Solutions: Chapter 5

Exercise 1

The solutions are for the scenario of exercise 3 chapter 4. For the other scenarios, however, the same code can be used.

```
> dat.chap5.exo1 <- rbind(data.untreated, data.treated)
> dat.chap5.exo1$treatment <- c(rep(0, 500), rep(1, 500))
> dat.chap5.exo1$id <- seq_len(nrow(dat.chap5.exo1))
```

First event analysis

```
> cox.first <- coxph(Surv(time, to != "cens") ~ treatment, dat.chap5.exo1)
```

Cause-specific hazard analysis

```
> cox.01 <- coxph(Surv(time, to == 1) ~ treatment, dat.chap5.exo1)
> cox.02 <- coxph(Surv(time, to == 2) ~ treatment, dat.chap5.exo1)
```

Breslow estimates

```
> breslow.01 <- basehaz(cox.01, centered = FALSE)
> breslow.02 <- basehaz(cox.02, centered = FALSE)
```

Prediction using the **mstate** package

```
> require(mstate)
```

Matrix defining the possible transitions

```
> tmat <- trans.comprisk(2)
```

Extend the data frame `dat.chap5.exo1`

```
> dat.ext <- rbind(dat.chap5.exo1, dat.chap5.exo1)
> dat.ext$trans <- c(rep(1, 1000), rep(2, 1000))
> dat.ext$new.status <- as.numeric(c(dat.chap5.exo1$to == 1,
+                                    dat.chap5.exo1$to == 2))
> dat.ext$treat.01 <- dat.ext$treatment * (dat.ext$trans == 1)
> dat.ext$treat.02 <- dat.ext$treatment * (dat.ext$trans == 2)
```

The Cox model (has to use the Breslow method for handling ties)

```
> fit.ms <- coxph(Surv(time, new.status) ~ treat.01 + treat.02 +
+                 strata(trans), dat.ext, method = "breslow")
```

Hypothetical data sets for making prediction, one for control:

```
> newdat.treat0 <- data.frame(treat.01 = c(0, 0), treat.02 = c(0, 0),
+                             strata = c(1, 2))
```

and one for treated:

```
> newdat.treat1 <- data.frame(treat.01 = c(1, 0), treat.02 = c(0, 1),
+                             strata = c(1, 2))
```

Predicted cumulative hazards

```
> msf.treat0 <- msfit(fit.ms, newdat.treat0, trans = tmat)
> msf.treat1 <- msfit(fit.ms, newdat.treat1, trans = tmat)
```

Predicted CIFs

```
> pt.treat0 <- probtrans(msf.treat0, 0)[[1]]
> pt.treat1 <- probtrans(msf.treat1, 0)[[1]]
```

Exercise 2

Prove that $\alpha_{01}(t) = \left(1 + \frac{\mathrm{P}(T \leq t, X_T = 2)}{\mathrm{P}(T > t)}\right)\lambda(t)$.

Proof. Write $F_{0j}(t) = \mathrm{P}(T \leq t, X_T = j)$, $j = 1, 2$, and $\mathrm{S}(t) = \mathrm{P}(T > t)$. Then

$$\begin{aligned}\left(1 + \frac{\mathrm{P}(T \leq t, X_T = 2)}{\mathrm{P}(T > t)}\right)\lambda(t) &= \left(1 + \frac{F_{02}(t)}{1 - F_{01}(t) - F_{02}(t)}\right)\frac{dF_{01}/dt}{1 - F_{01}(t)} \\ &= \frac{dF_{01}(t)/dt}{1 - F_{01}(t) - F_{02}(t)} \\ &= \alpha_{01}(t)\end{aligned}$$

□

Exercise 3

Subdistribution hazard analysis using the **cmprsk** package.

```
> sh.01 <- with(dat.chap5.exo1,
+               crr(time, to, treatment, cencode = "cens", failcode = 1))
> sh.02 <- with(dat.chap5.exo1,
+               crr(time, to, treatment, cencode = "cens", failcode = 2))
```

Using the **kmi** package, first for the event of interest

```
> set.seed(7720)
> imp.data.chap5.exo3.01 <- kmi(Surv(time, to != "cens") ~ 1,
+                               dat.chap5.exo1, etype = to,
+                               failcode = 1, nimp = 10)
> fit.kmi.01 <- cox.kmi(Surv(time, to == 1) ~ treatment,
+                       imp.data.chap5.exo3.01)
```

then for the competing risk

```
> set.seed(4428)
> imp.data.chap5.exo3.02 <- kmi(Surv(time, to != "cens") ~ 1,
+                               dat.chap5.exo1, etype = to,
+                               failcode = 2, nimp = 10)
> fit.kmi.02 <- cox.kmi(Surv(time, to == 2) ~ treatment,
+                       imp.data.chap5.exo3.02)
```

Exercise 4

Function to simulate competing risks data with proportional subdistribution hazards
Input:

`n`: number of people in the study

`p`: p (see Fine and Gray (1999) paper)

`gamma`: Regression parameter

`p.Z`: Probability for binomial experiment to decide on the covariate value. Default is 0.5

`cens.param`: vector of 2 elements. Parameters for the uniformly distributed censoring times

Output:
A data frame with the variables time, status (0, 1, 2 for censoring, event of interest and competing risk, respectively) and covariate Z.

```
> simul.fg <- function(n, p, gamma, p.Z = 0.5, cens.param = c(0, 5)) {
+
+     ## Draw the covariate
+     Z <- rbinom(n, 1, p.Z)
+
+     ## Binomial experiment to decide which event happens
+     prob.event1 <- 1 - (1 - p)^exp(gamma * Z)
+     event <- rbinom(n, 1, prob.event1)
+     event <- ifelse(event == 0, 2, event)
+
```

```
+     ## More or less the CDF conditional on event types. See e.g.,
+     ## Fine and Gray paper.
+     ## -u is for numeric inversion to get the event times
+     ## incr is the increment in the loop below
+     cdf1 <- function(t, u, incr) {
+         ((1 - (1 - p * (1 - exp(-t)))^exp(gamma * Z[incr])) /
+          prob.event1[incr]) - u
+     }
+
+     cdf2 <- function(t, u, incr) {
+         (((1 - p)^exp(gamma * Z[incr]) * (1 - exp(-t * exp(gamma * Z[incr])))) /
+          (1 - prob.event1[incr])) - u
+     }
+
+     time.wo.c <- numeric(n)
+     ## Us is the uniform drawings, for inverse transform sampling
+     Us <- runif(n)
+     for (i in seq_len(n)) {
+         time.wo.c[i] <- switch(as.character(event[i]),
+                                "1" = {
+                                    uniroot(cdf1, c(0, 100), tol = 0.0001,
+                                            u = Us[i], incr = i)$root
+                                },
+                                "2" = {
+                                    uniroot(cdf2, c(0, 100), tol = 0.0001,
+                                            u = Us[i], incr = i)$root
+                                })
+     }
+
+     cens <- runif(n, cens.param[1], cens.param[2])
+     time <- pmin(cens, time.wo.c)
+     status <- ifelse(time == cens, 0, event)
+
+     data.frame(time, status, Z)
+ }
```

Simulation of data

```
> set.seed(74426)
> dat.chap5.exo4 <- simul.fg(200, 0.6, 0.3, cens.param = c(0, 3))
```

Cause-specific hazards analysis

```
> csh.01.exo4 <- coxph(Surv(time, status == 1) ~ Z, dat.chap5.exo4)
> csh.02.exo4 <- coxph(Surv(time, status == 2) ~ Z, dat.chap5.exo4)
```

Subdistribution hazards analysis

```
> sh.01.exo4 <- with(dat.chap5.exo4,
+                    crr(time, status, Z, failcode = 1))
> sh.02.exo4 <- with(dat.chap5.exo4,
+                    crr(time, status, Z, failcode = 2))
```

Graphical check of proportionality of the subdistribution hazards for the event of interest First we need the CIFs. We have to transform the data into **etm** format.

```
> to <- with(dat.chap5.exo4, ifelse(status == 0, "cens", status))
> dat.chap5.exo4.etm <- data.frame(id = 1:nrow(dat.chap5.exo4),
+                                  from = rep(0, nrow(dat.chap5.exo4)),
+                                  to = to,
+                                  time = dat.chap5.exo4$time,
+                                  Z = dat.chap5.exo4$Z)
> tra.cp <- matrix(FALSE, ncol = 3, nrow = 3)
> tra.cp[1, 2:3] <- TRUE
> cif.exo4.Z0 <- etm(subset(dat.chap5.exo4.etm, Z == 0), c("0", "1", "2"),
+                    tra.cp, "cens", 0)
> cif.exo4.Z1 <- etm(subset(dat.chap5.exo4.etm, Z == 1), c("0", "1", "2"),
+                    tra.cp, "cens", 0)
```

computation of the subdistribution hazards

```
> times <- sort(c(cif.exo4.Z0$time, cif.exo4.Z1$time))
> ## take care that the event times are the same for both groups
> cif.Z0 <- trprob(cif.exo4.Z0, tr.choice = "0 1", timepoints = times)
> cif.Z1 <- trprob(cif.exo4.Z1, tr.choice = "0 1", timepoints = times)
> sub.haz.Z0 <- cumsum(1 - ((1 -  cif.Z0) /
+                           (1 - c(0, cif.Z0[-length(cif.Z0)]))))
> sub.haz.Z1 <- cumsum(1 - ((1 -  cif.Z1) /
+                           (1 - c(0, cif.Z1[-length(cif.Z1)]))))
```

Plots

```
> plot(sub.haz.Z0, sub.haz.Z1, lwd = 2, type = "s",
+      xlab = expression(hat(Lambda)(t, "Z = 0")),
+      ylab = expression(hat(Lambda)(t, "Z = 1")))
> abline(a = 0, b = exp(sh.01.exo4$coef), col = "darkgray", lwd = 2)
```

Exercise 5

We reuse `dat.chap5.exo1`. Generation of new censoring times:

```
> dat.chap5.exo1$time.new <- with(dat.chap5.exo1,
+                                 ifelse(time <= 3, time, 4))
> dat.chap5.exo1$to.new <- with(dat.chap5.exo1,
+                               ifelse(time <= 3, as.character(to), "cens"))
```

First event analysis

```
> cox.first.new <- coxph(Surv(time.new, to.new != "cens") ~ treatment,
+                        dat.chap5.exo1)
```

Cause-specific hazard analysis

```
> cox.01.new <- coxph(Surv(time.new, to.new == 1) ~ treatment, dat.chap5.exo1)
> cox.02.new <- coxph(Surv(time.new, to.new == 2) ~ treatment, dat.chap5.exo1)
```

Subdistribution hazard analysis

```
> sh.01.new <- with(dat.chap5.exo1,
+                   crr(time.new, to.new, treatment, cencode = "cens",
+                       failcode = 1))
> sh.02.new <- with(dat.chap5.exo1,
+                   crr(time.new, to.new, treatment, cencode = "cens",
+                       failcode = 2))
```

Exercise 6

All-cause-hazard.
Calculation of the Nelson-Aalen estimator via `survfit`

```
> surv.first <- survfit(Surv(time, to != "cens") ~ treatment,
+                       dat.chap5.exo1)
> all.na.Z0 <- cumsum(surv.first[1]$n.event / surv.first[1]$n.risk)
> all.na.Z1 <- cumsum(surv.first[2]$n.event / surv.first[2]$n.risk)
> times <- sort(surv.first$time)
> ind.Z0 <- findInterval(times, surv.first[1]$time)
> ind.Z0[ind.Z0 == 0] <- NA
> all.na.Z0 <- all.na.Z0[ind.Z0]
> ind.Z1 <- findInterval(times, surv.first[2]$time)
> ind.Z1[ind.Z1 == 0] <- NA
> all.na.Z1 <- all.na.Z1[ind.Z1]
```

For the cause-specific hazards (via `mvna`)

```
> mvna.unt <- mvna(data.untreated, c("0", "1", "2"), tra.cp, "cens")
> mvna.t <- mvna(data.treated, c("0", "1", "2"), tra.cp, "cens")
> times <- sort(unique(mvna.unt$time, mvna.t$time))
> na.Z0 <- predict(mvna.unt, times)
> na.Z1 <- predict(mvna.t, times)
```

Plot as Figure 5.14

```
> old.par <- par(no.readonly = TRUE)
> nf <- layout(t(matrix(c(1, 2, 3))), width = c(1, 1, 1))
> ## all-cause-hazard
```

```
> plot(all.na.Z0, all.na.Z1, type = "s", lwd = 2, #xlim = c(0, 6), ylim = c(0, 2),
+      xlab = expression(hat(A)["0.;0"](t)),
+      ylab = expression(hat(A)["0."](t, "Z = 1")),
+      main = "All-Cause Hazard")
> abline(a = 0, b = exp(cox.first$coef), col = "darkgray", lwd = 2)
> ## csh 01
> plot(na.Z0[["0 1"]]$na, na.Z1[["0 1"]]$na, type = "s", lwd = 2,
+      xlab = expression(hat(A)["01;0"](t)),
+      ylab = expression(hat(A)["01"](t, "Z = 1")),
+      main = "Event of interest")
> abline(a = 0, b = exp(cox.01$coef), col = "darkgray", lwd = 2)
> ## csh 02
> plot(na.Z0[["0 2"]]$na, na.Z1[["0 2"]]$na, type = "s", lwd = 2,
+      xlab = expression(hat(A)["01;0"](t)),
+      ylab = expression(hat(A)["01"](t, "Z = 1")),
+      main = "Competing Risk")
> abline(a = 0, b = exp(cox.01$coef), col = "darkgray", lwd = 2)
```

Subdistribution hazards.
Use the CIFs

```
> etm.unt <- etm(data.untreated, c("0", "1", "2"), tra.cp, "cens", 0)
> etm.t <- etm(data.treated, c("0", "1", "2"), tra.cp, "cens", 0)
> times <- sort(c(etm.unt$time, etm.t$time))
> cif.Z0 <- trprob(etm.unt, tr.choice = "0 1", timepoints = times)
> cif.Z1 <- trprob(etm.t, tr.choice = "0 1", timepoints = times)
> sub.haz.Z0 <- cumsum(1 - ((1 -  cif.Z0) /
+                           (1 - c(0, cif.Z0[-length(cif.Z0)]))))
> sub.haz.Z1 <- cumsum(1 - ((1 -  cif.Z1) /
+                           (1 - c(0, cif.Z1[-length(cif.Z1)]))))
```

Plot

```
> plot(sub.haz.Z0, sub.haz.Z1, lwd = 2, type = "s",
+      xlab = expression(hat(Lambda)(t, "Z = 0")),
+      ylab = expression(hat(Lambda)(t, "Z = 1")))
> abline(a = 0, b = exp(sh.01$coef), col = "darkgray", lwd = 2)
```

Exercise 7

Load the **compeir** package and the data set.

```
> require(compeir)
> data(okiss)
```

We group the competing risks “end of neutropenia” and “death” (see rational in Section 5.2.2)

```
> okiss$status <- with(okiss, ifelse(status %in% c(2, 7), 2, status))
> okiss$time <- unclass(okiss$time)
```

First, some graphical analysis:
Plot of the cause-specific hazards

```
> okiss$from <- rep(0, nrow(okiss))
> okiss$to <- okiss$status
> okiss$id <- seq_len(nrow(okiss))
> mvna.auto <- mvna(subset(okiss, allo == 0), c("0", "1", "2"),
+                   tra.cp, "11")
> mvna.allo <- mvna(subset(okiss, allo == 1), c("0", "1", "2"),
+                   tra.cp, "11")
> par(mfrow = c(1, 2))
> plot(mvna.auto, tr.choice = "0 1", legend = FALSE,
+             main = "Infection", ylim = c(0, 2))
> lines(mvna.allo, tr.choice = "0 1", col = 2)
> legend("topleft", c("Autologous", "Allogeneic"),
+        lty = 1, col = c(1, 2), bty = "n")
> ##
> plot(mvna.auto, tr.choice = "0 2", legend = FALSE,
+      main = "End of Neutropenia", ylim = c(0, 2))
> lines(mvna.allo, tr.choice = "0 2", col = 2)
```

Cumulative incidence functions

```
> etm.auto <- etm(subset(okiss, allo == 0), c("0", "1", "2"),
+                   tra.cp, "11", 0)
> etm.allo <- etm(subset(okiss, allo == 1), c("0", "1", "2"),
+                   tra.cp, "11", 0)
> par(mfrow = c(1, 2))
> plot(etm.auto, tr.choice = "0 1", legend = FALSE,
+             main = "Infection", ylim = c(0, 1))
> lines(etm.allo, tr.choice = "0 1", col = 2)
> legend("topleft", c("Autologous", "Allogeneic"),
+        lty = 1, col = c(1, 2), bty = "n")
> ##
> plot(etm.auto, tr.choice = "0 2", legend = FALSE,
+      main = "End of Neutropenia", ylim = c(0, 1))
> lines(etm.allo, tr.choice = "0 2", col = 2)
```

Cox model for the all-cause hazards

```
> cox.okiss.all <- coxph(Surv(time, status != 11) ~ allo, okiss)
> summary(cox.okiss.all)

Call:
coxph(formula = Surv(time, status != 11) ~ allo, data = okiss)
```

```
  n= 1000, number of events= 987

         coef exp(coef) se(coef)      z Pr(>|z|)
allo -1.10501   0.33121  0.06985 -15.82   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

     exp(coef) exp(-coef) lower .95 upper .95
allo    0.3312      3.019    0.2888    0.3798

Concordance= 0.64  (se = 0.009 )
Rsquare= 0.21   (max possible= 1 )
Likelihood ratio test= 236.3  on 1 df,   p=0
Wald test            = 250.3  on 1 df,   p=0
Score (logrank) test = 268.4  on 1 df,   p=0
```

Cause-specific analysis

```
> cox.okiss.01 <- coxph(Surv(time, status == 1) ~ allo, okiss)
> cox.okiss.02 <- coxph(Surv(time, status == 2) ~ allo, okiss)
> summary(cox.okiss.01)

Call:
coxph(formula = Surv(time, status == 1) ~ allo, data = okiss)

  n= 1000, number of events= 203

        coef exp(coef) se(coef)      z Pr(>|z|)
allo -0.2599    0.7711   0.1485 -1.751     0.08 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

     exp(coef) exp(-coef) lower .95 upper .95
allo    0.7711      1.297    0.5765     1.032

Concordance= 0.535  (se = 0.019 )
Rsquare= 0.003   (max possible= 0.929 )
Likelihood ratio test= 3.03  on 1 df,   p=0.08199
Wald test            = 3.06  on 1 df,   p=0.08003
Score (logrank) test = 3.08  on 1 df,   p=0.07928

> summary(cox.okiss.02)

Call:
coxph(formula = Surv(time, status == 2) ~ allo, data = okiss)
```

```
  n= 1000, number of events= 784

         coef exp(coef) se(coef)      z Pr(>|z|)
allo -1.32554   0.26566  0.07774 -17.05   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

     exp(coef) exp(-coef) lower .95 upper .95
allo    0.2657      3.764    0.2281    0.3094

Concordance= 0.683  (se = 0.01 )
Rsquare= 0.236   (max possible= 1 )
Likelihood ratio test= 269  on 1 df,   p=0
Wald test            = 290.8  on 1 df,   p=0
Score (logrank) test = 321.8  on 1 df,   p=0
```

Subdistribution hazards

```
> fg.okiss.01 <- with(okiss, crr(time, status, allo,
+                                failcode = 1, cencode = 11))
> fg.okiss.02 <- with(okiss, crr(time, status, allo,
+                                failcode = 2, cencode = 11))
> summary(fg.okiss.01)

Competing Risks Regression

Call:
crr(ftime = time, fstatus = status, cov1 = allo, failcode = 1,
    cencode = 11)

       coef exp(coef) se(coef)     z p-value
allo1 0.087      1.09    0.142 0.613    0.54

      exp(coef) exp(-coef)  2.5% 97.5%
allo1      1.09      0.917 0.826  1.44

Num. cases = 1000
Pseudo Log-likelihood = -1381
Pseudo likelihood ratio test = 0.37  on 1 df,

> summary(fg.okiss.02)

Competing Risks Regression

Call:
crr(ftime = time, fstatus = status, cov1 = allo, failcode = 2,
    cencode = 11)
```

```
        coef exp(coef) se(coef)     z p-value
allo1 -0.587     0.556   0.0786 -7.47 8.3e-14

      exp(coef) exp(-coef)  2.5% 97.5%
allo1     0.556        1.8 0.477 0.649

Num. cases = 1000
Pseudo Log-likelihood = -4940
Pseudo likelihood ratio test = 62.6  on 1 df,
```