.t[["0 1"]]$na <- tna.t[["0 1"]]$na * exp(-0.3)
> dna.t <- dna(tna.t)
> km.t <- prodint(dna.t, sminus = FALSE)
> data.exo4.treated <- gen.data(500, km.t$S, c(times, max(times) + epsilon),
+                           dna.t[[1]]$dna, dna.t[[2]]$dna,
+                           km.cens, cens.times, TRUE)
```

Plots of the CIF:

```
> etm.unt <- etm(data.exo4.untreated, c("0", "1", "2"), tra.fD, "cens", 0)
> etm.t <- etm(data.exo4.treated, c("0", "1", "2"), tra.fD, "cens", 0)
> olpar <- par(mfrow = c(1, 2))
> plot(etm.t, tr.choice = "0 1", legend = FALSE, col = 2,
+      main = "Interest", xlim = c(0, 6), ylim = c(0, 0.6))
> lines(etm.unt, tr.choice = "0 1")
> legend("topleft", c("Control", "Treated"), lty = 1, col = c(1, 2),
+        cex = 1.3, bty = "n")
> plot(etm.t, tr.choice = "0 2", legend = FALSE, col = 2,
+      main = "Competing", xlim = c(0, 6), ylim = c(0, 0.6))
> lines(etm.unt, tr.choice = "0 2")
> par(olpar)
```

Scenario 2

Untreated:

```
> tna.unt <- na.unt
> tna.unt[["0 1"]]$na <- tna.unt[["0 1"]]$na^(1/4)
> dna.unt <- dna(tna.unt)
> km.unt <- prodint(dna.unt)
> data.exo4.untreated <- gen.data(500, km.unt$S, c(times, max(times) + epsilon),
+                           dna.unt[[1]]$dna, dna.unt[[2]]$dna,
+                           km.cens, cens.times, TRUE)
```

Treated:

```
> tna.t <- tna.unt
> tna.t[["0 1"]]$na <- tna.t[["0 1"]]$na * exp(-0.3)
> dna.t <- dna(tna.t)
> km.t <- prodint(dna.t, sminus = FALSE)
> data.exo4.treated <- gen.data(500, km.t$S, c(times, max(times) + epsilon),
+                           dna.t[[1]]$dna, dna.t[[2]]$dna,
+                           km.cens, cens.times, TRUE)
```

Plots of the CIF:

```
> etm.unt <- etm(data.exo4.untreated, c("0", "1", "2"), tra.fD, "cens", 0)
> etm.t <- etm(data.exo4.treated, c("0", "1", "2"), tra.fD, "cens", 0)
> olpar <- par(mfrow = c(1, 2))
> plot(etm.t, tr.choice = "0 1", legend = FALSE, col = 2,
+      main = "Interest", xlim = c(0, 6), ylim = c(0, 0.6))
> lines(etm.unt, tr.choice = "0 1")
> legend("topleft", c("Control", "Treated"), lty = 1, col = c(1, 2),
+        cex = 1.3, bty = "n")
> plot(etm.t, tr.choice = "0 2", legend = FALSE, col = 2,
+      main = "Competing", xlim = c(0, 6), ylim = c(0, 0.6))
> lines(etm.unt, tr.choice = "0 2")
> par(olpar)
```

Scenario 3

Untreated:

```
> tna.unt <- na.unt
> dna.unt <- dna(tna.unt)
> km.unt <- prodint(dna.unt)
> data.exo4.untreated <- gen.data(500, km.unt$S, c(times, max(times) + epsilon),
+                           dna.unt[[1]]$dna, dna.unt[[2]]$dna,
+                           km.cens, cens.times, TRUE)
```

Treated:

```
> tna.t <- tna.unt
> tna.t[["0 1"]]$na <- tna.t[["0 1"]]$na * exp(-0.3)
> tna.t[["0 2"]]$na <- tna.t[["0 2"]]$na * exp(0.3)
> dna.t <- dna(tna.t)
> km.t <- prodint(dna.t, sminus = FALSE)
> data.exo4.treated <- gen.data(500, km.t$S, c(times, max(times) + epsilon),
+                           dna.t[[1]]$dna, dna.t[[2]]$dna,
+                           km.cens, cens.times, TRUE)
```

Plots of the CIF:

```
> etm.unt <- etm(data.exo4.untreated, c("0", "1", "2"), tra.fD, "cens", 0)
> etm.t <- etm(data.exo4.treated, c("0", "1", "2"), tra.fD, "cens", 0)
> olpar <- par(mfrow = c(1, 2))
> plot(etm.t, tr.choice = "0 1", legend = FALSE, col = 2,
+      main = "Interest", xlim = c(0, 6), ylim = c(0, 0.6))
> lines(etm.unt, tr.choice = "0 1")
> legend("topleft", c("Control", "Treated"), lty = 1, col = c(1, 2),
+        cex = 1.3, bty = "n")
> plot(etm.t, tr.choice = "0 2", legend = FALSE, col = 2,
+      main = "Competing", xlim = c(0, 6), ylim = c(0, 0.6))
> lines(etm.unt, tr.choice = "0 2")
> par(olpar)
```

Scenario 4

Untreated:

```
> tna.unt <- na.unt
> tna.unt[["0 2"]]$na <- tna.unt[["0 2"]]$na^2
> dna.unt <- dna(tna.unt)
> km.unt <- prodint(dna.unt)
> data.exo4.untreated <- gen.data(500, km.unt$S, c(times, max(times) + epsilon),
+                             dna.unt[[1]]$dna, dna.unt[[2]]$dna,
+                             km.cens, cens.times, TRUE)
```

Treated:

```
> tna.t <- tna.unt
> tna.t[["0 1"]]$na <- tna.t[["0 1"]]$na * exp(-0.3)
> tna.t[["0 2"]]$na <- tna.t[["0 2"]]$na * exp(0.3)
> dna.t <- dna(tna.t)
> km.t <- prodint(dna.t, sminus = FALSE)
> data.exo4.treated <- gen.data(500, km.t$S, c(times, max(times) + epsilon),
+                           dna.t[[1]]$dna, dna.t[[2]]$dna,
+                           km.cens, cens.times, TRUE)
```

Plots of the CIF:

```
> etm.unt <- etm(data.exo4.untreated, c("0", "1", "2"), tra.fD, "cens", 0)
> etm.t <- etm(data.exo4.treated, c("0", "1", "2"), tra.fD, "cens", 0)
> olpar <- par(mfrow = c(1, 2))
> plot(etm.t, tr.choice = "0 1", legend = FALSE, col = 2,
+      main = "Interest", xlim = c(0, 6), ylim = c(0, 0.6))
> lines(etm.unt, tr.choice = "0 1")
> legend("topleft", c("Control", "Treated"), lty = 1, col = c(1, 2),
+        cex = 1.3, bty = "n")
> plot(etm.t, tr.choice = "0 2", legend = FALSE, col = 2,
+      main = "Competing", xlim = c(0, 6), ylim = c(0, 0.6))
> lines(etm.unt, tr.choice = "0 2")
> par(olpar)
```

Scenario 5

Untreated:

```
> tna.unt <- na.unt
> tna.unt[["0 1"]]$na <- tna.unt[["0 1"]]$na^2
> tna.unt[["0 2"]]$na <- tna.unt[["0 2"]]$na^(1/4)
> dna.unt <- dna(tna.unt)
> km.unt <- prodint(dna.unt)
> data.exo4.untreated <- gen.data(500, km.unt$S, c(times, max(times) + epsilon),
+                            dna.unt[[1]]$dna, dna.unt[[2]]$dna,
+                            km.cens, cens.times, TRUE)
```

Treated:

```
> tna.t <- tna.unt
> tna.t[["0 1"]]$na <- tna.t[["0 1"]]$na * exp(-0.3)
> tna.t[["0 2"]]$na <- tna.t[["0 2"]]$na * exp(-0.3)
> dna.t <- dna(tna.t)
> km.t <- prodint(dna.t, sminus = FALSE)
> data.exo4.treated <- gen.data(500, km.t$S, c(times, max(times) + epsilon),
+                           dna.t[[1]]$dna, dna.t[[2]]$dna,
+                           km.cens, cens.times, TRUE)
```

Plots of the CIF

```
> etm.unt <- etm(data.exo4.untreated, c("0", "1", "2"), tra.fD, "cens", 0)
> etm.t <- etm(data.exo4.treated, c("0", "1", "2"), tra.fD, "cens", 0)
> olpar <- par(mfrow = c(1, 2))
> plot(etm.t, tr.choice = "0 1", legend = FALSE, col = 2,
+      main = "Interest", xlim = c(0, 6), ylim = c(0, 0.6))
> lines(etm.unt, tr.choice = "0 1")
> legend("topleft", c("Control", "Treated"), lty = 1, col = c(1, 2),
+        cex = 1.3, bty = "n")
> plot(etm.t, tr.choice = "0 2", legend = FALSE, col = 2,
```

```
+      main = "Competing", xlim = c(0, 6), ylim = c(0, 0.6))
> lines(etm.unt, tr.choice = "0 2")
> par(olpar)
```

Exercise 5

We now merge the two data set to make things a bit simpler.

```
> dat.chap4.exo5 <- rbind(data.untreated, data.treated)
> dat.chap4.exo5$treatment <- c(rep(0, 500), rep(1, 500))
```

Next, we add left-truncation:

```
> lt.time <- rlnorm(1000, 0.5, 0.3)
> ind.exit <- lt.time < dat.chap4.exo5$time
> dat.chap4.exo5 <- dat.chap4.exo5[ind.exit, ]
> names(dat.chap4.exo5)[names(dat.chap4.exo5) == "time"] <- "exit"
> dat.chap4.exo5$entry <- lt.time[ind.exit]
```

Analysis:

```
> mvna.unt <- mvna(dat.chap4.exo5[dat.chap4.exo5$treatment == 0, ],
+                  c("0", "1", "2"), tra.fD, "cens")
> mvna.t <- mvna(dat.chap4.exo5[dat.chap4.exo5$treatment == 1, ],
+                c("0", "1", "2"), tra.fD, "cens")
> olpar <- par(mfrow = c(1, 2))
> plot(mvna.t, tr.choice = "0 1", legend = FALSE, col = 2,
+      main = "Interest", xlim = c(0, 6), ylim = c(0, 1.5))
> lines(mvna.unt, tr.choice = "0 1")
> legend("topleft", c("Control", "Treated"), lty = 1, col = c(1, 2),
+        cex = 1.3, bty = "n")
> plot(mvna.t, tr.choice = "0 2", legend = FALSE, col = 2,
+      main = "Competing", xlim = c(0, 6), ylim = c(0, 1.5))
> lines(mvna.unt, tr.choice = "0 2")
> par(olpar)
```

Calcuation of Kaplan-Meier and CIFs using `etm`

```
> etm.unt <- etm(dat.chap4.exo5[dat.chap4.exo5$treatment == 0, ],
+                c("0", "1", "2"), tra.fD, "cens", 0)
> etm.t <- etm(dat.chap4.exo5[dat.chap4.exo5$treatment == 1, ],
+              c("0", "1", "2"), tra.fD, "cens", 0)
```

Plot of the Kaplan-Meier estimator:

```
> plot(etm.unt, tr.choice = "0 0", col = 1,
+      legend = FALSE)
> lines(etm.t, tr.choice = "0 0", col = 2)
> legend("topright", c("Control", "Treated"), lty = 1, col = c(1, 2),
+        cex = 1.3, bty = "n")
```

Plot of the CIFs:

```
> olpar <- par(mfrow = c(1, 2))
> plot(etm.t, tr.choice = "0 1", legend = FALSE, col = 2,
+      main = "Interest", xlim = c(0, 6), ylim = c(0, 0.6))
> lines(etm.unt, tr.choice = "0 1")
> legend("topleft", c("Control", "Treated"), lty = 1, col = c(1, 2),
+        cex = 1.3, bty = "n")
> plot(etm.t, tr.choice = "0 2", legend = FALSE, col = 2,
+      main = "Competing", xlim = c(0, 6), ylim = c(0, 0.6))
> lines(etm.unt, tr.choice = "0 2")
> par(olpar)
```

Exercise 6

Consider $T^{(1)}$ the time until spontaneous abortion, $T^{(2)}$ the time until induced abortion and $T^{(3)}$ the time until live birth. $T = T^{(1)} \wedge T^{(2)} \wedge T^{(3)}$ and $X_T = 1 \iff \min(T^{(1)}, T^{(2)}, T^{(3)}) = T^{(1)}$.

Assuming independence between the latent times, $T^{(1)}$ could be interpreted as the time until spontaneous abortion in a world where neither induced abortion nor life birth were possible, which is not of big interest. The same interpretation goes for $T^{(2)}$. Estimating $\mathrm{P}(T^{(1)} > t)$ and $\mathrm{P}(T^{(2)} > t)$, one would find that coumarin increases the risk of spontaneous abortion and induced abortion, but these probabilities are rather useless.

More interesting might be to analyse $T^{(3)}$ the time until live birth in a world where every pregnancy would end in a live birth. One would find that $\mathrm{P}(T^{(3)} > t)$ equals 1 for both groups at the end of the follow-up, but babies whose mothers have been exposed to coumarin derivatives seem to be born earlier.