Detailed answers to the computer practicals in the book "Data Analysis with Competing Risks and Intermediate States"

Ronald B. Geskus
Department of Clinical Epidemiology, Biostatistics and Bioinformatics
Academic Medical Center
Meibergdreef 15, 1105 AZ Amsterdam
Email: r.b.geskus@amc.uva.nl

```
R version 3.2.1 (2015-06-18)
Platform: i386-w64-mingw32/i386 (32-bit)
Running under: Windows XP (build 2600) Service Pack 3

locale:
[1] LC_COLLATE=Dutch_Netherlands.1252  LC_CTYPE=Dutch_Netherlands.1252
[3] LC_MONETARY=Dutch_Netherlands.1252 LC_NUMERIC=C
[5] LC_TIME=Dutch_Netherlands.1252

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base

other attached packages:
[1] crrSC_1.1       msSurv_1.2-2    graph_1.46.0    mvna_1.2-3      etm_0.6-2
[6] mstate_0.2.8    survival_2.38-3 knitr_1.10.5

loaded via a namespace (and not attached):
 [1] lattice_0.20-33     class_7.3-13        grid_3.2.1          stats4_3.2.1
 [5] formatR_1.2         magrittr_1.5        evaluate_0.7        highr_0.5
 [9] stringi_0.5-5       splines_3.2.1       RColorBrewer_1.1-2  tools_3.2.1
[13] stringr_1.0.0       parallel_3.2.1      compiler_3.2.1      BiocGenerics_0.14.0
```

Chapter 1. Basic Concepts

The data are part of the `mstate` package. Make sure that you have installed the package `mstate` from one of the CRAN servers (`https://cran.r-project.org`). We load the package and the data that we use in this practical via

```
> library(mstate)
> data(ebmt1)
```

1. A first look at the data

Have a look at the help file of the `ebmt1` data set to get information on the meaning of the variables and their possible values.

Inspect the data by looking at the first couple of individuals or by viewing the complete data set. How many rows and columns does the data set have? Look at the type of the variables (count, continuous or categorical).

For inspection of larger data sets, the functions `head` and `tail` are useful. By default they show the first and last six lines of the data set.

```
> head(ebmt1)
```

```
  patid  srv srvstat  rel relstat      yrel age       score
1     1 1610       0 1610       0      <NA>  28 Medium risk
2     2  961       1  165       1 1997-1999  33 Medium risk
3     3 1508       0 1508       0      <NA>  38 Medium risk
4     4 1376       0 1376       0      <NA>  15 Medium risk
5     5 1585       0  133       1 1997-1999  26 Medium risk
6     6  190       1   63       1 1997-1999  22 Medium risk
```

```
> tail(ebmt1)
```

```
     patid srv srvstat rel relstat      yrel age       score
1972  1972 746       1 746       0      <NA>  47 Medium risk
1973  1973 123       1 122       1 1997-1999  44   High risk
1974  1974 632       1 632       0      <NA>  32    Low risk
1975  1975 911       0 215       1     2000-  39 Medium risk
1976  1976  78       1  78       0      <NA>  42 Medium risk
1977  1977 180       1 180       0      <NA>  19 Medium risk
```

The complete data set is shown via the command `View(ebmt1)`.

The function `dim` is used to find the number of rows (patients in this case) and columns (variables) in the data.

```
> dim(ebmt1)
```

```
[1] 1977    8
```

An alternative is the function `str`. This function shows information per column (type of variable, first records, levels in case of factor variables) as well as the total number of rows and columns.

```
> str(ebmt1)
```

```
'data.frame':        1977 obs. of  8 variables:
 $ patid  : int  1 2 3 4 5 6 7 8 9 10 ...
 $ srv    : num  1610 961 1508 1376 1585 ...
 $ srvstat: int  0 1 0 0 0 1 0 1 0 1 ...
 $ rel    : num  1610 165 1508 1376 133 ...
 $ relstat: int  0 1 0 0 1 1 0 0 1 1 ...
```

```
$ yrel   : Factor w/ 3 levels "1993-1996","1997-1999",..: NA 2 NA NA 2 2 NA NA 2 2 ..
$ age    : int  28 33 38 15 26 22 7 44 38 47 ...
$ score  : Factor w/ 3 levels "Low risk","Medium risk",..: 2 2 2 2 2 2 1 2 2 3 ...
```

Summarize all the variables. Why are there 1521 *NA*'s in the variable yrel? How many relapses were observed and how many deaths?

A quick way to obtain a summary of all variables at once is via the function `summary` applied to the complete data frame.

```
> summary(ebmt1)
```

```
     patid           srv            srvstat           rel            relstat
 Min.   :   1   Min.   :   0.5   Min.   :0.000   Min.   :   0.5   Min.   :0.000
 1st Qu.: 495   1st Qu.: 157.0   1st Qu.:0.000   1st Qu.: 113.0   1st Qu.:0.000
 Median : 989   Median : 692.0   Median :0.000   Median : 395.0   Median :0.000
 Mean   : 989   Mean   : 854.5   Mean   :0.441   Mean   : 701.3   Mean   :0.231
 3rd Qu.:1483   3rd Qu.:1400.0   3rd Qu.:1.000   3rd Qu.:1127.0   3rd Qu.:0.000
 Max.   :1977   Max.   :3088.0   Max.   :1.000   Max.   :3088.0   Max.   :1.000
       yrel           age             score
 1993-1996: 103   Min.   : 0.0   Low risk   : 406
 1997-1999: 208   1st Qu.:28.0   Medium risk:1404
 2000-    : 145   Median :36.0   High risk  : 167
 NA's     :1521   Mean   :35.8
                  3rd Qu.:45.0
                  Max.   :64.0
```

Missing values in R are represented as *NA*. There are 1521 *NA*'s in the variable `yrel` because these individuals did not have a relapse observed.

`survstat` and `relstat` have been interpreted as numeric variables and are therefore summarized via mean and quantiles. To see how many events have occurred, the function `table` can be used.

```
> table(ebmt1$srvstat)
```

```
   0    1
1105  872
```

```
> table(ebmt1$relstat)
```

```
   0    1
1521  456
```

2. Some data preparation

In the `ebmt1` data set, information with respect to relapse and death is given in separate columns. In a competing risks analysis, the usual data presentation is via a single time and status column.

Add two more columns to the `ebmt1` data set. One column, to be called time, contains the time of the first of the two events, relapse and death, or the time of censoring if neither was observed. Use year instead of day as time unit. The second column, to be called stat, is a column that has the value 0 for right censored individuals, 1 if there was a relapse and 2 if the individual died without having had a relapse. You can add a third column that explains the meaning of 0, 1 and 2.

The function `pmin` computes the minimum value of two of more columns of numeric variables. It is a vectorized function, meaning that the minimum per row is computed. Since death never occurs before relapse, we can see which individuals are censored from the `srvstat` column. There are many ways to add the columns, but one efficient way is the following

```
> ebmt1 <- within(ebmt1,{
+   time <- pmin(srv,rel)/365.25
+   stat <-  ifelse(rel<srv, relstat, srvstat*2)
+   type <- factor(stat, labels=c("Event free","Relapse","Death"))
+   })
```

3. Estimation of overall cumulative incidence

Compute the Kaplan-Meier estimator for relapse-free survival. Relapse-free survival means that the individual neither had a relapse nor died, i.e. both end points are combined as event. Obtain the estimates of relapse-free survival at one and five years. Plot the estimate of the cumulative incidence (the complement of the Kaplan-Meier survival curve).

The Kaplan-Meier is obtained via the `survfit` function.

```
> KM.overall <- survfit(Surv(time,stat>0)~1, data=ebmt1)
```

The result is a so-called `survfit` object, for which a number of functions ("methods" is the better word) are defined. If you just give the name of the object as command, the `print` method is invoked. It gives a very basic summary of the estimated curve.

```
> KM.overall

Call: survfit(formula = Surv(time, stat > 0) ~ 1, data = ebmt1)

      n  events  median 0.95LCL 0.95UCL
1977.00 1141.00    1.36    1.11    1.79
```

The `summary.survfit` function gives the complete information: the event times, the Kaplan-Meier estimate at those time points, and also other useful things like the number at risk, number of events etcetera. It has an argument *times* that we can use to obtain relapse-free survival at specific time points only.

```
> summary(KM.overall, times=c(1,5))

Call: survfit(formula = Surv(time, stat > 0) ~ 1, data = ebmt1)

 time n.risk n.event survival std.err lower 95% CI upper 95% CI
    1   1012     897    0.538  0.0113        0.516        0.560
    5    198     233    0.378  0.0125        0.355        0.404
```

A `survfit` object also has a `plot` method. By default it plots the curve as survival function, i.e. starting at the value 1 and going down. A plot on the complementary scale of the cumulative incidence is specified via the *fun* argument. By default, all censoring times are shown via small vertical lines. They are left out by setting the argument *mark.time* to *FALSE*. The result is shown in Figure 1. We use `par(las=1)` to plot the labels along the y-axis horizontally.

```
>   par(las=1)
>   plot(KM.overall, fun="event", mark.time=FALSE,
+        xlab="time since transplant (years)", ylab="cumulative incidence")
```

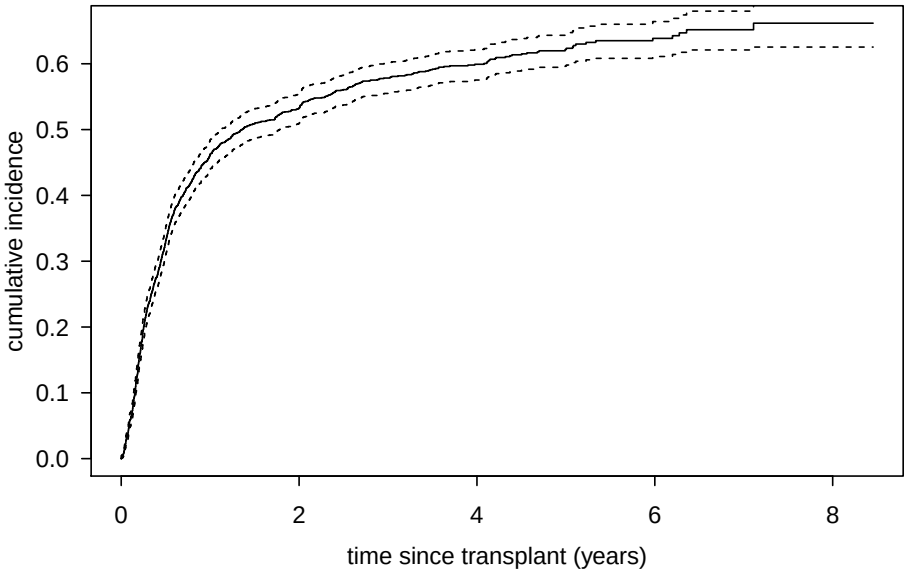

Figure 1: The Kaplan-Meier curve for relapse-free survival, on the scale of the cumulative incidence

4. A Cox model for relapse-free survival

Fit a Cox model for the effect of the EBMT score on relapse-free survival. Locate the hazard ratios. What is your conclusion with respect to the effect of the EBMT score?

Proportional hazards models are fitted with the function `coxph`. Just like with the `survfit` function, the `print.coxph` command gives a short summary of the model fit, whereas the `summary.coxph` function gives more details. Again, if only the name of the object is specified, the print function is assumed.

```
> PH.relsurv <- coxph(Surv(time, stat>0) ~ score, data = ebmt1)
> PH.relsurv

Call:
coxph(formula = Surv(time, stat > 0) ~ score, data = ebmt1)


                  coef exp(coef) se(coef)    z       p
scoreMedium risk 0.550     1.733    0.084 6.55 5.9e-11
scoreHigh risk   1.133     3.105    0.117 9.67 < 2e-16

Likelihood ratio test=94.4  on 2 df, p=0
n= 1977, number of events= 1141

> summary(PH.relsurv)

Call:
coxph(formula = Surv(time, stat > 0) ~ score, data = ebmt1)

  n= 1977, number of events= 1141
```

```
                   coef exp(coef) se(coef)    z Pr(>|z|)
scoreMedium risk 0.550     1.733    0.084 6.55  5.9e-11 ***
scoreHigh risk   1.133     3.105    0.117 9.67  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

                  exp(coef) exp(-coef) lower .95 upper .95
scoreMedium risk      1.73      0.577      1.47      2.04
scoreHigh risk        3.10      0.322      2.47      3.91

Concordance= 0.569  (se = 0.007 )
Rsquare= 0.047   (max possible= 1 )
Likelihood ratio test= 94.4  on 2 df,   p=0
Wald test            = 94.4  on 2 df,   p=0
Score (logrank) test = 98.9  on 2 df,   p=0
```

The hazard ratios are in the column `exp(coef)`. We see that those with medium and high value of the EBMT score have a significantly higher rate of losing the event-free survival status.

Test whether the effect of the EBMT score is non-proportional.

We test for non-proportionality based on Schoenfeld residuals using the `cox.zph` function. The test for non-proportionality of the EBMT score is not significant.

```
> cox.zph(PH.relsurv)

                     rho chisq     p
scoreMedium risk -0.0436  2.15 0.142
scoreHigh risk   -0.0342  1.32 0.250
GLOBAL                NA  2.27 0.322
```

We obtain an impression of the time trend by plotting a smooth curve through the Schoenfeld residuals. We leave out the residuals from the plot in Figure 2 via *resid=FALSE*. `par(mfrow=c(1,2))` creates space for two plots next to each other. The plot suggests that there may be some non-proportionality after two years, but before that time the curves lie around the coefficients 0.55 and 1.133.

5. Log-rank test

Use the log-rank test to investigate whether the EBMT score influences relapse-free survival. Make a plot of the cumulative hazard per level of the EBMT score.

The log-rank test is performed via the `survdiff` function.

```
> survdiff(Surv(time, stat>0) ~ score, data = ebmt1)

Call:
survdiff(formula = Surv(time, stat > 0) ~ score, data = ebmt1)

                     N Observed Expected (O-E)^2/E (O-E)^2/V
score=Low risk     406      171    278.6     41.57     55.36
score=Medium risk 1404      841    793.9      2.79      9.21
score=High risk    167      129     68.5     53.50     57.28

 Chisq= 98.8  on 2 degrees of freedom, p= 0
```

We again see that the effect of the EBMT risk score is highly significant. The effect is visualized in Figure 3. Note that, if the hazards are proportional, the curves should differ by a constant on the multiplicative scale ($h_M = \kappa \times h_L$ etcetera).

```
>    par(mfrow=c(1,2))
>    plot(cox.zph(PH.relsurv), resid=FALSE)
```

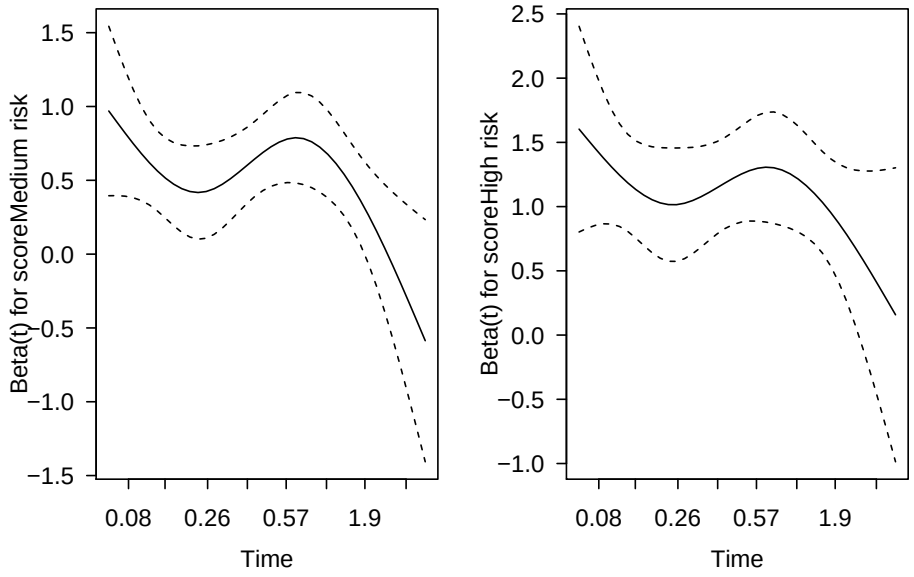

Figure 2: Smooth trend though the Schoenfeld residuals.

```
>    par(mfrow=c(1,1))
>    plot(survfit( Surv(time, stat>0) ~ score, data = ebmt1), fun="cumhaz",
+                                  mark.time=FALSE,conf.int=TRUE, col=1:3,
+         xlab="time since transplant (years)", ylab="cumulative hazard")
>    legend("bottomright",legend=levels(ebmt1$score),col=1:3,lty=1)
```

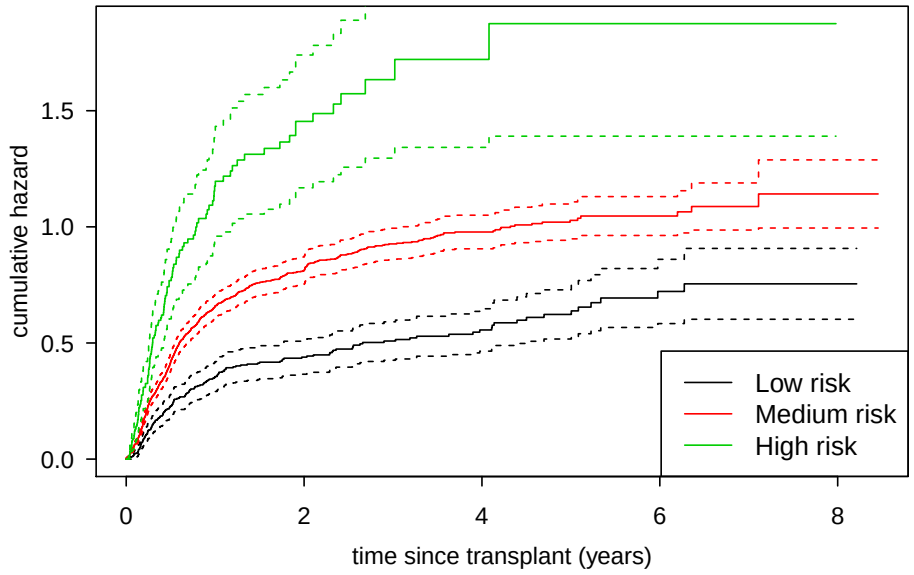

Figure 3: Cumulative hazard by value of the EBMT risk score, with 95% confidence intervals.

Chapter 2. Competing Risks; Nonparametric Estimation

1. Estimation

> **Estimate the cause-specific cumulative incidence for relapse and relapse-free mortality. Save the results in an R object. What is the probability, with 95% confidence intervals on one of the log scales, to have a relapse within one year and within five years. Try approach (c) based on the weighted product-limit form and at least one of the two approaches that use the Aalen-Johansen form, (a) or (b).**

(a) Using the standard code in the `survival` package.
The cumulative incidence for both end points is estimated via

```
> cuminc.1 <- survfit(Surv(time,stat,type="mstate")~1, ebmt1)
```

In order to obtain the values at one and five years, we can use the `summary.survfitms` function:

```
> summ.1 <- summary(cuminc.1, times=c(1,5))
> summ.1

Call: survfit(formula = Surv(time, stat, type = "mstate") ~ 1, data = ebmt1)

 time n.risk n.event prevalence1 prevalence2
    1   1012     897       0.156       0.306
    5    198     233       0.258       0.363
```

Confidence intervals are not shown when printing the output. They have been calculated and can be obtained via

```
> summ.1$lower

       [,1]   [,2]
[1,] 0.1399 0.2852
[2,] 0.2362 0.3405

> summ.1$upper

       [,1]   [,2]
[1,] 0.1723 0.3262
[2,] 0.2798 0.3853
```

The first column is for the first event type (relapse, the value *1* in the `stat` column), the second for death. By default, they are calculated on the "log" scale (formula (1.20) on page 35). We can change the scale via the *`conf.type`* argument in the `survfit` function.

(b) Using the `etm` package.
The cumulative incidence for both end points is estimated via

```
> library(etm)
> cuminc.2 <- etmCIF(Surv(time,stat!=0)~1, ebmt1, etype=stat, failcode=c(1,2))
```

`cuminc.2` is an object of class `etmCIF`. In order to obtain the estimates, we use the `summary.etmCIF` function. Since it does not have a *`times`* argument, we need some extra code to automatically select the latest time point before 1 and 5 years. By default, confidence intervals are given on the "log-log" scale. We can change the scale via the *`ci.fun`* argument in `summary.etmCIF`.

```
> summ.etm <- summary(cuminc.2, ci.fun="log")
> summ.etm.1 <- summ.etm[[1]][["CIF 1"]]
> summ.etm.2 <- summ.etm[[1]][["CIF 2"]]
> select1and5 <- c(sum(summ.etm.1$time<=1),sum(summ.etm.1$time<=5))
> summ.etm.1[select1and5,]
```

```
                        P   time       var  lower  upper n.risk n.event
0.999315537303217 0.1562 0.9993 6.826e-05 0.1409 0.1733   1014       2
4.99383983572895  0.2583 4.9938 1.237e-04 0.2374 0.2811    199       0
```

```
> summ.etm.2[select1and5,]
```

```
                        P   time       var  lower  upper n.risk n.event
0.999315537303217 0.3060 0.9993 0.0001095 0.2862 0.3272   1014       0
4.99383983572895  0.3633 4.9938 0.0001310 0.3416 0.3864    199       0
```

The estimates are the same as in approach 1.(a). The confidence intervals are slightly different, because `etm` uses the formula in (1.23) on page 36 of the book for the "log", which is different from the "log" scale based on (1.20) which is used by the `survfit` function in (a).

Another option is to evaluate the `trprob.etm` and `trcov.etm` functions with as first argument `cuminc.2[[1]]`.

```
> est.01 <- trprob(cuminc.2[[1]], tr.choice="0 1", timepoints=c(1,5))
> est.02 <- trprob(cuminc.2[[1]], tr.choice="0 2", timepoints=c(1,5))
> est.01
```

```
[1] 0.1562 0.2583
```

```
> est.02
```

```
[1] 0.3060 0.3633
```

We have to compute the confidence intervals ourselves, based on the estimated standard error. The same result on the "log" scale is obtained via

```
> CI.01 <- trcov(cuminc.2[[1]], tr.choice="0 1", timepoints=c(1,5))
> CI.02 <- trcov(cuminc.2[[1]], tr.choice="0 2", timepoints=c(1,5))
> est.01*exp(-qnorm(0.975)*sqrt(CI.01)/est.01)
```

```
[1] 0.1409 0.2374
```

```
> est.01*exp(qnorm(0.975)*sqrt(CI.01)/est.01)
```

```
[1] 0.1733 0.2811
```

```
> est.02*exp(-qnorm(0.975)*sqrt(CI.02)/est.02)
```

```
[1] 0.2862 0.3416
```

```
> est.02*exp(qnorm(0.975)*sqrt(CI.02)/est.02)
```

```
[1] 0.3272 0.3864
```

(c) Using the weighted product-limit form.

> **Create the weighted data set via the `crprep` function, and store it as an object named `Webmt1`. Include the covariables score and age and also store the type column in the new data set. Compute the weights for both end points.**
>
> **Inspect the first ten rows of the weighted data set that contain information for each of the two end points. What do the newly created variables represent? How many rows does the newly created object have?**

We use the specification via the column names:

```
> Webmt1 <- crprep(Tstop="time", status="stat", data=ebmt1, trans=1:2,
+                       cens=0, id="patid", keep=c("score","age","type"))
```

Via *trans=1:2* we compute the weights for both event types at once.

Here are two alternative codings to obtain the same result:

```
> Webmt1a <- with(ebmt1, crprep(Tstop=time, status=stat, trans=1:2, cens=0,
+                               id=patid, keep=ebmt1[,c("score","age","type")]))
> Webmt2 <- crprep(Tstop="time", status="type", data=ebmt1, id="patid",
+                           trans=c("Relapse","Death"), cens="Event free",
+                                                keep=c("score","age","type"))
```

`Webmt1a` is completely similar to `Webmt1`, whereas `Webmt2` stores the detailed event type information instead of the numeric coding.

We inspect the first ten rows for the two competing end points:

```
> head(Webmt1,10)
```

```
   patid Tstart  Tstop status weight.cens       score age       type count failcode
1      1 0.0000 4.4079      0      1.0000 Medium risk  28 Event free     1        1
2      2 0.0000 0.4517      1      1.0000 Medium risk  33    Relapse     1        1
3      3 0.0000 4.1287      0      1.0000 Medium risk  38 Event free     1        1
4      4 0.0000 3.7673      0      1.0000 Medium risk  15 Event free     1        1
5      5 0.0000 0.3641      1      1.0000 Medium risk  26    Relapse     1        1
6      6 0.0000 0.1725      1      1.0000 Medium risk  22    Relapse     1        1
7      7 0.0000 2.3491      0      1.0000    Low risk   7 Event free     1        1
8      8 0.0000 0.1889      2      1.0000 Medium risk  44      Death     1        1
9      8 0.1889 0.1944      2      1.0000 Medium risk  44      Death     2        1
10     8 0.1944 0.2081      2      0.9994 Medium risk  44      Death     3        1
```

```
> head(subset(Webmt1, failcode==2)[1:10,],10)
```

```
      patid Tstart  Tstop status weight.cens       score age       type count
94285     1 0.0000 4.4079      0      1.0000 Medium risk  28 Event free     1
94286     2 0.0000 0.4517      1      1.0000 Medium risk  33    Relapse     1
94287     2 0.4517 0.4983      1      1.0000 Medium risk  33    Relapse     2
94288     2 0.4983 0.5038      1      0.9992 Medium risk  33    Relapse     3
94289     2 0.5038 0.5229      1      0.9984 Medium risk  33    Relapse     4
94290     2 0.5229 0.5503      1      0.9977 Medium risk  33    Relapse     5
94291     2 0.5503 0.5558      1      0.9960 Medium risk  33    Relapse     6
94292     2 0.5558 0.5722      1      0.9952 Medium risk  33    Relapse     7
94293     2 0.5722 0.5886      1      0.9944 Medium risk  33    Relapse     8
94294     2 0.5886 0.5941      1      0.9936 Medium risk  33    Relapse     9
      failcode
94285        2
94286        2
94287        2
94288        2
94289        2
94290        2
94291        2
94292        2
94293        2
94294        2
```

```
> dim(Webmt1)
```

```
[1] 127730     10
```

Apart from the `Tstart`, `Tstop` and `status` variables that represent the event time information in the counting process format, a column `weight.cens` has been created that contains the censoring weights. If there had been left truncated data (specified via the *Tstart* argument in `crprep`), another column `weight.trunc` with the

truncation weights would have been created. The variables that we transferred via the *keep* argument, score, age and type, don't change value within an individual. Two further columns have been added. count gives the row number in the sequence of rows from the same individual. Its usefulness will become apparent in Chapter 4. failcode refers to the event of interest; it serves to separate the weighted data sets by event type.

In the first ten rows, relapse is the event of interest. Since there were no death events among the first 7 individuals, no extra extra follow-up was created for them. Individual number 8 died, and remains in the risk set after death with a weight that changes over time. The second half of the data set is used for the analyses with death as event of interest. Here we see that individual 2, who had a relapse observed, remains in the risk set after relapse, again with the weight that changes over time.

It is seen that the number of rows has increased from 1977 in ebmt to 127730 in the weighted data set.

Once we have created the data set with weights, it is easy to obtain the estimates.

```
> cuminc.3 <- survfit(Surv(Tstart,Tstop,status==failcode)~failcode,
+                                          data=Webmt1, weights=weight.cens)
```

For investigation of the estimates, we can use the strength and flexibility of the existing functions in the survival package. We easily obtain the estimates and confidence intervals at one and five years via

```
> summary(cuminc.3[1], times=c(1,5))

Call: survfit(formula = Surv(Tstart, Tstop, status == failcode) ~ failcode,
    data = Webmt1, weights = weight.cens)

 time n.risk n.event survival std.err lower 95% CI upper 95% CI
    1   1588     302    0.844 0.00826        0.828        0.860
    5    369     146    0.742 0.01118        0.720        0.764

> summary(cuminc.3[2], times=c(1,5))

Call: survfit(formula = Surv(Tstart, Tstop, status == failcode) ~ failcode,
    data = Webmt1, weights = weight.cens)

 time n.risk n.event survival std.err lower 95% CI upper 95% CI
    1   1306     595    0.694  0.0105        0.674        0.715
    5    332      87    0.637  0.0115        0.615        0.660
```

A drawback may be that they are given on the survival scale. A summary on the scale of the cumulative incidence is obtained via

```
> F1 <- summary(cuminc.3[1], times=c(1,5))
> F2 <- summary(cuminc.3[2], times=c(1,5))
> 1-matrix(c(F1$surv,F1$upper,F1$lower),nrow=2)

       [,1]   [,2]   [,3]
[1,] 0.1562 0.1399 0.1723
[2,] 0.2583 0.2361 0.2799

> 1-matrix(c(F2$surv,F2$upper,F2$lower),nrow=2)

       [,1]   [,2]   [,3]
[1,] 0.3060 0.2852 0.3262
[2,] 0.3633 0.3405 0.3854
```

The confidence intervals are almost the same as in (a). They are computed on the same "log" scale (1.20). They slightly differ from (a) because they are calculated based on the product-limit estimator instead of the Aalen-Johansen estimator.

Compare the estimates with the relapse-free survival distribution, i.e. the probability to remain free of both end points.

Both cause-specific cumulative incidences add up to the overall incidence that was calculated in the previous chapter. For example, after one year we have $F_1(1) + F_2(1) = 0.156 + 0.306 = 0.462 = 1 - 0.538 = \overline{F}(1)$.

2. Some plots

(a) Plot the estimated cause-specific cumulative incidence for each end point using the overlaid display format. Plot the 95% confidence intervals on the scale of your choice.

Making plots in the overlaid format on the scale of the cumulative incidence is straightforward. In Figure 4 we show the curve based on the `plot.etmCIF` function based on `cuminc.2`, using the "log" scale.

```
> plot(cuminc.2, ci.type="pointwise", ci.fun="log",  ylim=c(0,0.4))
```

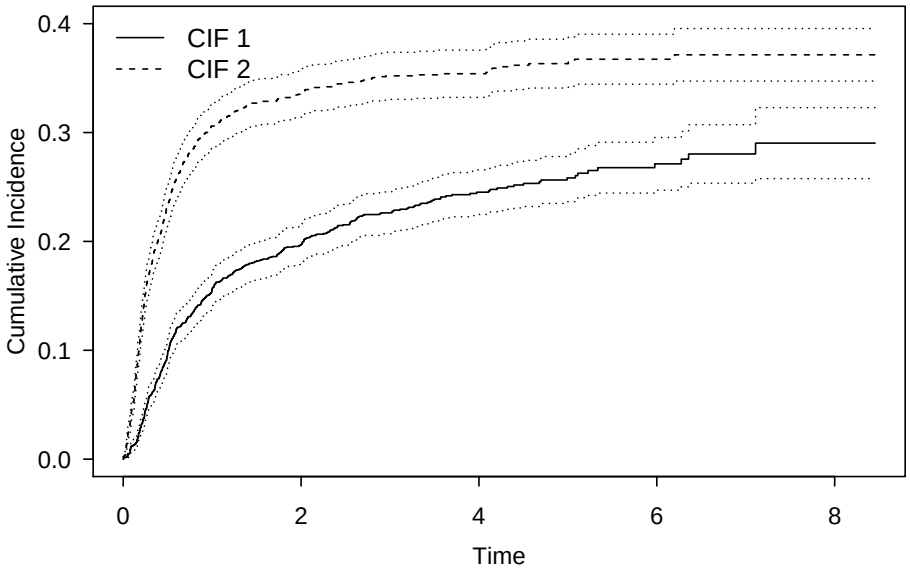

Figure 4: Estimate of cause-specific cumulative incidence, overlaid display format.

The curves based on the other three approaches from exercise 1 are very similar if we use[1]:

```
> plot(cuminc.1, conf.int=TRUE, mark.time=FALSE, ylim=c(0,0.4))
> plot(cuminc.3, conf.int=TRUE, mark.time=FALSE, fun="event", ylim=c(0,0.4))
```

(b) Plot the estimated cause-specific cumulative incidence using the stacked format (without the confidence intervals). First plot the relapse-specific cumulative incidence, and plot the death-specific cumulative incidence on top of this curve.

In order to use the correct range on the y-axis, we first plot the overall cumulative incidence, i.e. the estimate for both end points combined. Then we add the curve for relapse using the `lines.survfit` function. In Figure 5 we show the result using approach (a).

Using approaches (b) and (c), the second line of code is replaced by

```
> lines(cuminc.2[[1]], tr.choice="0 1", conf.int=FALSE, ci.fun="log", lwd=2, lty=2)
> lines(cuminc.3[1], mark.time=FALSE, conf.int=FALSE, fun="event", lwd=2, lty=2)
```

[1] Approach (a) gives an error message in package version 2.38-1 if we had chosen the log-log scale for the confidence interval.

```
>   plot(KM.overall, mark.time=FALSE, lwd=2, lty=1, xaxs="i", ylim=c(0,0.7),
+              conf.int=FALSE, fun="event", xlab="time since transplant")
>   lines(cuminc.1[1],  mark.time=FALSE, lwd=2, lty=2, conf.int=FALSE)
>   text(c(6,6),c(0.1,0.4),c("relapse","death"))
```

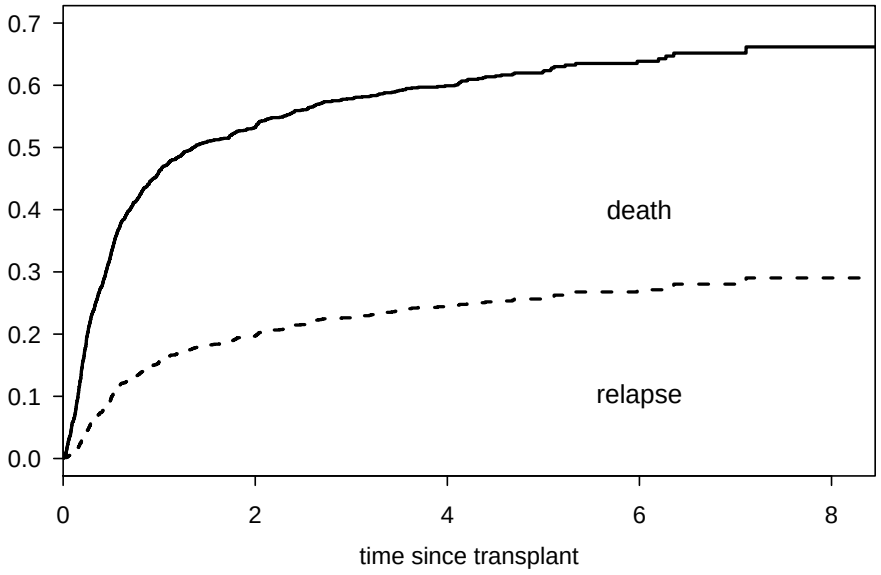

Figure 5: Estimate of cause-specific cumulative incidence, stacked display format.

Note that there is no `lines.etmCIF` function in `etm`. If we want to add a curve, we use the `lines.etm` function, which needs as input the first component of the list `cuminc.2`.

(c) Plot the cause-specific cumulative incidence estimates for each end point using the alternate display format. You can leave out the confidence intervals.

In Figure 6 we show the result from approach (c). We also include the confidence intervals.

Using approach (a), we obtain exactly the same curves if we write

```
> plot(cuminc.1[2], fun="identity", mark.time=FALSE,
+                   xlab="time since transplant", ylim=c(0,1))
> lines(cuminc.1[1], mark.time=FALSE)
```

The `plot.etmCIF` function can only plot on the scale of the cumulative incidence, hence the alternate format cannot be chosen. It is possible to create the alternate display format–all estimates are there–but then we have to distill the relevant components ourselves from `cuminc.2[[1]]` or `summary(cuminc.2)[[1]]`; for the confidence intervals we have to use `summary(cuminc.2)[[1]]`.

```
> plot(cuminc.2, which.cif=1, xlab="time since transplant",
+                                  ci.type="pointwise", legend=FALSE)
> with(cuminc.2[[1]], lines(time,1-est[1,3,]))
> with(summary(cuminc.2)[[1]][["CIF 2"]], lines(time,1-lower,lty=2))
> with(summary(cuminc.2)[[1]][["CIF 2"]], lines(time,1-upper,lty=2))
```

```
> plot(cuminc.3[2], mark.time=FALSE, xlab="time since transplant")
> lines(cuminc.3[1], mark.time=FALSE, fun="event")
> text(c(6,6),c(0.1,0.9),c("relapse","death"))
```

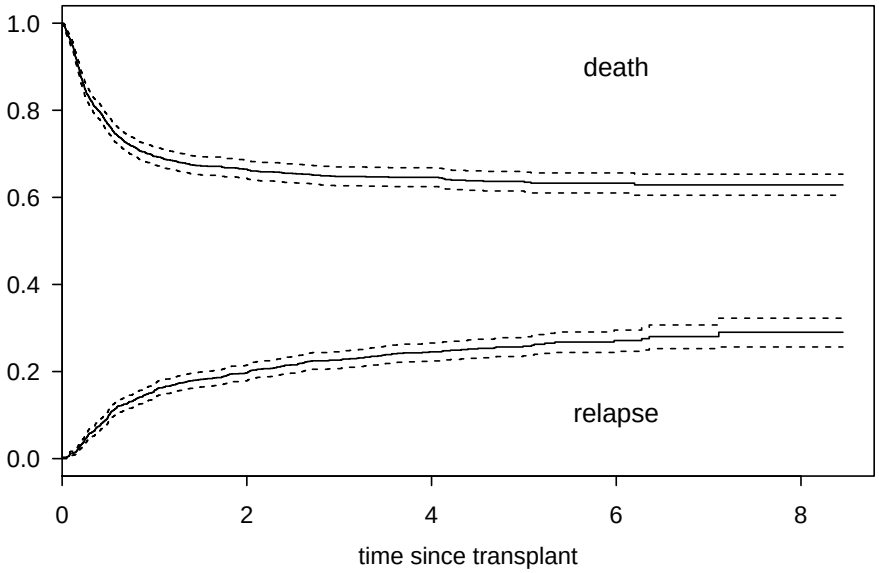

Figure 6: Estimate of cause-specific cumulative incidence, alternate display format.

3. Effect of EBMT risk score

We investigate the effect of the EBMT risk score on the cause-specific cumulative incidence (the variable score in the data set).

Estimate the cause-specific cumulative incidence for each of the three EBMT risk scores and for both end points. Use the Aalen-Johansen form (i.e. approaches (a) or (b)).

Calculations are very similar to the overall curves. The only thing that changes is the specification of `score` on the right hand side of the formula.

```
> cuminc.1.score <- survfit(Surv(time,stat,type="mstate")~score, ebmt1)
> cuminc.2.score <- etmCIF(Surv(time,stat!=0)~score, ebmt1, etype=stat,
+                                                    failcode=c(1,2))
```

Plot the estimates; use one plot window for relapse and another one for death.

In Figure 7 we show the curves via approach (b).

```
>   par(mfrow=c(1,2))
>   plot(cuminc.2.score, which.cif=1, ylim=c(0,0.55), legend.pos="bottomright",
+                                                     curvlab=levels(ebmt1$score))
>   plot(cuminc.2.score, which.cif=2, ylim=c(0,0.55), legend.pos="bottomright",
+                                                     curvlab=levels(ebmt1$score))
```

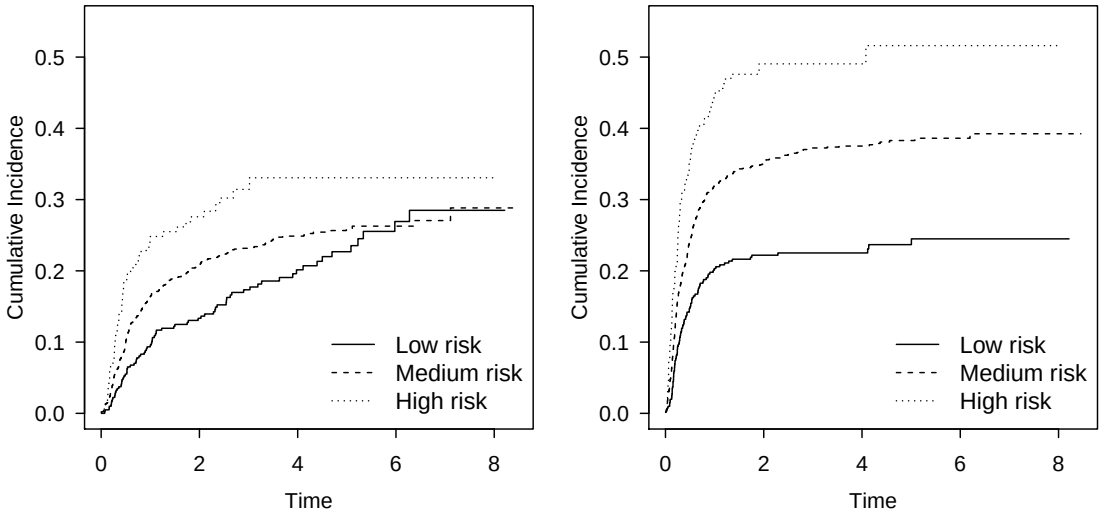

Figure 7: Estimate of cause-specific cumulative incidence by EBMT risk score.

In approach (a), the code to produce the same plots is considerably more difficult. The reason is that the `plot.survfit` function that is invoked plots all six curves in one plot. The `survfit` function that computes the curves treats the `score` variable as a stratum. The only way to obtain the plots as in figure 7 is by distilling the components from the `cuminc.1.score` object. The estimates are in the `prev` component of `cuminc.1.score`, which is a matrix with one column per event type. The estimates per value of `score` are appended in one vector. They can be separated using the information in the `strata` component, which gives the number of values that belong to the respective strata.

```
> par(mfrow=c(1,2))
> with(cuminc.1.score, plot(time[1:strata[1]], prev[1:strata[1],1],
+          type="s", ylim=c(0,0.55), xlab="Time", ylab="Cumulative Incidence"))
> with(cuminc.1.score, lines(time[(strata[1]+1):sum(strata[1:2])],
+                          prev[(strata[1]+1):sum(strata[1:2]),1],type="s",lty=2))
> with(cuminc.1.score, lines(time[(sum(strata[1:2])+1):sum(strata[1:3])],
+                 prev[((sum(strata[1:2])+1)):sum(strata[1:3]),1],type="s",lty=3))
> legend("bottomright", levels(ebmt1$score), lty=1:3,bty="n")
>
> with(cuminc.1.score, plot(time[1:strata[1]],prev[1:strata[1],2],
+         type="s", ylim=c(0,0.55), xlab="Time", ylab="Cumulative Incidence"))
> with(cuminc.1.score, lines(time[(strata[1]+1):sum(strata[1:2])],
+                          prev[(strata[1]+1):sum(strata[1:2]),2],type="s",lty=2))
> with(cuminc.1.score, lines(time[(sum(strata[1:2])+1):sum(strata[1:3])],
+                 prev[((sum(strata[1:2])+1)):sum(strata[1:3]),2],type="s",lty=3))
> legend("bottomright", levels(ebmt1$score), lty=1:3,bty="n")
```

What do you conclude with respect to the effect of the EBMT risk score. Are the effects proportional?

It is clear that the EBMT risk score has a profound influence on both relapse and death. Note that the low risk and

medium risk curves cross for relapse as end point. This suggests some form of non-proportionality. In Chapters 4 and 5 this is investigated more formally.

4. Choice of the weight function in the product-limit form

If we want to calculate the product-limit form that is equivalent to the Aalen-Johansen form for each value of score, the weights based on the censoring distribution need to be calculated separately as well. Create a data set that has the score-specific weights and name it `Webmt.score`.

We obtain separate censoring weights per value of EBMT score by using the *strata* argument in the `crprep` function. We define the transitions by name, but we could have used numbers as well, as we did when creating `Webmt1`.

```
> Webmt.score <- crprep(Tstop="time", status="type", data=ebmt1,
+     trans=c("Relapse","Death"), cens="Event free", id="patid", strata="score")
```

Make plots to compare the results with the curves when we use the same weights per value of EBMT score, as was created in the data set `Webmt1`. What do you conclude with respect to the choice of the weight function?

We make a separate plot per event type, and use the overlaid format for the values of EBMT risk score. We compare the estimates using the score-specific weights with the ones using one overall weight. Contrary to approach (a), we can compute the estimates for each event type separately, which makes creation of the plots easier.

If we use one overall weight function $\widehat{\widehat{\Gamma}}$, not stratified by the value of **score**, we obtain almost the same estimates.

The statistic $\widehat{\widehat{\Gamma}}$ can be interpreted as an estimate of the time to censoring distribution. Therefore, we check to what extent this distribution depends on the value of the EBMT risk score. Estimate and plot the time to censoring distribution for each value of score. Does the distribution depend on the value of the EBMT risk score? Does this observation correspond to the amount of dependence on the weight function in the estimate of the crude risk?

We change the role of event and censoring. This is a classical survival setting, not a competing risks setting. Therefore, we can use standard code for survival analysis. We obtain figure 9.

The time-to-censoring distribution is almost the same for the low and medium risk groups. For the high risk group, the censoring occurs earlier. We can also use a Cox model to quantify the difference.

```
> coxph(Surv(time,stat==0)~score,data=ebmt1)

Call:
coxph(formula = Surv(time, stat == 0) ~ score, data = ebmt1)


                   coef exp(coef) se(coef)    z     p
scoreMedium risk 0.1027    1.1082   0.0779 1.32 0.187
scoreHigh risk   0.4231    1.5267   0.1758 2.41 0.016

Likelihood ratio test=5.76  on 2 df, p=0.0561
n= 1977, number of events= 836
```

The difference is statistically significant. The reason why the choice of the weight function has a marginal effect on the estimates is probably because most events in the high-risk group occur in the first two years, when the curves in Figure 9 are still fairly similar.

5. Log-rank tests **Run and compare the three commands given below. Why is the output from the first and the second different and what do these commands test? Why do the second and the third give the same value of the test statistic?**

```
> par(mfrow=c(1,2))
> plot(survfit(Surv(Tstart,Tstop,status=="Relapse")~score,
+   data=subset(Webmt.score,failcode=="Relapse"), weights=weight.cens), lwd=3,
+   mark.time=FALSE, col="black", fun="event", ylim=c(0,0.5))
> lines(survfit(Surv(Tstart,Tstop,status==1)~score, data=subset(Webmt1,failcode==1),
+               weights=weight.cens), lwd=1, mark.time=FALSE, col="red", fun="event")
> title("Relapse")
>
> plot(survfit(Surv(Tstart,Tstop,status=="Death")~score,
+   data=subset(Webmt.score,failcode=="Death"), weights=weight.cens), lwd=3,
+   mark.time=FALSE, col="black", fun="event", ylim=c(0,0.5))
> lines(survfit(Surv(Tstart,Tstop,status==2)~score, data=subset(Webmt1,failcode==2),
+               weights=weight.cens), lwd=1, mark.time=FALSE, col="red", fun="event")
> title("Death")
> legend("bottomright", levels(Webmt1$score), col=c("black","red","green"), lwd=3)
```

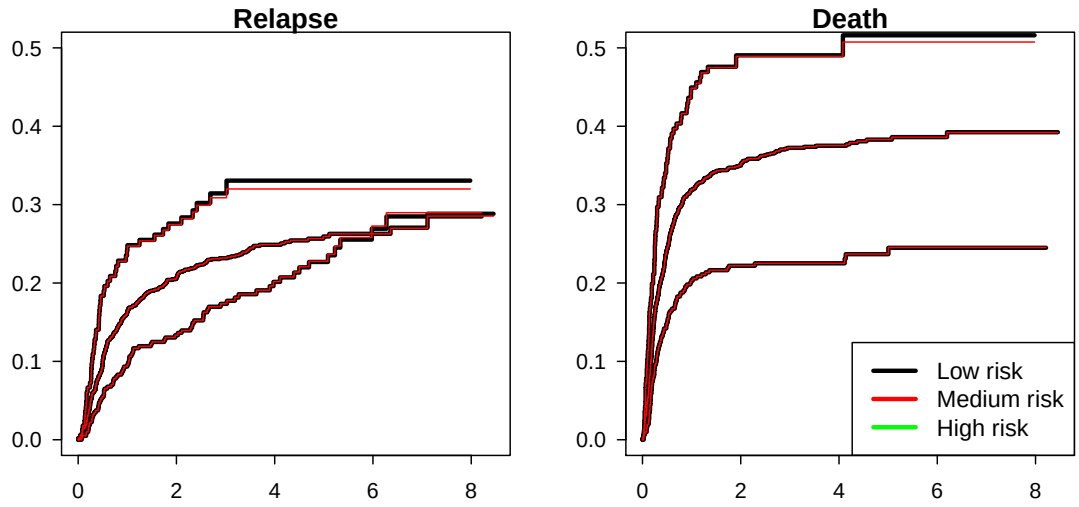

Figure 8: Effect of choice in weight function to correct for censoring in product-limit form. Black: stratum specific weight; red: one overall weight

```
> coxph(Surv(Tstart,Tstop,status=="Relapse")~score,
+                              data=subset(Webmt.score,failcode=="Relapse"),
+                                          weights=weight.cens)$score

[1] 13.37

> coxph(Surv(Tstart,Tstop,status=="Relapse")~score,
+         data=subset(Webmt.score,failcode=="Relapse"&count==1))$score

[1] 36.91

> survdiff(Surv(time,stat==1)~score,data=ebmt1)

Call:
survdiff(formula = Surv(time, stat == 1) ~ score, data = ebmt1)

                     N Observed Expected (O-E)^2/E (O-E)^2/V
score=Low risk     406       79    116.4    12.000     16.21
```

```
> par(mfrow=c(1,1))
> plot(survfit(Surv(time,stat==0)~score, data=ebmt1), mark.time=FALSE,
+                                                col=c("black","red","green"))
```

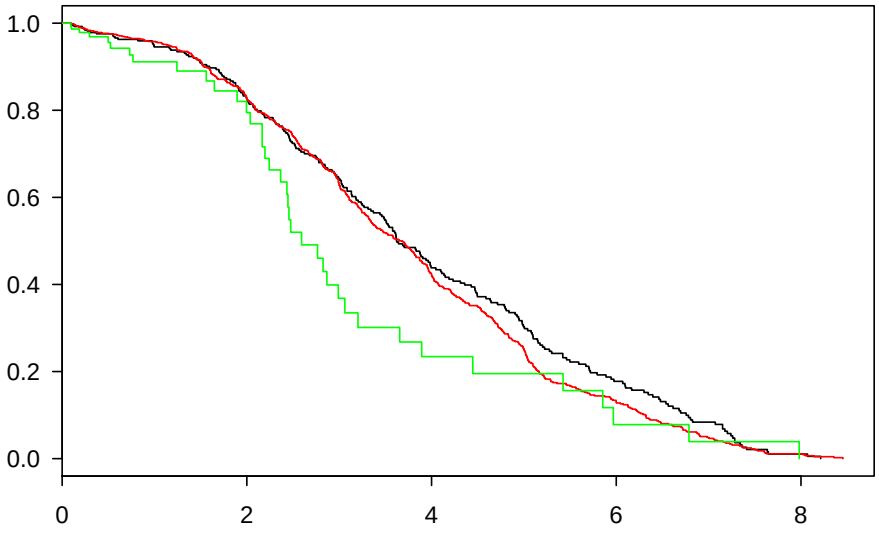

Figure 9: Kaplan-Meier estimate of time-to-censoring distribution, Black: low risk; red: medium risk; green: high risk.

```
score=Medium risk 1404      328    315.0     0.539      1.75
score=High risk    167       49     24.7    24.023     25.54

 Chisq= 36.9  on 2 degrees of freedom, p= 9.72e-09
```

Answer. The first line of code performs a log-rank test with respect to the subdistribution hazards. Hence, it tests whether the relapse-specific cumulative incidences are equal for the three EBMT scores. The second and third perform a log-rank test on the cause-specific hazards. Hence, they perform the nonparametric test for equality of the cause-specific hazards.

Chapter 3. Intermediate Events. Nonparametric Estimation.

We look beyond relapse and also quantify the transition hazard and transition probability from relapse to death. We use a multi-state model with three states, 1: Transplant (T), 2: Relapse (R), and 3: Death (D). The are three possible transitions: 1: T $\rightarrow$ R, 2: R $\rightarrow$ D and 3: T $\rightarrow$ D. We will show code and results from all three packages `etm`, `msSurv` or `mstate`.

1. Define the structure

Make a directed graph of the multi-state model. Indicate the state numbers and names as well as the transition numbers.

The model is an illness-death model which can be represented as in Figure 10.

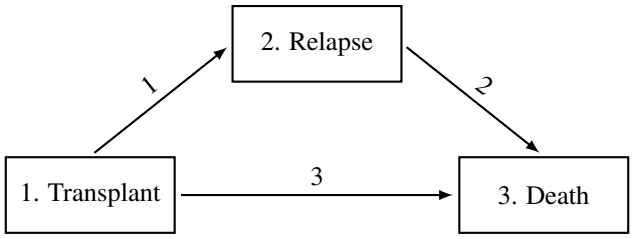

Figure 10: Graphical representation of disease process from transplant to death, with relapse as possible intermediate state.

2. Create the stacked data set

What is the format of the `ebmt1` data set?

Data are in state ordered wide format.

Transform it into a format that can be used for estimation of the transition probabilities. Compare the data of the first two patients with those in the original wide format and see whether this makes sense to you.

All three packages require the data to be in long format; `etm` and `msSurv` need the transition based format, whereas `mstate` needs the stacked format. You can create the long format in your program of choice, but `etm` and `mstate` also have a function to create the required format. Before we can use these functions, we first need to define the transitions.

etm We define a 3 $\times$ 3 transition matrix with logical values. Allowed direct transitions have the value *TRUE*; transitions that are not allowed have the value *FALSE*. Although not necessary, we can attach names to the states via the `dimnames` function.

```
> library(etm)
> tmat.etm <- matrix(FALSE,nrow=3,ncol=3)
> tmat.etm[rbind( c(1,2), c(1,3), c(2,3) )] <- TRUE
> dimnames(tmat.etm) <- list(c("T","R", "D"), c("T","R","D"))
> tmat.etm
```

```
      T     R     D
T FALSE  TRUE  TRUE
R FALSE FALSE  TRUE
D FALSE FALSE FALSE
```

We use the `etmprep` function to transform the data set to the transition based long format. Note that the order of the elements of the *time* and *status* arguments has to be the same as the order of the states as specified in the transition matrix.

```
> ebmt.etm <- etmprep(time=c(NA,"rel","srv"), status=c(NA,"relstat","srvstat"),
+    state.names=c("transplant","relapse", "death"), data=ebmt1, tra=tmat.etm,
+                                                          cens.name="cens")
> ebmt1[1:2,]
  patid  srv srvstat  rel relstat      yrel age       score       type stat   time
1     1 1610       0 1610       0      <NA>  28 Medium risk Event free    0 4.4079
2     2  961       1  165       1 1997-1999  33 Medium risk    Relapse    1 0.4517
> subset(ebmt.etm, id %in% c(1,2))
  id entry exit       from      to
1  1     0 1610 transplant    cens
2  2     0  165 transplant relapse
3  2   165  961    relapse   death
```

Since the **patid** column in `ebmt1` has the values 1 to 1977, there is no need to specify the column name in the *id* argument. Note that the name of the column with the person identifiers is always changed to `id`. Individual 1 had no event. Since it was censored in the `transplant` state, it only has one row in the transition-based long format. Individual 2 had a relapse at 165 days, after which he died at 961 days. This information is spread over two rows.

---

msSurv The `msSurv` package does not have a function to transform the data to the transition based long format. We could use the code from the `etm` package to create the data set, as was shown for the example in the book. After this, we have to rename the columns and transform the state values as required by the `msSurv` package.

```
> ebmt.msSurv <- ebmt.etm # as created via the etm package
> names(ebmt.msSurv) <- c("id","start","stop","start.stage","end.stage")
> ebmt.msSurv$start.stage <- as.numeric(factor(ebmt.msSurv$start.stage,
+                                levels=c("transplant","relapse", "death")))
> ebmt.msSurv$end.stage <- as.numeric(factor(ebmt.msSurv$end.stage,
+                      levels=c("cens","transplant","relapse", "death")))-1
```

Although the states 1 to 3 need to be in character format when specifying the allowed transitions in the `graphNel` object (see below), the values of the states can be numeric.

An alternative is to create the data set ourselves. For an illness-death model this is not too difficult. We first specify the values for the first transition. If the first transition was a relapse (`subset(ebmt1, relstat==1)`), we specify the values for the subsequent transition to death. These data frames are combined via `rbind`.

```
> ebmt.msSurv <- with(ebmt1, data.frame(id=patid, start=0, stop=pmin(srv,rel),
+                       start.stage=1, end.stage=ifelse(relstat==1,2,3*srvstat)))
> tmp <- with(subset(ebmt1, relstat==1), data.frame(id=patid, start=rel, stop=srv,
+                                          start.stage=2, end.stage=3*srvstat))
> ebmt.msSurv <- rbind(ebmt.msSurv, tmp)
> ebmt.msSurv <- ebmt.msSurv[order(ebmt.msSurv$id,ebmt.msSurv$start),]
> ebmt1[1:2,]
  patid  srv srvstat  rel relstat      yrel age       score       type stat   time
1     1 1610       0 1610       0      <NA>  28 Medium risk Event free    0 4.4079
2     2  961       1  165       1 1997-1999  33 Medium risk    Relapse    1 0.4517
> subset(ebmt.msSurv, id %in% c(1,2))
     id start stop start.stage end.stage
1     1     0 1610           1         0
2     2     0  165           1         2
1978  2   165  961           2         3
```

Individual 1 had no event. Since it was censored in the `transplant` state, it only has one row in the transition based long format. Individual 2 had a relapse at 165 days, after which he died at 961 days. This information is spread over two rows.

---

mstate In the transition matrix, allowed direct transitions are numbered 1 through 3. If the (i, j) entry of this matrix equals k, then the kth transition runs from state i to state j. *NA* is used for the elements on the diagonal and the transitions that are not allowed. Because illness-death models are a common type of multi-state models, a function has been defined in `mstate` to construct the transition matrix.

```
> tmat.mst <- trans.illdeath(names=c("transplant","relapse","death"))
> tmat.mst
```

```
            to
from         transplant relapse death
  transplant         NA       1     2
  relapse            NA      NA     3
  death              NA      NA    NA
```

The same matrix as above can be created by hand as

```
> tmat.mst <- matrix(NA,3,3)
> tmat.mst[1,2:3] <- 1:2
> tmat.mst[2,3] <- 3
> dimnames(tmat.mst) <- list(c("transplant","relapse","death"),
+                                          c("transplant","relapse","death"))
```

The dimnames to define the states are not required. Since the `trans.illdeath` function numbers the states by row and from left to right, we need to create the matrix by hand if we want another numbering of the transitions.

We use the `msprep` function to transform the data set to the required stacked long format. Note that the order of the elements in the *time* and *status* arguments has to be the same as the order of the states as specified in the transition matrix.

```
> ebmt.mstate <- msprep(time=c(NA,"rel","srv"), status=c(NA,"relstat","srvstat"),
+                                                  data=ebmt1, trans=tmat.mst)
> ebmt1[1:2,]
```

```
  patid  srv srvstat  rel relstat      yrel age       score       type stat   time
1     1 1610       0 1610       0      <NA>  28 Medium risk Event free    0 4.4079
2     2  961       1  165       1 1997-1999  33 Medium risk    Relapse    1 0.4517
```

```
> subset(ebmt.mstate, id %in% c(1,2))
```

```
An object of class 'msdata'

Data:
  id from to trans Tstart Tstop time status
1  1    1  2     1      0  1610 1610      0
2  1    1  3     2      0  1610 1610      0
3  2    1  2     1      0   165  165      1
4  2    1  3     2      0   165  165      0
5  2    2  3     3    165   961  796      1
```

The result is an object of class `msdata`, which is a data frame with the information in stacked long format, and with the transition matrix as an attribute. Individual 1 had no event; he was censored in state 1 (`transplant`). Since there are two possible competing events from state 1, he has two rows in the stacked format. The values in `Tstart`, `Tstop` and `time` are equal in both rows, and `trans` denotes the event type. Individual 2 had a relapse at 165 days, after which he died at 961 days. Since there is only one state that can be reached from relapse, he has one extra row in the stacked data set, hence three rows in total.

---

3. Estimation of cumulative hazard Computation of the transition probability is based on the hazard. But inspection of the hazard itself can be informative as well. Therefore, we first have a look at the cumulative hazard.

Compute the cumulative hazard function for all three transitions, using a clock-forward approach. Plot them in one single figure in overlaid display format.

Compare the estimate for the transition from relapse to death with the one that is computed based on a Cox model without covariables. Add this estimate to the plot.

etm We obtain the estimates via the `etm` function that computes transition probabilities, but it is easier to use the `mvna` function from the `mvna` package that has been written for this purpose. It a computes all cumulative hazards at once.

```
> library(mvna)
> NA.etm <- mvna(ebmt.etm, c("transplant","relapse", "death") , tmat.etm,
+                                                          cens.name="cens")
```

We plot the estimates using the `plot.mvna` function with default values of the arguments (Figure 11).

```
> plot(NA.etm, xlab="time since transplant (days)")
```

We can also compute the estimate for relapse using the `coxph` function, and add it to the plot via

```
> CumHaz.2to3 <- survfit(coxph(Surv(entry,exit,to=="death")~1,
+                                    data=subset(ebmt.etm,from=="relapse")))
> lines(CumHaz.2to3,col="red", fun="cumhaz", mark.time=FALSE)
```

You will see that the same estimate is obtained.

---

msSurv We compute the hazard and the transition probabilities via the `msSurv` function. Before we can use this function, we first need to define the allowed transitons by creating a `graphNel` tree object.

```
> library(msSurv)
> Nodes <- as.character(1:3)
> Edges <- list("1" = list(edges=c("2","3")),
+               "2" = list(edges=c("3")),
+               "3" = list(edges=NULL))
>
> trans.ms <- new("graphNEL", nodes=Nodes, edgeL=Edges, edgemode="directed")
```

Computations are done via

```
> Prob.msSurv <- msSurv(ebmt.msSurv, trans.ms)

Entry distributions calculated for states 3 .

Exit distributions calculated for states 1 .
```

The package does not have a standard function to plot the cumulative hazard. We need to distill the relevant information from the `Prob.msSurv` object, after which we use the standard `plot` function. The package works with S4 classes, in which objects have *slots*. The `et` slot gives the transition times. The estimates of the transition hazards can be obtained from the off-diagonal elements in the matrices that make up the Aalen-Johansen estimate. They are in the `I.dA` slot. We obtain Figure 11 via

```
> plot(et(Prob.msSurv),cumsum(I.dA(Prob.msSurv)[1,2,]),lty=1,ylim=c(0,1.7),
+      type="s", xlab="time since transplant (days)", ylab="Cumulative Hazard")
> lines(et(Prob.msSurv),cumsum(I.dA(Prob.msSurv)[1,3 ,]),lty=2,type="s")
> lines(et(Prob.msSurv),cumsum(I.dA(Prob.msSurv)[2,3 ,]),lty=3,type="s")
> legend("topleft", legend=c("transplant relapse","transplant death",
+                                        "relapse death"), lty=c(1,2,3),bty="n")
```

We can also compute the estimate for relapse using the `coxph` function, and add it to the plot via

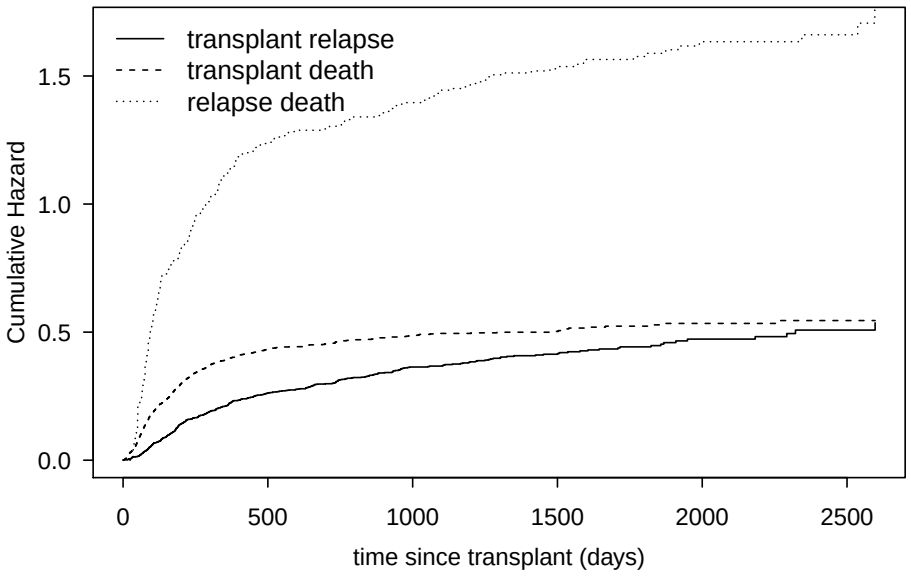

Figure 11: Estimate of cumulative hazards.

```
> CumHaz.2to3 <- survfit(coxph(Surv(entry,exit,to=="death")~1,
+                                    data=subset(ebmt.etm,from=="relapse")))
> lines(CumHaz.2to3,col="red", fun="cumhaz", mark.time=FALSE)
```

You will see that the same estimate is obtained.

Instead of using the `I.dA` slot, we can also use the `dNs` and `Ys` slots, which give the number of jumps and the number at risk. The code is slightly more complicated. Since there are no individuals in the relapse state at the first observed transition at 0.5, the number at risk is zero. Therefore, we need to remove the first elements in `dNs(Prob.msSurv)` and `Ys(Prob.msSurv)` for the transition from relapse to death.

```
> plot(et(Prob.msSurv),
+     cumsum(dNs(Prob.msSurv)[,"dN 1 2"]/Ys(Prob.msSurv)[,"y 1"]),
+                                      lty=1,ylim=c(0,1.7), type="s",
+      xlab="time since transplant (days)",ylab="Cumulative Hazard")
> lines(et(Prob.msSurv),
+       cumsum(dNs(Prob.msSurv)[,"dN 1 3"]/Ys(Prob.msSurv)[,"y 1"]),
+       lty=2,type="s")
> lines(et(Prob.msSurv)[-1],
+       cumsum(dNs(Prob.msSurv)[-1,"dN 2 3"]/Ys(Prob.msSurv)[-1,"y 2"]),
+       lty=3,type="s")
> legend("topleft",legend=c("transplant relapse","transplant death",
+                                  "relapse death"), lty=c(1,2,3),bty="n")
```

---

mstate The stacked format allows to calculate all cumulative hazards via the basic `coxph` function.

```
> NA.surv <- coxph(Surv(Tstart,Tstop,status)~strata(trans), data=ebmt.mstate,
+                                                              ties="breslow")
```

The `strata(trans)` specification is used to calculate all three estimates at once. Since there are tied event times, we need to specify `ties="breslow"` in order to obtain the Aalen-Johansen estimator of the transition probability. If we only want the cumulative hazard for the relapse $\rightarrow$ death transition, we can select the rows that refer to transition 3.

```
> CumHaz.2to3 <- survfit(coxph(Surv(Tstart,Tstop,status)~1,
+                                    data=subset(ebmt.mstate,trans==3)))
```

We can use the `plot.survfit` function from the survival package to plot the estimates.

```
> plot(survfit(NA.surv), fun="cumhaz", mark.time=FALSE)
```

The alternative is to first apply the `msfit` function, which we also need when computing the transition probabilities.

```
> NA.mstate <- msfit(NA.surv, vartype="greenwood", trans=tmat.mst)
```

It transforms the estimates to another format, for which there is a special `plot.msfit` function, and we obtain Figure 12.

```
> plot(NA.mstate)
```

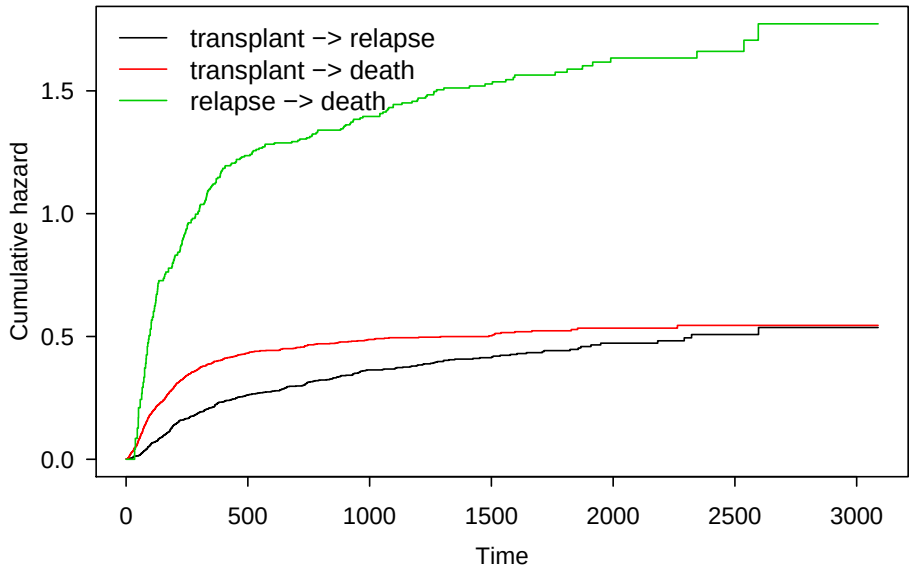

Figure 12: Estimate of cumulative hazards based on `mstate` package.

If an individual is still free of relapse, which of the events is more likely to happen, relapse or death? What is the effect of relapse on the mortality rate?

For this, we look at the transition rate. It is the slope of the cumulative hazard that we plotted in Figure 11 or 12. Early after transplant, the death rate is higher than the relapse rate, but this changes after about 500 days. The higher death rate in the beginning is due to transplant-related mortality. After a relapse, the transition rate to death becomes much larger. Hence, relapse is an event that greatly worsens prognosis.

4. Estimation of transition probabilities The cumulative hazard is one part of the story; it reflects the instantaneous transition rate. But we also want to know the probability to be in each of the three states over time.

Compute the fixed history transition probabilities for all three transitions from the time origin onwards. Give a numeric summary of the state prevalence at 5 years after transplant. Compute the estimates as well as the 95% confidence intervals on the linear scale as defined in (1.18) on Page 34.

The state prevalence is another word for state occupation probability; we compute the estimates $\widehat{P}_{1,k}(0,t)$ for $k = 1,2,3$.

etm We use the `etm` function to calculate the estimates.

```
> Prob.etm <- etm(ebmt.etm, c("transplant","relapse", "death") , tmat.etm,
+                                                    cens.name="cens", s=0)
```

All information is in the output of the `summary.etm` function. We cannot select the time points at which we want the numeric values via this function; we obtain what we want via the `subset` and `tail` functions.

```
> Prob.etm.summ <- summary(Prob.etm, ci.fun="linear")
> tail(subset(Prob.etm.summ[["transplant relapse"]],time<=5*365.25),1)
         P time       var  lower  upper n.risk n.event
1826 0.145 1826 0.0001007 0.1253 0.1647    198       0
> tail(subset(Prob.etm.summ[["transplant death"]],time<=5*365.25),1)
          P time       var  lower  upper n.risk n.event
1826 0.4766 1826 0.0001566 0.4521 0.5012    198       0
> tail(subset(Prob.etm.summ[["transplant transplant"]],time<=5*365.25),1)
          P time       var  lower  upper n.risk n.event
1826 0.3784 1826 0.0001557 0.3539 0.4028    198       0
```

---

msSurv Any estimate can be obtained by selecting the apropriate slot in the output of the `msSurv` function as obtained in Exercise 3.8.3.

The estimates of the state occupation probabilities at five years are best obtained via the `SOPt` slot. We have to calculate the confidence intervals ourselves based on the estimated variance.

```
> tmp <- SOPt(Prob.msSurv, t=5*365.25, covar=TRUE)
The state occupation probabilities at time 1826.25 are:
State 1: 0.3784
State 2: 0.145
State 3: 0.4766

Covariance estimates for state occupation probabilities:
State 1: 2e-04
State 2: 1e-04
State 3: 2e-04
> data.frame(P=tmp$SOPt, lci=tmp$SOPt-qnorm(0.975)*sqrt(tmp$var.sop),
+            uci=tmp$SOPt+qnorm(0.975)*sqrt(tmp$var.sop))
         P    lci    uci
p 1 0.3784 0.3539 0.4028
p 2 0.1450 0.1253 0.1647
p 3 0.4766 0.4521 0.5012
```

---

mstate We use the `probtrans` function to compute the estimates.

```
> Prob.mst <- probtrans(NA.mstate, predt=0, method="greenwood")
```

The result is a list, and the first component gives the estimates for the transitions out of state 1, which is what we need. We cannot select the time points at which we want the numeric values via this function; we obtain what we want via the `subset` and `tail` functions.

```
> tmp <- tail(subset(Prob.mst[[1]],time<=5*365.25),1)
> tmp.prob <- c(tmp$pstate1,tmp$pstate2,tmp$pstate3)
> Prob.occ5 <- data.frame(P=tmp.prob,
+                 lci=tmp.prob-qnorm(0.975)*c(tmp$se1,tmp$se2,tmp$se3),
+                 uci=tmp.prob+qnorm(0.975)*c(tmp$se1,tmp$se2,tmp$se3))
> Prob.occ5
```

```
       P    lci    uci
1 0.3784 0.3539 0.4028
2 0.1450 0.1253 0.1647
3 0.4766 0.4521 0.5012
```

---

5. Creation of informative plots The results are much better summarized by creating informative plots.

Make two different plots, without confidence intervals:

1. **All three state occupation probabilities in stacked format.**
2. **Transition probabilities for the three transitions T $\rightarrow$ R, R $\rightarrow$ D and T $\rightarrow$ D, in overlaid format.**

etm The `plot.etm` function does not allow to plot in stacked format. Therefore, we first create a data frame with all relevant information from the output of `summary.etm` that we obtained in Exercise 3.8.4.

```
> Prob.etm.from1 <- data.frame(time=Prob.etm.summ[["transplant relapse"]]$time,
+                              tr.rel=Prob.etm.summ[["transplant relapse"]]$P,
+                            tr.death=Prob.etm.summ[["transplant death"]]$P)
```

Now we can use the standard `plot` and `lines` functions to obtain Figure 13:

```
> plot(1-tr.death~time, data=Prob.etm.from1, type="s", ylim=c(0,1),
+       xlab="time since transplant (days)", ylab="State Occupation Probability")
> lines(1-I(tr.rel+tr.death)~time, data=Prob.etm.from1, type="s")
> text(rep(3000,3),c(0.2,0.4,0.7),c("T","R","D"))
```

For the second plot we don't need to first create a data frame. It is easily made from the `Prob.etm` object with the `plot.etm` function (Figure 14):

```
> plot(Prob.etm, tr.choice=c("relapse death","transplant death",
+                                                "transplant relapse"))
```

---

msSurv We use the `et` and `AJs` slots from the output of the `msSurv` function. The stacked state occupation probabilities (Figure 13) can be plotted via:

```
> plot(et(Prob.msSurv), AJs(Prob.msSurv)[1,1,], ylim=c(0,1), type="s",
+      xlab="time since transplant (days)", ylab="State Occupation Probability")
> lines(et(Prob.msSurv), AJs(Prob.msSurv)[1,1,]+AJs(Prob.msSurv)[1,2,], type="s")
> text(rep(2500,3),c(0.2,0.4,0.7),c("T","R","D"))
```

The transition probabilities for the direct transitions (Figure 14) can be obtained via

```
> plot(et(Prob.msSurv), AJs(Prob.msSurv)[2,3,], ylim=c(0,1), type="s",
+      xlab="time since transplant (days)", ylab="Transition Probability")
> lines(et(Prob.msSurv), AJs(Prob.msSurv)[1,3,], type="s", lty=2)
> lines(et(Prob.msSurv), AJs(Prob.msSurv)[1,2,], type="s", lty=3)
> legend("topleft", lty=1:3, bty="n",
+       legend=c("relapse death","transplant death","transplant relapse"))
```

---

mstate The standard `plot.probtrans` function by default plots the transition probabilities out of state 1 in stacked display format. Therefore, Figure 15 is simply obtained via.

```
> plot(Prob.mst)
```

The figure that combines the transition probabilities for all direct transitions is more difficult to obtain, because the only option in `plot.probtrans` function is to plot all transition probabilities from one specific state. We first need to create a data frame with all estimates.

```
> Prob.mst.123 <- data.frame(time=Prob.mst[[1]]$time,
+         tr.rel=Prob.mst[[1]]$pstate2, tr.death=Prob.mst[[1]]$pstate3,
+                                       rel.death=Prob.mst[[2]]$pstate3)
```

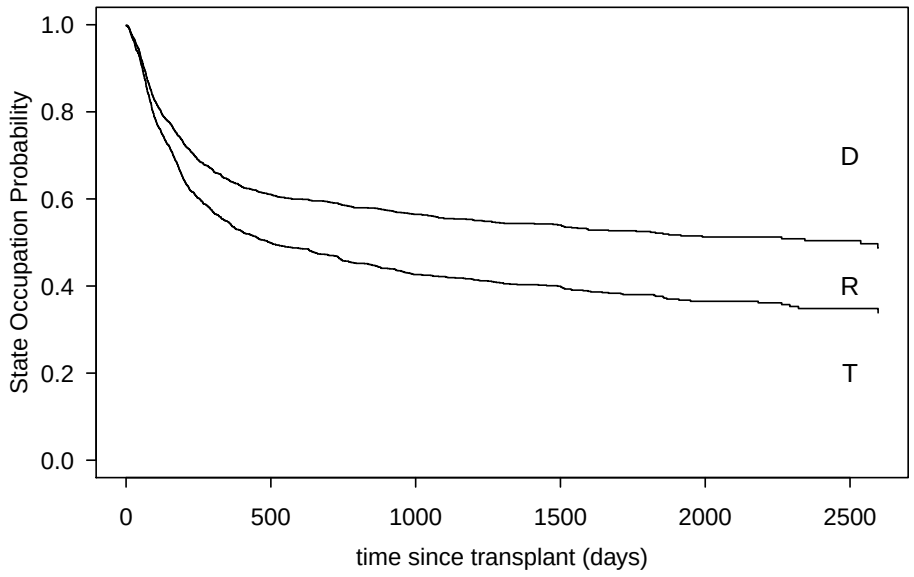

Figure 13: State occupation probabilities.

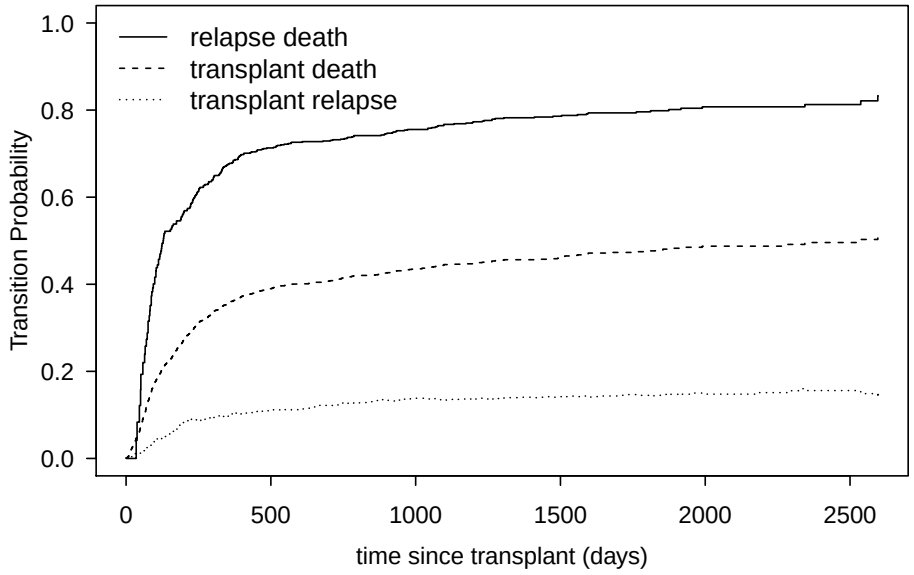

Figure 14: Transition probabilities for all direct transitions.

Next, we create Figure 14 via

```
> plot(rel.death~time, data=Prob.mst.123, type="s", ylim=c(0,1),
+      xlab="time since transplant (days)", ylab="Transition Probability")
> lines(tr.death~time, data=Prob.mst.123, type="s", lty=2)
> lines(tr.rel~time, data=Prob.mst.123, type="s", lty=3)
> legend("topleft",legend=c("relapse death","transplant death",
+                     "transplant relapse"), lty=1:3, bty="n")
```

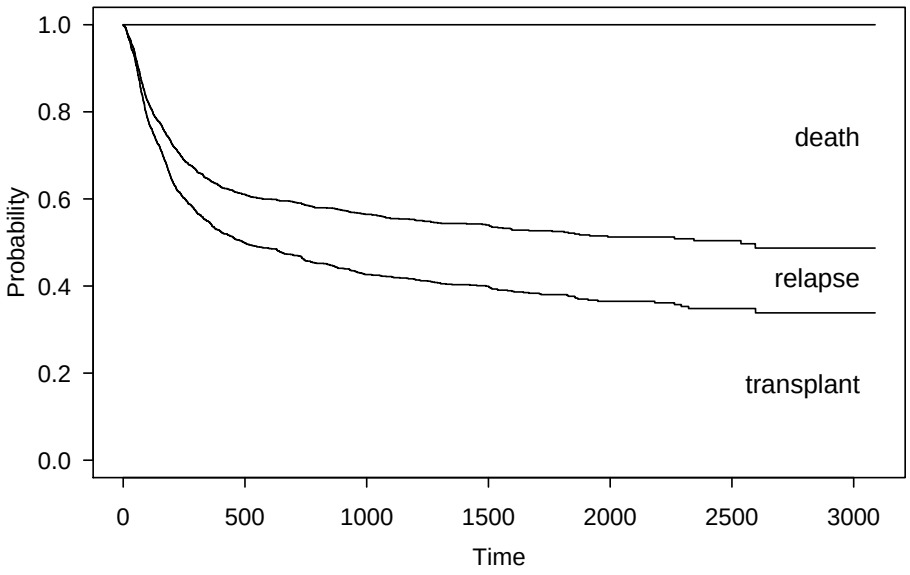

Figure 15: State occupation probabilities (`mstate` package).

- What is the relation between the three curves in the first figure.
  They add up to the value 1.
- Why can the curves in the second figure add up to a number larger than one?
  The transition probability from relapse to death does not have the same denominator as the other two. It is a conditional probability, given that a relapse has occurred. Hence, it has no relation to the other probabilities.
- Which of the curves in the second figure can also decrease over time? Why?
  The probability to be in the relapse state, because it is an intermediate state that can also be left.
- Why is the transition probability much higher from transplantation to death than from transplantation to relapse, whereas this is not seen for the cumulative hazards?
  Individuals can also leave the relapse state, which makes the curve go down. In fact, once they leave the relapse state, they contribute to the transition probability from transplant to death, namely through the indirect path via relapse. In contrast, the cumulative hazard to death only considers the direct transition.

6. Time from transplantation to death For the transition from transplantation to death, we can also use the classical Kaplan-Meier.

Compare the estimated distribution of time from transplantation to death using the classical Kaplan-Meier and using the multi-state Aalen-Johansen estimator. Also plot and compare the 95% confidence intervals.

We first compute the Kaplan-Meier

```
> KM.death <- survfit(Surv(srv,srvstat)~1, data=ebmt1)
```

The Kaplan-Meier is plotted first, the Aalen-Johansen estimate is added.

etm The Aalen-Johansen estimate can be added via the `lines.etm` function and we obtain Figure 16.

```
> plot(KM.death,fun="event",lwd=2,mark.time=FALSE)
> lines(Prob.etm, tr.choice=c("transplant death"), lwd=2, conf.int=TRUE,
+                                                  ci.fun="log", col="red")
```

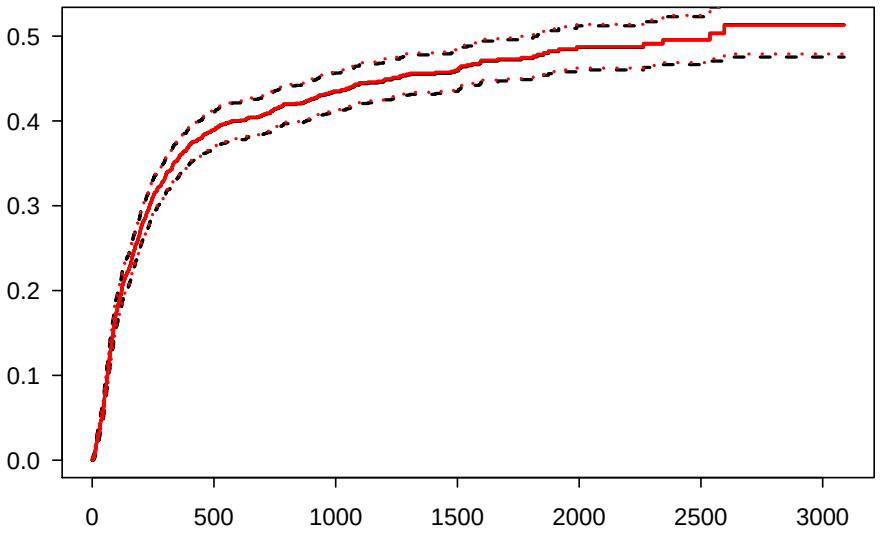

Figure 16: Time from transplantation to death. Black: Kaplan-Meier; red: Aalen-Johansen.

msSurv We first create a data frame with the Aalen-Johansen estimates:

```
> Prob.1to3 <- data.frame(time=et(Prob.msSurv), prob=AJs(Prob.msSurv)[1,3,],
+                         var=cov.AJs(Prob.msSurv)["1 3","1 3",])
> Prob.1to3$lower.ci <- pmax(0,1-(1-Prob.1to3$prob)*
+        exp(qnorm(0.975)*sqrt(Prob.1to3$var)/(1-Prob.1to3$prob)))
> Prob.1to3$upper.ci <- 1-(1-Prob.1to3$prob)*
+        exp(-qnorm(0.975)*sqrt(Prob.1to3$var)/(1-Prob.1to3$prob))
```

Figure 16 is created via

```
> plot(KM.death,fun="event",lwd=2,mark.time=FALSE)
> lines(prob~time, data=Prob.1to3, type="s", col="red")
> lines(lower.ci~time, data=Prob.1to3, type="s", lty=3, col="red")
> lines(upper.ci~time, data=Prob.1to3, type="s", lty=3, col="red")
```

---

mstate We first create a data frame with the Aalen-Johansen estimates:

```
> Prob.1to3 <- data.frame(time=Prob.mst[[1]]$time,
+              prob=Prob.mst[[1]]$pstate3, se=Prob.mst[[1]]$se3)
> Prob.1to3$lower.ci <- pmax(0,1-(1-Prob.1to3$prob)*
+             exp(qnorm(0.975)*Prob.1to3$se/(1-Prob.1to3$prob)))
> Prob.1to3$upper.ci <- 1-(1-Prob.1to3$prob)*
+               exp(-qnorm(0.975)*Prob.1to3$se/(1-Prob.1to3$prob))
```

Figure 16 is created via

```
> plot(KM.death,fun="event",lwd=2,mark.time=FALSE)
> lines(prob~time, data=Prob.1to3, type="s", col="red")
> lines(lower.ci~time, data=Prob.1to3, type="s", lty=3, col="red")
> lines(upper.ci~time, data=Prob.1to3, type="s", lty=3, col="red")
```

We observe that both approaches give almost identical estimates.

Chapter 4. Regression. Cause-Specific/Transition Hazard.

Competing risks analysis

We quantify the effect of the EBMT score on the cause-specific hazards for relapse and death-without-relapse.

1. Separate analyses We first perform the analyses separately per end point.

> **Quantify the effect of the EBMT score on both events by fitting two separate cause-specific proportional hazards models. Does a higher value of EBMT risk score increase the cause-specific hazards? Are the effects of EBMT score comparable for both event types?**

A proportional hazards model on the relapse-specific hazard can be fitted by censoring individuals when they experience the competing event, i.e. death. Individuals with `stat` equal to 1 had a relapse, those with `stat` equal to 2 are censored. Similarly for death: individuals with `stat` equal to 2 died without relapse; all others are censored. We use a logical statement to specify this event information and use the `print` function to summarize the results.

```
> fit.rel <- coxph(Surv(time,stat==1)~score, data=ebmt1)
> fit.rel

Call:
coxph(formula = Surv(time, stat == 1) ~ score, data = ebmt1)


                  coef exp(coef) se(coef)    z       p
scoreMedium risk 0.432     1.540    0.126 3.44 0.00058
scoreHigh risk   1.085     2.961    0.183 5.93 2.9e-09

Likelihood ratio test=33  on 2 df, p=6.73e-08
n= 1977, number of events= 456

> fit.death <- coxph(Surv(time,stat==2)~score, data=ebmt1)
> fit.death

Call:
coxph(formula = Surv(time, stat == 2) ~ score, data = ebmt1)


                  coef exp(coef) se(coef)    z       p
scoreMedium risk 0.639     1.895    0.113 5.64 1.7e-08
scoreHigh risk   1.178     3.249    0.153 7.68 1.6e-14

Likelihood ratio test=62.9  on 2 df, p=2.19e-14
n= 1977, number of events= 685
```

A higher value of the EBMT risk score increases both cause-specific hazards. Its effect on mortality seems to be a somewhat larger, but is this effect significant? This can only be answered in a combined analysis, as we do in Exercise 4.10.3.

2. Combined analysis We combine the effect of the EBMT risk score on both events in one analysis. We check whether results coincide with the ones from the separate analyses.

> **Repeat the analyses, but now by fitting one proportional cause-specific hazards model for both competing risks at once. You can use the stacked data set `Webmt1`, which was created via the `crprep` function on page 9.**
>
> **Fit the model in three ways**

1. **By including an interaction between the EBMT score and the type of end point**
2. **By creating a single compound variable that combines the EBMT risk score and the type of end point**
3. **By creating type-specific covariables**

Do all approaches give the same estimates? If not, can you translate one output into the other? Compare the estimates with those from the separate analyses.

Test whether the proportional hazards assumption is reasonable.

When using the `Webmt1` data set, we need to restrict to the rows that correspond to the periods in which the individuals were observed and in follow-up, i.e. the rows with `count` equal to 1. We specify an interaction between the EBMT score and the event type

```
> coxph(Surv(Tstop,status==failcode)~strata(failcode)*score, data=Webmt1,
+                                                          subset=count==1)

Call:
coxph(formula = Surv(Tstop, status == failcode) ~ strata(failcode) *
    score, data = Webmt1, subset = count == 1)


                                              coef exp(coef) se(coef)    z       p
scoreMedium risk                            0.4319    1.5401   0.1255 3.44 0.00058
scoreHigh risk                              1.0855    2.9609   0.1829 5.93 2.9e-09
strata(failcode)failcode=2:scoreMedium risk 0.2074    1.2305   0.1691 1.23 0.21997
strata(failcode)failcode=2:scoreHigh risk   0.0927    1.0972   0.2387 0.39 0.69764

Likelihood ratio test=95.9  on 4 df, p=0
n= 3954, number of events= 1141
```

Since the EBMT score is a categorical variable, we can also perform the analysis via a compound score variable. We allow for separate baseline hazards for each event type, which are the hazards for the reference value of the EBMT score (Low risk). Therefore, we only need one level for the reference Low risk group. We create four levels that give the contrasts of Medium risk and High risk to Low risk for each of the two event types. An efficient, although maybe not very insightful way to create the compound score variable is via

```
> Webmt1$score.comb <- with(Webmt1, factor( (as.numeric(score)-1)*(failcode==1)+
+                                          3*(as.numeric(score)-1)*(failcode==2),
+           labels=c("Low","Medium.Rel","High.Rel","Medium.Death","High.Death"))
+        )
```

We write

```
> coxph(Surv(Tstop,status==failcode)~strata(failcode)+ score.comb, data=Webmt1,
+                                                              subset=count==1)

Call:
coxph(formula = Surv(Tstop, status == failcode) ~ strata(failcode) +
    score.comb, data = Webmt1, subset = count == 1)


                        coef exp(coef) se(coef)    z       p
score.combMedium.Rel   0.432     1.540    0.126 3.44 0.00058
score.combHigh.Rel     1.085     2.961    0.183 5.93 2.9e-09
score.combMedium.Death 0.639     1.895    0.113 5.64 1.7e-08
score.combHigh.Death   1.178     3.249    0.153 7.68 1.6e-14
```

```
Likelihood ratio test=95.9  on 4 df, p=0
n= 3954, number of events= 1141
```

The value of the likelihood ratio test and the first two parameter estimates are exactly equal to the specification with the interaction term, but the last two parameter estimates are different. These estimates and p-values are equal to the results from the analyses per event type in the previous exercise.

The specification with the interaction term gives the *additional* effect of Medium risk and High risk for death as event type, relative to their effects on relapse: $0.4319 + 0.2074 = 0.6393$ and $1.0855 + 0.0927 = 1.1782$.

When fitting the model with the combined score variable, the values in `score.comb` are internally transformed into four columns with dummy variables. These are the type-specific variables. We can also create the type-specific variables ourselves, e.g. via the `expand.covs` function in the `mstate` package

```
> Webmt1 <- expand.covs(Webmt1, covs="score", longnames=TRUE)
```

If we create them using basic Rcode, we have more control over the naming of the columns:

```
> Webmt1$Medium.Rel <- with(Webmt1, ifelse(score=="Medium risk"&failcode==1, 1, 0))
> Webmt1$High.Rel <- with(Webmt1, ifelse(score=="High risk"&failcode==1, 1, 0))
> Webmt1$Medium.Mort <- with(Webmt1, ifelse(score=="Medium risk"&failcode==2, 1, 0))
> Webmt1$High.Mort <- with(Webmt1, ifelse(score=="High risk"&failcode==2, 1, 0))
```

The model is fitted using these variables via

```
> fit.csh.comb <- coxph(Surv(Tstop,status==failcode)~strata(failcode)+
+           Medium.Rel+High.Rel+Medium.Mort+High.Mort, data=Webmt1, subset=count==1)
```

Apart from the naming of the variables, the output is the same as when the compound variable is used. We store this output in `fit.csh.comb` for later use.

We use the `cox.zph` function to test for proportionality

```
> cox.zph(fit.csh.comb)
```

```
                rho chisq      p
Medium.Rel  -0.0743 6.205 0.0127
High.Rel    -0.0613 4.183 0.0408
Medium.Mort  0.0208 0.493 0.4826
High.Mort    0.0145 0.241 0.6233
GLOBAL           NA 7.329 0.1195
```

Although the overall test is not significant, there is some suggestion of non-proportionality in the effect of the risk score on relapse specific hazard. One may be inclined to think that this was also observable from Figure 7. However, this is not necessarily the case in the competing risks setting. In Chapter 5 of the book, it is shown that curves for the cumulative incidence can cross, even though the cause-specific hazards are proportional.

3. Test for equality of effects We formally test whether the effects of Medium risk and High risk on both event types are comparable.

> **Test the null hypothesis that the hazard of medium risk, relative to low risk, is the same for both event types. And similarly for high risk, relative to low risk.**
>
> **Perform the likelihood ratio test of the null hypothesis that the effect of both medium and high risk is the same for both event types.**

From the specification with the interaction term in the previous exercise we could see that both the effect of medium risk and the effect of high risk on death were not significantly different from the corresponding effects on relapse. We can obtain the same result via

```
> coxph(Surv(Tstop,status==failcode)~strata(failcode)+score+Medium.Mort+High.Mort,
+                                                  data=Webmt1, subset=count==1)
```

Before comparing each level of the EBMT score separately, it is better to first perform an overall test of equality. We perform a likelihood ratio test, comparing the above model with the one in which the effect of the EBMT score is assumed to be equal for both transitions.

```
> fit.csh.eq <- coxph(Surv(Tstop,status==failcode)~strata(failcode)+ score,
+                                                  data=Webmt1, subset=count==1)
> anova(fit.csh.eq, fit.csh.comb)

Analysis of Deviance Table
 Cox model: response is  Surv(Tstop, status == failcode)
 Model 1: ~ strata(failcode) + score
 Model 2: ~ strata(failcode) + Medium.Rel + High.Rel + Medium.Mort + High.Mort
  loglik Chisq Df P(>|Chi|)
1  -8080
2  -8079  1.67  2      0.43
```

Hence there is no reason to conclude that the effect of the EBMT score on both competing events is different.

In Section 4.3.3 of the book we explained that another way to specify a model with equal effects of risk score is

```
> coxph(Surv(time,stat>=1)~score, data=ebmt1)

Call:
coxph(formula = Surv(time, stat >= 1) ~ score, data = ebmt1)


                   coef exp(coef) se(coef)    z       p
scoreMedium risk 0.550     1.733    0.084 6.55 5.9e-11
scoreHigh risk   1.133     3.105    0.117 9.67 < 2e-16

Likelihood ratio test=94.4  on 2 df, p=0
n= 1977, number of events= 1141
```

Since it uses a different data structure, it has a different likelihood and therefore cannot be used in the likelihood ratio test.

Multi-state analysis

We continue with the illness-death multi-state model that was introduced in Chapter 3. We additionally investigate the effect of the EBMT risk score on death after relapse. Furthermore, we investigate whether the effect of relapse on death has changed over time.

4. Create and inspect the stacked data set

> Create the stacked data set that is needed for the analyses. Include the variables EBMT score (`score`) and year of relapse (`yrel`). Store the stacked data set in an object named `msebmt`. Compare the data of the first two individuals with the information in the `ebmt1` data set.

We use the same `msprep` function as on Page 21, but additionally specify the *keep* argument.

```
> tmat.mst <- trans.illdeath(names=c("T","R","D"))
> msebmt <- msprep(time=c(NA,"rel","srv"), status=c(NA,"relstat","srvstat"),
+                  data=ebmt1, trans=tmat.mst, keep=c("score","yrel"))
```

We check whether the columns `score` and `yrel` have been added by looking at the first two patients.

```
> subset(msebmt, id %in% c(1,2))
```

```
An object of class 'msdata'

Data:
  id from to trans Tstart Tstop time status       score      yrel
1  1    1  2     1      0  1610 1610      0 Medium risk      <NA>
2  1    1  3     2      0  1610 1610      0 Medium risk      <NA>
3  2    1  2     1      0   165  165      1 Medium risk 1997-1999
4  2    1  3     2      0   165  165      0 Medium risk 1997-1999
5  2    2  3     3    165   961  796      1 Medium risk 1997-1999
```

Count the number of transitions between the different states, using the `events` function. Do the numbers correspond with the number of competing events and the overall number of deaths if you make the summary yourself based on the data in `ebmt1`? What additional information does the `events` function provide?

For the summary based on `ebmt1` we use the `table` function.

```
> events(msebmt)

$Frequencies
    to
from    T    R    D no event total entering
   T    0  456  685      836           1977
   R    0    0  187      269            456
   D    0    0    0        0              0

$Proportions
    to
from      T      R      D no event
   T 0.0000 0.2307 0.3465   0.4229
   R 0.0000 0.0000 0.4101   0.5899
   D

> table(ebmt1$stat)


  0   1   2
836 456 685

> table(ebmt1$srvstat)


   0    1
1105  872
```

Both give the frequencies, but in a slightly different way. The tabulation of the `stat` variable and the first row in the `Frequencies` component of the `events` output give the number of competing events from state "T" (transplant) as well as the number of individuals that had no event observed. The values in the `srvstat` tabulation are equal to the sum of the values in the columns called "no event" and "D" respectively. The `events` function additionally reports the proportions.

5. Create transition-specific covariables

Create the transition-specific covariables for the EBMT score and year of relapse using the `expand.covs` function. Have a look at the resulting data set. Study the values of the transition-specific covariables in relation to the basic covariables and the transitions.

```
> msebmt.tmp <- expand.covs(msebmt, c("score","yrel"))
```

By default the transition-specific covariables will have names with suffix ".1", ".2", ..., ".K":

```
> head(msebmt.tmp)

An object of class 'msdata'

Data:
  id from to trans Tstart Tstop time status       score      yrel scoreMedium.risk.1
1  1    1  2     1      0  1610 1610      0 Medium risk      <NA>                   1
2  1    1  3     2      0  1610 1610      0 Medium risk      <NA>                   0
3  2    1  2     1      0   165  165      1 Medium risk 1997-1999                   1
4  2    1  3     2      0   165  165      0 Medium risk 1997-1999                   0
5  2    2  3     3    165   961  796      1 Medium risk 1997-1999                   0
6  3    1  2     1      0  1508 1508      0 Medium risk      <NA>                   1
  scoreMedium.risk.2 scoreMedium.risk.3 scoreHigh.risk.1 scoreHigh.risk.2
1                  0                  0                0                0
2                  1                  0                0                0
3                  0                  0                0                0
4                  1                  0                0                0
5                  0                  1                0                0
6                  0                  0                0                0
  scoreHigh.risk.3 yrel1997.1999.1 yrel1997.1999.2 yrel1997.1999.3 yrel2000..1
1                0              NA              NA              NA          NA
2                0              NA              NA              NA          NA
3                0               1               0               0           0
4                0               0               1               0           0
5                0               0               0               1           0
6                0              NA              NA              NA          NA
  yrel2000..2 yrel2000..3
1          NA          NA
2          NA          NA
3           0           0
4           0           0
5           0           0
6          NA          NA
```

These names of the transition-specific covariables are overly long. Often it is preferable to replace the factor levels in the column names by numerical values, which can be done via the *longnames* argument of expand.covs. A disadvantage is that the user needs to remember what these numbers represent.

Create the transition-specific covariables using the short variable naming and add them to the msebmt object. Compare this with the previous naming. What does for instance score2.3 represent?

```
> msebmt <- expand.covs(msebmt, c("score","yrel"), longnames=FALSE)
```

We have a look at the first couple of rows again.

```
> head(msebmt)

An object of class 'msdata'

Data:
  id from to trans Tstart Tstop time status       score      yrel score1.1 score1.2
1  1    1  2     1      0  1610 1610      0 Medium risk      <NA>        1        0
2  1    1  3     2      0  1610 1610      0 Medium risk      <NA>        0        1
3  2    1  2     1      0   165  165      1 Medium risk 1997-1999        1        0
4  2    1  3     2      0   165  165      0 Medium risk 1997-1999        0        1
```

```
5  2    2  3     3    165   961  796      1 Medium risk 1997-1999        0        0
6  3    1  2     1      0  1508 1508      0 Medium risk      <NA>        1        0
  score1.3 score2.1 score2.2 score2.3 yrel1.1 yrel1.2 yrel1.3 yrel2.1 yrel2.2
1        0        0        0        0      NA      NA      NA      NA      NA
2        0        0        0        0      NA      NA      NA      NA      NA
3        0        0        0        0       1       0       0       0       0
4        0        0        0        0       0       1       0       0       0
5        1        0        0        0       0       0       1       0       0
6        0        0        0        0      NA      NA      NA      NA      NA
  yrel2.3
1      NA
2      NA
3       0
4       0
5       0
6      NA
```

`score2.3` is the effect of high risk (second non-reference level) on transition 3. In the long expansion, it was called `scoreHigh.risk.3`.

The effect of `yrel` is only relevant for transition 3; why?

Year of relapse is only relevant after a relapse has happened. So as not to be tempted to use it for other transitions, we delete the others.

```
> msebmt <- msebmt[,-grep("yrel..[1-2]",names(msebmt))]
```

6. General model

Use the transition-specific covariables to estimate the effect of the EBMT score on all transitions, allowing this effect to differ by transition.

Compare the hazard ratios of `score` for transitions 1 and 2 with those of the competing risks analyses in practicals 1. and 2. of this section. What do you notice? Do you have an explanation?

Similar to the competing risks setting, we allow for a separate baseline hazard per transition by including `trans` as stratum variable. We generate the more extended summary via the `summary` function.

```
> fit.ebmt1 <- coxph(Surv(Tstart,Tstop,status) ~ score1.1 + score2.1 +
+     score1.2 + score2.2 + score1.3 + score2.3 + strata(trans), data=msebmt)
> summary(fit.ebmt1)

Call:
coxph(formula = Surv(Tstart, Tstop, status) ~ score1.1 + score2.1 +
    score1.2 + score2.2 + score1.3 + score2.3 + strata(trans),
    data = msebmt)

  n= 4410, number of events= 1328

           coef exp(coef) se(coef)    z Pr(>|z|)
score1.1 0.432     1.540    0.126 3.44  0.00058 ***
score2.1 1.085     2.961    0.183 5.93  2.9e-09 ***
score1.2 0.639     1.895    0.113 5.64  1.7e-08 ***
score2.2 1.178     3.249    0.153 7.68  1.6e-14 ***
score1.3 0.542     1.720    0.260 2.08  0.03707 *
score2.3 1.473     4.362    0.301 4.90  9.6e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
         exp(coef) exp(-coef) lower .95 upper .95
score1.1      1.54      0.649      1.20      1.97
score2.1      2.96      0.338      2.07      4.24
score1.2      1.90      0.528      1.52      2.37
score2.2      3.25      0.308      2.41      4.39
score1.3      1.72      0.581      1.03      2.86
score2.3      4.36      0.229      2.42      7.86

Concordance= 0.569  (se = 0.01 )
Rsquare= 0.028   (max possible= 0.983 )
Likelihood ratio test= 124  on 6 df,   p=0
Wald test            = 127  on 6 df,   p=0
Score (logrank) test = 135  on 6 df,   p=0
```

The hazard ratios and p-values for transitions 1 and 2 are exactly the same as in the competing risks analysis. This is not surprising because we are basically fitting three separate models for the transition hazard, one for each transition.

This model could be obtained as well by fitting separate Cox regression models for each transition. How would you do this? You don't have to perform the actual computations.

The stacked format of the data set `msebmt` contains the relevant information for each transition separately; in fact, it can be seen as being obtained by stacking the data sets as needed for the separate analyses. For example, we could fit the model for transition 1 via

```
> coxph(Surv(Tstart,Tstop,status) ~ score, data=msebmt, subset=(trans==1))

Call:
coxph(formula = Surv(Tstart, Tstop, status) ~ score, data = msebmt,
    subset = (trans == 1))


                  coef exp(coef) se(coef)    z       p
scoreMedium risk 0.432     1.540    0.126 3.44 0.00058
scoreHigh risk   1.085     2.961    0.183 5.93 2.9e-09

Likelihood ratio test=33  on 2 df, p=6.73e-08
n= 1977, number of events= 456
```

Note that we do not need to use the transition-specific covariables here because we restrict to a single transition.

7. Same hazard ratios? We already noticed that the effect of the EBMT score was similar for the competing risks relapse and death. Now we additionally consider the third transition.

Looking at the hazard ratios of the medium EBMT score across the different transitions, would you say they are similar? And what about the hazard ratios for a high EBMT score? Perform the likelihood ratio test to see whether the hazard ratios for both the medium and high score may be considered equal across the transitions.

They seem to be fairly similar; the confidence intervals of `score1.1`, `score1.2` and `score1.3` overlap; the same holds for the other three parameters. We investigate this more formally using the likelihood ratio test. In order to perform this test, we first need to fit another model in which the effect of the EBMT score is assumed equal over all transitions, still allowing for separate baseline hazards. Then we use the `anova` function.

```
> fit.ebmt2 <- coxph(Surv(Tstart,Tstop,status) ~ score + strata(trans), data=msebmt)
> fit.ebmt2
```

```
Call:
coxph(formula = Surv(Tstart, Tstop, status) ~ score + strata(trans),
    data = msebmt)


                   coef exp(coef) se(coef)     z       p
scoreMedium risk 0.5459    1.7262   0.0799  6.83 8.4e-12
scoreHigh risk   1.1924    3.2949   0.1076 11.08 < 2e-16

Likelihood ratio test=120  on 2 df, p=0
n= 4410, number of events= 1328
```

```
> anova(fit.ebmt2,fit.ebmt1)

Analysis of Deviance Table
 Cox model: response is  Surv(Tstart, Tstop, status)
 Model 1: ~ score + strata(trans)
 Model 2: ~ score1.1 + score2.1 + score1.2 + score2.2 + score1.3 + score2.3 + strata(t
  loglik Chisq Df P(>|Chi|)
1  -8976
2  -8974  4.28  4      0.37
```

The p-value gives little reason to conclude that the effects are different.

8. Proportional baseline hazards Based on the likelihood ratio test result, we continue with a model in which the effect of the EBMT score is assumed to be equal across the transitions.

In order to quantify how much the death rate changes after a relapse, we will now look at a model where the baseline transition hazards into death are assumed to be proportional. Thus, the transitions 2 and 3 are assumed to share a common baseline.

> **Create a covariable `rel.surv` that allows you to fit a model in which the baseline hazards to death are proportional. Fit the model and have a look at the results. What does the hazard ratio of `rel.srv` imply? Have the hazard ratios of `score` (or their standard errors) changed compared to the model in which we had a separate baseline hazard per transition? Test whether proportionality for the baseline hazards to death is a reasonable assumption.**
>
> **Compare the estimate of the standard error with a robust sandwich-type estimate.**

We define a time-dependent covariable `rel.surv` which is 0 if no relapse has occurred and 1 if a relapse has occurred. It can be created via

```
> msebmt$rel.srv <- 0
> msebmt$rel.srv[msebmt$trans==3] <- 1
```

We check the new variable in the data set:

```
> head(msebmt)[,c(1:9,19)]

An object of class 'msdata'

Data:
  id from to trans Tstart Tstop time status       score rel.srv
1  1    1  2     1      0  1610 1610      0 Medium risk       0
2  1    1  3     2      0  1610 1610      0 Medium risk       0
3  2    1  2     1      0   165  165      1 Medium risk       0
4  2    1  3     2      0   165  165      0 Medium risk       0
5  2    2  3     3    165   961  796      1 Medium risk       1
6  3    1  2     1      0  1508 1508      0 Medium risk       0
```

The model can be fitted using a stratified Cox regression model with strata defined by the receiving state (`to` in the stacked data set), and with `rel.srv` as additional covariable.

```
> fit.ebmt3 <- coxph(Surv(Tstart,Tstop,status) ~ score + rel.srv +
+                                          strata(to), data=msebmt)
> fit.ebmt3

Call:
coxph(formula = Surv(Tstart, Tstop, status) ~ score + rel.srv +
    strata(to), data = msebmt)


                   coef exp(coef) se(coef)     z      p
scoreMedium risk 0.5451    1.7248   0.0799  6.82  9e-12
scoreHigh risk   1.1920    3.2937   0.1076 11.08 <2e-16
rel.srv          1.0549    2.8718   0.0893 11.82 <2e-16

Likelihood ratio test=256  on 3 df, p=0
n= 4410, number of events= 1328
```

`rel.srv` measures the effect of relapse on survival. We see that mortality is almost three times higher after relapse. The other covariables and their standard errrors do not change much. This is not surprising if proportionality is in correspondence with the data.

Indeed, proportionality is a reasonable assumption (p-value 0.3267):

```
> cox.zph(fit.ebmt3)

                     rho chisq     p
scoreMedium risk -0.0346 1.584 0.208
scoreHigh risk   -0.0165 0.357 0.550
rel.srv           0.0273 0.962 0.327
GLOBAL                NA 2.500 0.475
```

We can also compare the estimates of the baseline hazards from both models visually. In Figure 17 we plot the estimated cumulative transition hazards to death for the group with the low risk score. For this, we can apply the `survfit` function to the output of the `coxph` function (see also Section 5.7.1 of the book). `survfit` computes the curves for every value of the stratum variable; we select the components of interest using square brackets.

Using a robust estimate of the standard error gives almost the same results:

```
> coxph(Surv(Tstart,Tstop,status) ~ score + rel.srv +
+                            strata(to) + cluster(id), data=msebmt)

Call:
coxph(formula = Surv(Tstart, Tstop, status) ~ score + rel.srv +
    strata(to) + cluster(id), data = msebmt)


                   coef exp(coef) se(coef) robust se     z       p
scoreMedium risk 0.5451    1.7248   0.0799    0.0788  6.92 4.6e-12
scoreHigh risk   1.1920    3.2937   0.1076    0.1082 11.01 < 2e-16
rel.srv          1.0549    2.8718   0.0893    0.0911 11.58 < 2e-16

Likelihood ratio test=256  on 3 df, p=0
n= 4410, number of events= 1328
```

```
> plot(survfit(fit.ebmt2, newdata=data.frame(score="Low risk"))[c(2,3)], fun="cumhaz",
+                                                  mark.time=FALSE,lwd=2)
> lines(survfit(fit.ebmt3, newdata=data.frame(score="Low risk",rel.srv=0))[2],
+        fun="cumhaz",  mark.time=FALSE, col="red", conf.int=FALSE,lwd=2)
> lines(survfit(fit.ebmt3, newdata=data.frame(score="Low risk",rel.srv=1))[2],
+              fun="cumhaz", mark.time=FALSE, col="red", conf.int=FALSE,lwd=2)
```

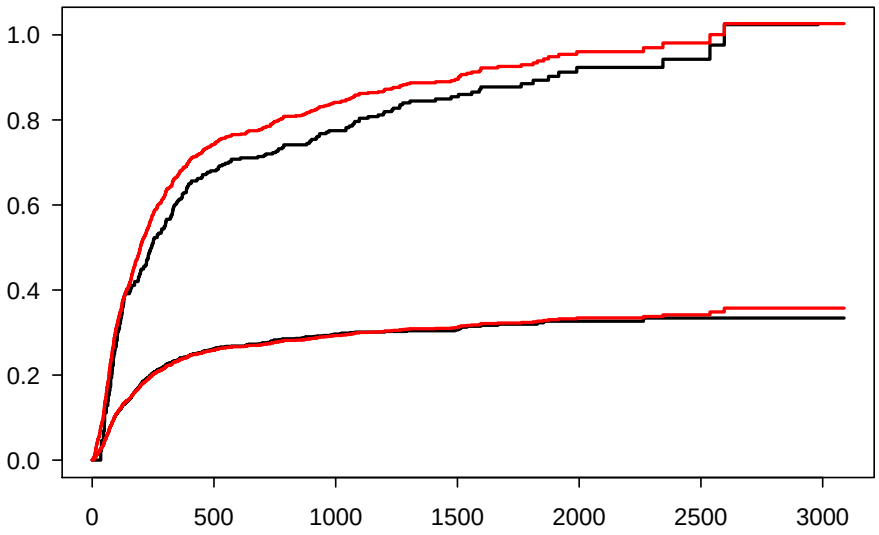

Figure 17: Comparison of cumulative hazards to death with and without assuming proportionality. Black: separate baseline hazard per transition; red: transitions into death have same baseline hazard. Upper lines: transition from relapse to death. Lower lines: transition from transplant to death.

9. Time trend in relapse? Now we are ready to study whether the effect of a relapse on mortality has changed over time. We will start from the model in the previous exercise with proportional baseline hazards for the two transitions into death.

> **For transition 3, `yrel` is of importance, but the rows that refer to transitions 1 and 2 have missing values. Rows with missing values are removed when fitting a Cox model, which is not what we want. How do you solve this problem?**

We replace the missing values by 0 in the transition-specific variables `yrel1.3` and `yrel2.3`. These are dummy variables that represent the impacts of the periods 1997-1999 and 2000- on the transition from relapse to death, relative to 1993-1996. We check the resulting data set.

```
> msebmt$yrel1.3[is.na(msebmt$yrel1.3)] <- 0
> msebmt$yrel2.3[is.na(msebmt$yrel2.3)] <- 0
> head(msebmt)[,c(1:8,10,17:18)]

An object of class 'msdata'

Data:
  id from to trans Tstart Tstop time status      yrel yrel1.3 yrel2.3
1  1    1  2     1      0  1610 1610      0      <NA>       0       0
2  1    1  3     2      0  1610 1610      0      <NA>       0       0
3  2    1  2     1      0   165  165      1 1997-1999       0       0
4  2    1  3     2      0   165  165      0 1997-1999       0       0
```

```
5  2    2  3     3    165   961  796       1 1997-1999        1        0
6  3    1  2     1      0  1508 1508       0       <NA>        0        0
```

Add period of relapse to the model. What is your conclusion with regard to the impact of period of relapse? Has the hazard ratio of `rel.srv` changed? Has the *meaning* of this hazard ratio changed?

```
> fit.ebmt4 <- coxph(Surv(Tstart,Tstop,status) ~ score + yrel1.3 +
+       yrel2.3 + rel.srv + strata(to), data=msebmt)
> fit.ebmt4

Call:
coxph(formula = Surv(Tstart, Tstop, status) ~ score + yrel1.3 +
    yrel2.3 + rel.srv + strata(to), data = msebmt)


                   coef exp(coef) se(coef)     z       p
scoreMedium risk  0.542     1.720    0.080  6.78 1.2e-11
scoreHigh risk    1.202     3.326    0.108 11.17 < 2e-16
yrel1.3          -0.163     0.850    0.164 -0.99 0.32022
yrel2.3          -0.844     0.430    0.240 -3.52 0.00043
rel.srv           1.300     3.670    0.140  9.28 < 2e-16

Likelihood ratio test=271  on 5 df, p=0
n= 4410, number of events= 1328
```

We see that the hazard is decreased in calendar period 1997-1999 (`yrel1.3`) and even more so in calendar period 2000- (`yrel1.3`). The value of `rel.srv` has increased. This is because it now refers to the relative transition hazard for the reference period, 1993-1996, which is the period with the highest mortality after relapse.

What is the hazard ratio of the transition into death after relapse in 2000+ with respect to the transition into death without relapse? Does a relapse nowadays (after 2000) have a (statistically) significant impact on the transition rate to death?

If we want to know whether the transition rate from relapse to death still differs in the latest calendar period, we can add the fourth and fifth parameter to obtain 0.4565. If we want to test whether this value is different from zero, it is easiest to redefine the `yrel2.3` variable such that it refers to the period 1993-1996 relative to 2000+.

```
> tmp <- msebmt
> tmp <- within(tmp,{
+     yrel2.3[!is.na(yrel)&yrel=="1993-1996"&trans==3] <- 1
+     yrel2.3[!is.na(yrel)&yrel=="2000-"] <- 0
+    })
> coxph(Surv(Tstart,Tstop,status) ~ score + yrel1.3 + yrel2.3 + rel.srv
+                                          + strata(to), data=tmp)

Call:
coxph(formula = Surv(Tstart, Tstop, status) ~ score + yrel1.3 +
    yrel2.3 + rel.srv + strata(to), data = tmp)


                  coef exp(coef) se(coef)     z       p
scoreMedium risk 0.542     1.720    0.080  6.78 1.2e-11
scoreHigh risk   1.202     3.326    0.108 11.17 < 2e-16
yrel1.3          0.681     1.975    0.224  3.04 0.00235
yrel2.3          0.844     2.325    0.240  3.52 0.00043
```

```
rel.srv          0.457     1.579    0.206  2.21 0.02694

Likelihood ratio test=271  on 5 df, p=0
n= 4410, number of events= 1328
```

We see that the effect of `rel.surv` is still significant.

Chapter 5. Regression; Translation to Cumulative Scale

Competing risks analysis

We compute the cause-specific cumulative incidence of both relapse and death-before-relapse for each value of EBMT score. We use two models: the proportional cause-specific hazards and the proportional subdistribution hazards model. In both approaches, we allow the effect of EBMT score to differ by event type and we assume separate baseline hazards.

1. From cause-specific to cumulative We plug the estimated cause-specific hazards into the Aalen-Johansen form of the cause-specific cumulative incidence.

Perform the calculations based on the model for the cause-specific hazards from Section 4.10.

We use the `mstate` package. We need to define the transition matrix for the competing risks setting, fit the cause-specific hazards model, create the data set that has the specific covariable combination and apply the functions `msfit` and `probtrans`.

We can use the cause-specific regression model `fit.csh.comb` that we fitted in Section . We need to define the transition matrix and create three data sets, one for each value of the EBMT score. This is most efficiently done via a list structure, such that we can use a for loop when applying the `msfit` function.

```
> trans.CR <- trans.comprisk(2,c("transplant","relapse","death"))
> ndata <- list(3)
> ndata[[1]] <- data.frame(Medium.Rel=c(0,0), High.Rel=c(0,0),
+                   Medium.Mort=c(0,0),High.Mort=c(0,0),strata=1:2)
> ndata[[2]] <- data.frame(Medium.Rel=c(1,0), High.Rel=c(0,0),
+                   Medium.Mort=c(0,1),High.Mort=c(0,0),strata=1:2)
> ndata[[3]] <- data.frame(Medium.Rel=c(0,0), High.Rel=c(1,0),
+                   Medium.Mort=c(0,0),High.Mort=c(0,1),strata=1:2)
```

The computation of the cumulative cause-specific hazard and cause-specific cumulative incidence goes in the same way as in Section .

```
> HvH <- list(3)
> for(i in 1:3) HvH[[i]] <- msfit(fit.csh.comb, newdata=ndata[[i]], trans=trans.CR)
> pt <- list(3)
> for(i in 1:3) pt[[i]] <- probtrans(HvH[[i]], predt=0)
```

An alternative is to start with the original `ebmt1` data set and repeat the procedure that we used in Chapter 4 for the illness-death model, but now for the competing risks model. When using the `msprep` function, the code is slightly shorter if we use the alternative specification by value instead of by name (see Page 138 of the book). The next step is to fit the model via the `coxph` function. We use type-specific covariables. Note that the column with the EBMT risk score that is created is called `keep1`; we change its name to `score`.

```
> trans.CR <- trans.comprisk(2,c("transplant","relapse","death"))
> ebmt.CR <- with(ebmt1, msprep(time=cbind(NA,time,time),
+     status=cbind(NA,stat==1,stat==2), trans=trans.CR, keep=score))
> names(ebmt.CR)[9] <- "score"
> ebmt.CR <- expand.covs(ebmt.CR, covs="score")
> names(ebmt.CR)[10:13] <- c("Medium.Rel","Medium.Mort","High.Rel","High.Mort")
> fit.csh.comb <- coxph(Surv(Tstop,status)~strata(trans)+
+                           Medium.Rel+High.Rel+Medium.Mort+High.Mort, data=ebmt.CR)
```

The rest is done in the same way as described above.

Make a graph that compares the predicted cause-specific cumulative incidence with the non-parametric estimates. Use the overlaid format for the three values of risk score and the separate format for the two event types. Do the estimates correspond?

We plot the predicted curves using the standard `lines` function (Figure 18). We first plot the nonparametric estimates for comparison. We use a separate panel per end point. Recall that the first component of the `probtrans` output contains the estimates for the transitions out of the initial state, which is what we need here. Note that the estimates per value of the EBMT score were stored as a list as well. Hence, when we write `pt[[1]][[1]]`, the first `[[1]]` refers to the Low risk group, whereas the second `[[1]]` refers to the transitions out of the transplantation state.

```
> tmp.col <- c("black","red","green")
> par(mfrow=c(1,2),las=1)
> plot(survfit(Surv(Tstart,Tstop,status=="Relapse")~score,
+              data=subset(Webmt.score,failcode=="Relapse"), weights=weight.cens), lwd=1,
+      mark.time=FALSE, col=tmp.col, fun="event", ylim=c(0,0.5))
> lines(pt[[1]][[1]]$time, pt[[1]][[1]]$pstate2, col="black", lwd=3, type="s")
> lines(pt[[2]][[1]]$time, pt[[2]][[1]]$pstate2, col="red", lwd=3, type="s")
> lines(pt[[3]][[1]]$time, pt[[3]][[1]]$pstate2, col="green", lwd=3, type="s")
> title("Relapse")
> plot(survfit(Surv(Tstart,Tstop,status=="Death")~score,
+              data=subset(Webmt.score,failcode=="Death"), weights=weight.cens), lwd=1,
+      mark.time=FALSE, col=tmp.col, fun="event", ylim=c(0,0.55))
> lines(pt[[1]][[1]]$time, pt[[1]][[1]]$pstate3, col="black", lwd=3, type="s")
> lines(pt[[2]][[1]]$time, pt[[2]][[1]]$pstate3, col="red", lwd=3, type="s")
> lines(pt[[3]][[1]]$time, pt[[3]][[1]]$pstate3, col="green", lwd=3, type="s")
> title("Death")
> legend("bottomright", levels(Webmt1$score), col=tmp.col, lwd=3)
```

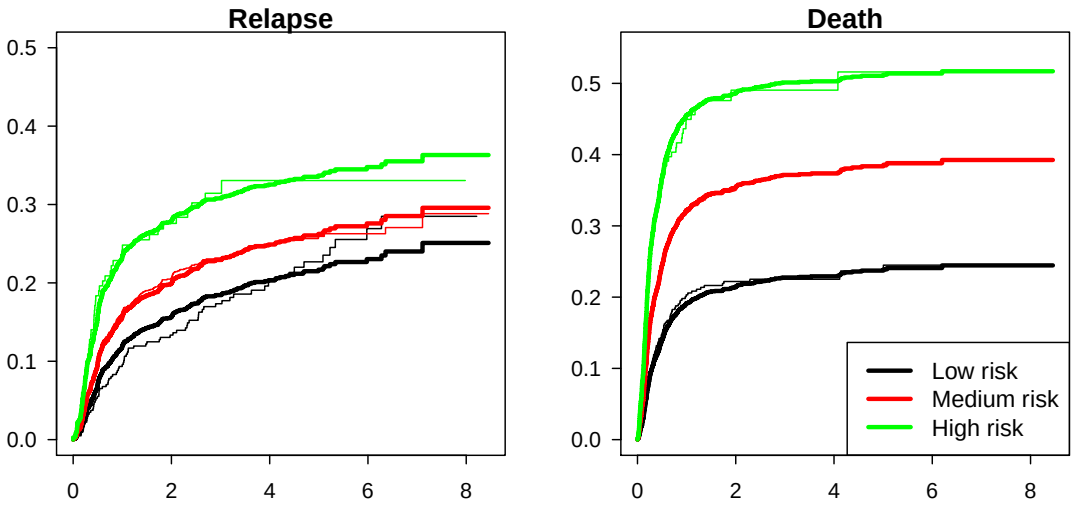

Figure 18: Predicted cause-specific cumulative incidence based on proportional cause-specific hazards model. Thin lines: nonparametric estimates.

We see that the curves based on the proportional hazards model follow the nonparametric estimates fairly closely, except with relapse as event type for the low risk group.

2. Effects on subdistribution hazard An alternative is to compute the cause-specific cumulative incidence based on a proportional subdistribution hazards model. We first look at the parameter estimates from this model.

Quantify the effect of EBMT score on the subdistribution hazards for both event types. Compare the parameter estimates with the ones from the proportional cause-specific hazards model. Can you explain the difference?

We use the score-specific weights that were stored in the `Webmt.score` object. We add the type-specific covariables, just like we did when fitting the proportional cause-specific hazards model. We can use the `expand.covs` function, which creates type-specific variables named **scoreMedium.risk.Relapse**, **scoreMedium.risk.Death**, **scoreHigh.risk.Relapse** and **scoreHigh.risk.Death**.

```
> Webmt.score <- expand.covs(Webmt.score, covs="score", longnames=TRUE)
```

An alternative is to create them ourselves, e.g. as

```
> Webmt.score$Medium.Rel <- with(Webmt.score,
+                                ifelse(score=="Medium risk"&failcode==1, 1, 0))
> Webmt.score$High.Rel <- with(Webmt.score,
+                                ifelse(score=="High risk"&failcode==1, 1, 0))
> Webmt.score$Medium.Mort <- with(Webmt.score,
+                                ifelse(score=="Medium risk"&failcode==2, 1, 0))
> Webmt.score$High.Mort <- with(Webmt.score,
+                                ifelse(score=="High risk"&failcode==2, 1, 0))
```

The basic structure of the model is the same as when fitting a proportional cause-specific hazards model. The only difference is that now individuals that experienced the competing event remain included with a weight specified via the *weights* argument.

```
> fit.sdh.comb <- coxph(Surv(Tstart,Tstop,status==failcode)~strata(failcode)+
+                           scoreMedium.risk.Relapse+scoreHigh.risk.Relapse+
+                              scoreMedium.risk.Death+scoreHigh.risk.Death,
+                                     data=Webmt.score, weights=weight.cens)
> fit.sdh.comb

Call:
coxph(formula = Surv(Tstart, Tstop, status == failcode) ~ strata(failcode) +
    scoreMedium.risk.Relapse + scoreHigh.risk.Relapse + scoreMedium.risk.Death +
    scoreHigh.risk.Death, data = Webmt.score, weights = weight.cens)


                          coef exp(coef) se(coef)    z       p
scoreMedium.risk.Relapse 0.244     1.276    0.125 1.95 0.05154
scoreHigh.risk.Relapse   0.661     1.937    0.182 3.62 0.00029
scoreMedium.risk.Death   0.567     1.763    0.113 5.01 5.5e-07
scoreHigh.risk.Death     1.009     2.742    0.153 6.59 4.4e-11

Likelihood ratio test=59.5  on 4 df, p=3.66e-12
n= 70262, number of events= 1141
```

We see that all parameter estimates are closer to zero than those from the model for the cause-specific hazards. The parameters for relapse have changed more than the ones for death. This can be explained by the effect of the EBMT score on the death-specific mortality. Since the Medium risk and High risk group have a larger death-specific mortality than the Low risk group and this effect is less pronounced for relapse, we will see more relapses on the cumulative scale.

If we choose one overall weight function instead of the score-specific one, then we can use `Webmt1` instead of `Webmt.score` and obtain

```
> coxph(Surv(Tstart,Tstop,status==failcode)~strata(failcode) + Medium.Rel +
+     High.Rel + Medium.Mort + High.Mort, data=Webmt1, weights=weight.cens)

Call:
coxph(formula = Surv(Tstart, Tstop, status == failcode) ~ strata(failcode) +
```

```
    Medium.Rel + High.Rel + Medium.Mort + High.Mort, data = Webmt1,
    weights = weight.cens)


              coef exp(coef) se(coef)    z       p
Medium.Rel   0.240     1.271    0.125 1.92  0.0554
High.Rel     0.571     1.770    0.182 3.14  0.0017
Medium.Mort 0.570     1.769    0.113 5.04 4.8e-07
High.Mort    0.983     2.673    0.153 6.43 1.3e-10

Likelihood ratio test=55.4  on 4 df, p=2.72e-11
n= 127730, number of events= 1141
```

We see that parameter estimates are fairly similar and the conclusions with respect to significance do not change.

Make a graph that compares the predicted cause-specific cumulative incidence with the non-parametric estimates. Use the overlaid format for the three values of risk score and the separate format for the two event types.

This is somewhat easier than in the previous exercise. We can use the standard functionality of the `survival` package in the form of the `survfit` function. It can be used for prediction for several individuals at once. We create a data frame with six rows. Each row represents a combination of a value of risk score and event type.

```
> indivs <- data.frame(scoreMedium.risk.Relapse=c(0,0,1,0,0,0),
+   scoreMedium.risk.Death=c(0,0,0,1,0,0), scoreHigh.risk.Relapse=c(0,0,0,0,1,0),
+   scoreHigh.risk.Death=c(0,0,0,0,0,1), failcode=factor(rep(c("Relapse","Death"),3)))
> indivs

  scoreMedium.risk.Relapse scoreMedium.risk.Death scoreHigh.risk.Relapse
1                        0                       0                      0
2                        0                       0                      0
3                        1                       0                      0
4                        0                       1                      0
5                        0                       0                      1
6                        0                       0                      0
  scoreHigh.risk.Death failcode
1                    0  Relapse
2                    0    Death
3                    0  Relapse
4                    0    Death
5                    0  Relapse
6                    1    Death

> pred.sdh <- survfit(fit.sdh.comb, newdata=indivs)
```

We plot the curves (Figure 19).

```
> tmp.col <- c("black","red","green")
> par(mfrow=c(1,2),las=1)
> plot(survfit(Surv(Tstart,Tstop,status=="Relapse")~score,
+              data=subset(Webmt.score,failcode=="Relapse"), weights=weight.cens),
+      lwd=1, mark.time=FALSE, col=tmp.col, fun="event", ylim=c(0,0.5))
> lines(pred.sdh[c(1,3,5)],fun="event", col=tmp.col, mark.time=FALSE, lwd=3)
> title("Relapse")
> plot(survfit(Surv(Tstart,Tstop,status=="Death")~score,
+              data=subset(Webmt.score,failcode=="Death"), weights=weight.cens),
```

```
+      lwd=1, mark.time=FALSE, col=tmp.col, fun="event", ylim=c(0,0.55))
> lines(pred.sdh[c(2,4,6)],fun="event", col=tmp.col, mark.time=FALSE, lwd=3)
> title("Death")
> legend("bottomright", levels(Webmt1$score), col=tmp.col, lwd=3 )
```

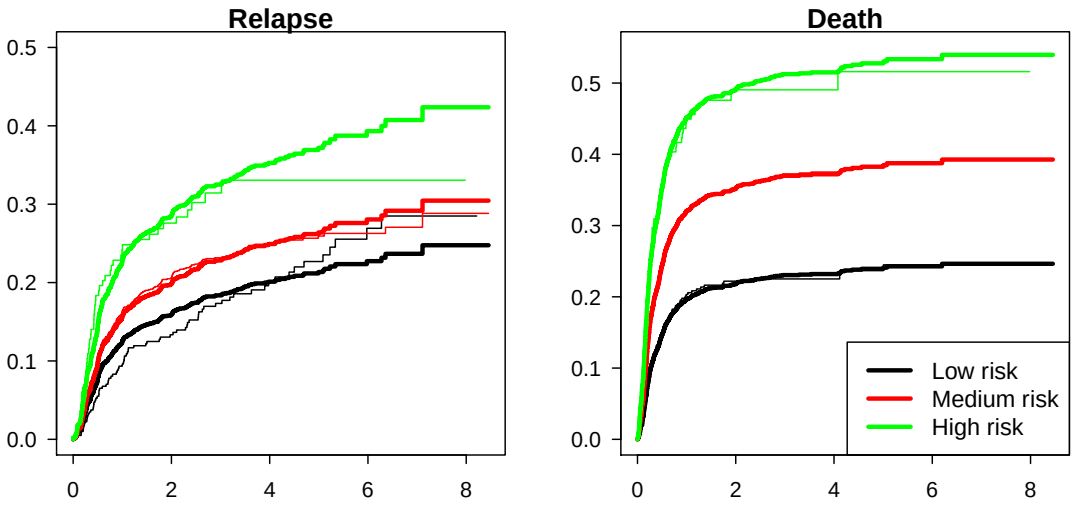

Figure 19: Predicted cause-specific cumulative incidence based on proportional subdistribution hazards model. Thin lines: nonparametric estimates.

We see that the fit is slightly worse than for the estimates based on the cause-specific hazards. Again, the largest difference is seen with relapse as event type for the low risk group.

Finally, we have a look at the assumption of proportional hazards for this model.

Test whether the proportional hazards assumption is reasonable.

We use the `cox.zph` function and obtain

```
> cox.zph(fit.sdh.comb, transform="identity")
```

```
                               rho    chisq        p
scoreMedium.risk.Relapse -0.153029 2.63e+01 2.85e-07
scoreHigh.risk.Relapse   -0.106127 1.25e+01 3.99e-04
scoreMedium.risk.Death    0.003889 1.72e-02 8.96e-01
scoreHigh.risk.Death      0.000268 8.15e-05 9.93e-01
GLOBAL                          NA 2.71e+01 1.93e-05
```

In Figure 20 we plot the estimate of the time trend in the parameter based on the scaled Schoenfeld residuals. It is seen that log hazard ratio remains fairly constant only during the first year.

We could have used the default transformation of time, based on the Kaplan-Meier. However, we choose not to transform time so that we can easily compare results with those from the `psh.test` function in the `crrSC` package. The `psh.test` function uses a sandwich estimator of the standard error. Since `psh.test` requires the event of interest to have value 1, for mortality as event type we create a temporary data set. We obtain

```
> library(crrSC)
> ## scoreMedium.risk.Relapse
> with(subset(Webmt1,failcode==1&count==1),
+      psh.test(Tstop, status, z=cbind(Medium.Rel,High.Rel), D=c(1,0),
+               tf=function(x) x))
```

```
> par(mfrow=c(2,2))
> plot(cox.zph(fit.csh.comb, transform="identity"),resid=FALSE)
```

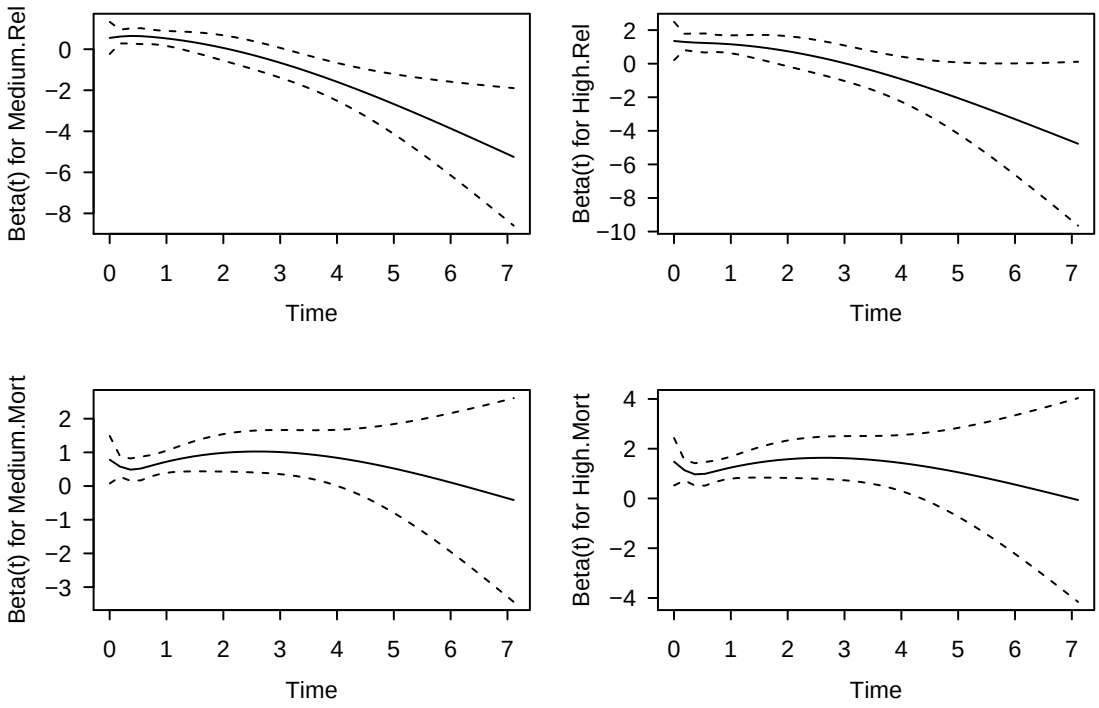

Figure 20: (Nonproportional) effect of EBMT score on the subdistribution hazard.

```
      % cen % cause 1 Test Statistic d.f. p-value
[1,] 0.4229    0.2307          5.104    1  0.0239

> ## scoreHigh.risk.Relapse
> with(subset(Webmt1,failcode==1&count==1),
+      psh.test(Tstop, status, z=cbind(Medium.Rel,High.Rel), D=c(0,1),
+               tf=function(x) x))

      % cen % cause 1 Test Statistic d.f. p-value
[1,] 0.4229    0.2307          7.839    1  0.0051

> ## death as event
> tmp <- subset(Webmt1,failcode==2&count==1)
> tmp$status <- ifelse(tmp$status==2,1,tmp$status*2)
> ## scoreMedium.risk.Death
> with(tmp, psh.test(Tstop,status,z=cbind(Medium.Mort,High.Mort), D=c(1,0),
+                    tf=function(x) x))

      % cen % cause 1 Test Statistic d.f. p-value
[1,] 0.4229    0.3465         0.6535    1  0.4189

> ## scoreHigh.risk.Death
> with(tmp, psh.test(Tstop,status,z=cbind(Medium.Mort,High.Mort), D=c(0,1),
+                    tf=function(x) x))

      % cen % cause 1 Test Statistic d.f. p-value
[1,] 0.4229    0.3465          1.277    1  0.2585
```

P-values are quite different, but the conclusions do not change: assuming proportionality for the impact of the EBMT score on the subdistribution hazard is probably not correct. This was already visible in Figures 18 and 19.

Multi-state analysis

In Section 4.10 we investigated the effect of the EBMT score on the transition hazards. Now we quantify what happens with respect to the transition probabilities. We use the proportional transition hazards model that i) assumes the effect of EBMT score to be the same for all transitions, ii) includes an effect of calendar period on the transition from relapse to death and iii) assumes the two transition hazards to death to be proportional. Hence, we neglect that the effect of the EBMT score on the relapse-specific hazard may be non-proportional.

3. Cumulative transition hazards

Compute all three cumulative transition hazards for an individual with a low EBMT score. For the transition from relapse to death, assume that relapse occurred in the period 1993-1996. Assign the result to an object named `HvH` and have a look at its structure.

The output of this proportional transition hazards model was stored in `fit.ebmt4`. Before we can use the `msfit` function, we need to create a data frame that represents an individual with Low risk score. This data frame contains values for all covariables used in the model on which it is based. It has as many rows as there are transitions in the model, and the values in row *k* refer to the *k*-th transition. We also need to define a **strata** column.

When creating the data frame, it is often convenient to start with the overall covariables and to mimic the relevant steps of the data building process in this new dataset (in particular `expand.covs` can be used). Since we don't have many covariables, it is quicker to define the data frame directly. In our illness-death multi-state model the first transition corresponds to the first stratum, and transitions 2 and 3 both correspond to the second stratum. Hence, we define *`strata=c(1,2,2)`*. Although not needed, we also specify columns **trans**, **from** and **to**.

```
> new.data <- data.frame(trans=1:3, from=c(1,1,2), to=c(2,3,3), score=1)
> new.data$score <- factor(new.data$score, levels=1:3, labels=levels(msebmt$score))
> new.data$strata <- c(1,2,2)
> new.data$yrel2.3 <- new.data$yrel1.3 <- 0
> new.data$rel.srv <- c(0,0,1)
> new.data

  trans from to    score strata yrel1.3 yrel2.3 rel.srv
1     1    1  2 Low risk      1       0       0       0
2     2    1  3 Low risk      2       0       0       0
3     3    2  3 Low risk      2       0       0       1
```

Now we can call the `msfit` function.

```
> HvH <- msfit(fit.ebmt4, newdata=new.data, trans=tmat.mst)
```

The resulting object is a list with three components, `Haz`, `varHaz` and `trans`.

```
> str(HvH)

List of 3
 $ Haz   :'data.frame':        1806 obs. of  3 variables:
  ..$ time : num [1:1806] 0.5 1 2 5 7 8 9 10 11 12 ...
  ..$ Haz  : num [1:1806] 0.000592 0.000889 0.000889 0.000889 0.000889 ...
  ..$ trans: int [1:1806] 1 1 1 1 1 1 1 1 1 1 ...
 $ varHaz:'data.frame':        3612 obs. of  4 variables:
  ..$ time  : num [1:3612] 0.5 1 2 5 7 8 9 10 11 12 ...
  ..$ varHaz: num [1:3612] 1.77e-07 2.67e-07 2.67e-07 2.67e-07 2.67e-07 ...
```

```
  ..$ trans1: int [1:3612] 1 1 1 1 1 1 1 1 1 1 ...
  ..$ trans2: int [1:3612] 1 1 1 1 1 1 1 1 1 1 ...
 $ trans : num [1:3, 1:3] NA NA NA 1 NA NA 2 3 NA
  ..- attr(*, "dimnames")=List of 2
  .. ..$ from: chr [1:3] "T" "R" "D"
  .. ..$ to  : chr [1:3] "T" "R" "D"
 - attr(*, "class")= chr "msfit"
```

The first component, `Haz`, is a data frame that contains the estimated patient-specific cumulative hazard for each of the transitions in the multi-state model; its third column gives the transition that the rows refers to. The second component, `varHaz`, is a data frame that contains the covariances of the estimated patient-specific cumulative hazards for each pair of transitions; the third and fourth column refer to the corresponding combination of two transitions $\widehat{\text{cov}}\text{ar}\{\widehat{H}_{\texttt{trans1}}(t), \widehat{H}_{\texttt{trans2}}(t)\}$. Those cumulative hazards and their covariances are evaluated at each unique time point with an observed transition (of any type). We can use the `summary` method for `msfit` objects to obtain a brief summary.

```
> summary(HvH)
```

By default it gives the head and tail of the hazard estimates of each of the transitions (result not shown).

Understand what is happening here:

```
> H0 <- HvH$Haz[HvH$Haz$trans==2,]
> H1 <- HvH$Haz[HvH$Haz$trans==3,]
> head(H1$Haz/H0$Haz)

[1] 3.67 3.67 3.67 3.67 3.67 3.67
```

We see that the quotient has the same value at each time point. This is not surprising since we assumed proportional baseline hazards for transitions 2 and 3.

Make a plot of the three cumulative transition hazards in overlaid format.

There is a `plot` method for `msfit` objects. Using `plot(HvH)` we obtain Figure 21. Note that, since Low risk is the reference value, the estimates for the transitions out of the transplant state T are the cumulative baseline hazards.

4. Impact of year of relapse on hazard We now try to get an impression of the effect of period of relapse on the hazard of death after relapse.

Make a graph with the cumulative transition hazards from relapse to death for each relapse period as well as the transition hazard directly from transplantation to death. Use the reference category for the EBMT score (i.e. Low risk).

We already defined `H0` and `H1` as the cumulative hazards for transplant to death (which does not depend on calendar period) and for relapse to death in 1993-1996. To obtain the other two hazards corresponding to period of relapse 1997-1999 and 2000+, we multiply the hazard in `H1` with the hazard ratios of the two other periods with respect to the first. This can be done because we assume proportionality, which on the scale of the cumulative hazard translates to a product as long as the covariable does not change over time. These hazard ratios can be extracted from the `coef` component of the `coxph` object. The code to define the hazards is:

```
> H2 <- H3 <- H1
> H2$Haz <- H2$Haz*exp(fit.ebmt4$coef[3])
> H3$Haz <- H3$Haz*exp(fit.ebmt4$coef[4])
```

Alternatively, we could rebuild the datasets and use `msfit` again.

In Figure 22 we see that the effect of calendar period of relapse on the transition hazard is fairly strong.

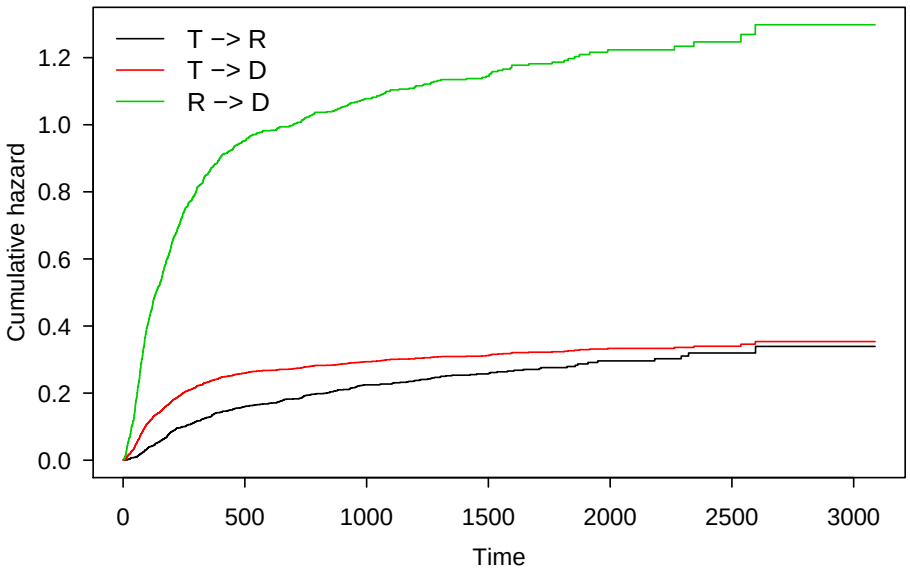

Figure 21: Cumulative transition hazards for an individual with low EBMT score. We assume calendar period 1993-1996 for the transition from relapse to death.

5. Transition probabilities The impact of the period of relapse on death looks quite impressive, but the question remains how large the impact is for a population followed from transplantation; after all, if hardly anyone gets a relapse, a large difference in mortality after relapse will not have a big impact on a population. In order to study this we need to compute the probability of dying after transplantation, with or without relapse, for the three different relapse periods.

Compute the transition probabilities for an individual that had a low EBMT score and a relapse in 1993-1996. Make a graph that shows all state occupation probabilities for such an individual. Think about the best display format of the curves.

What does a probability of death, starting from transplantation, mean for a patient with a relapse between 1993 and 1996?

The result of `msfit` can be used as input for `probtrans`.

```
> tpr.cal1 <- probtrans(HvH, predt=0, direction="forward")
```

We can use the `summary.probtrans` function to obtain heads and tails of predictions from each starting state. By default it gives the head and tail of the estimates of the transition probabilities out of each of the transitions (results not shown).

```
> summary(tpr.cal1)
```

The prediction probabilities can also be accessed more directly. For instance, predictions from state 1 are in the first component of the list:

```
> tpr.cal1.1 <- tpr.cal1[[1]]
> head(tpr.cal1.1)
```

```
  time pstate1   pstate2   pstate3       se1       se2       se3
1  0.0  1.0000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
2  0.5  0.9988 0.0005924 0.0005924 0.0005975 0.0004204 0.0004207
3  1.0  0.9985 0.0008888 0.0005924 0.0006696 0.0005163 0.0004207
```

```
> plot(H1$time/365.25, H1$Haz, type="s", ylim=c(0,max(H1$Haz)),
+  xlab="Years since transplant", ylab="Cumulative hazard", lwd=2, col="red2")
> lines(H2$time/365.25, H2$Haz, type="s", lwd=2, col="orangered")
> lines(H3$time/365.25, H3$Haz, type="s", lwd=2, col="orange")
> lines(H0$time/365.25, H0$Haz, type="s", lwd=2, col=3)
> legend("topleft", c("Relapse in 1993-1996","Relapse in 1997-1999",
+        "Relapse in 2000 or later","No relapse"), lwd=2,
+         col=c("red2","orangered","orange",3), bty="n")
```

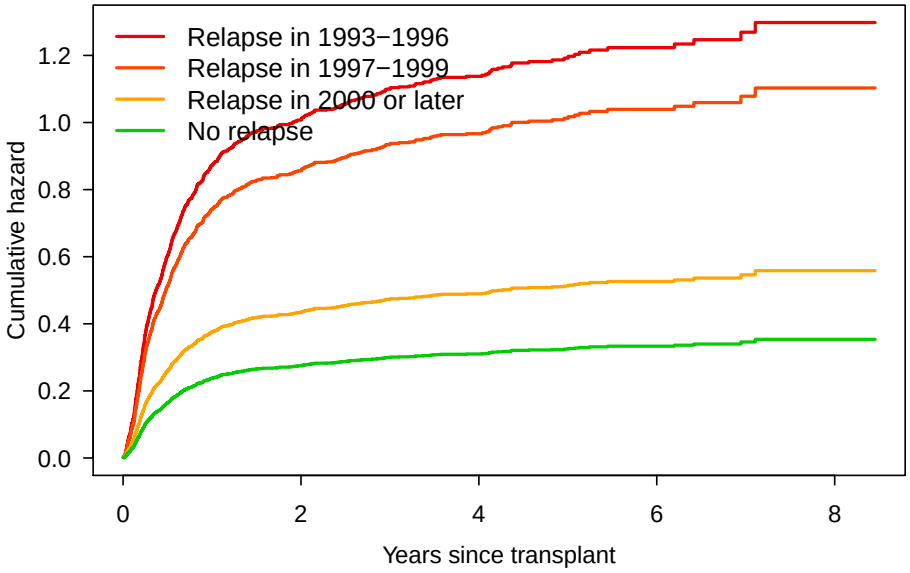

Figure 22: Cumulative transition hazards from transplantation to death (without relapse) and from relapse to death in three calendar periods

```
4   2.0   0.9982 0.0008879 0.0008884 0.0007347 0.0005157 0.0005159
5   5.0   0.9973 0.0008850 0.0017766 0.0009048 0.0005140 0.0007338
6   7.0   0.9967 0.0008831 0.0023690 0.0010043 0.0005129 0.0008510
```

The column `pstate3` gives the estimate of the cumulative death probability.

How do we interpret this value? The origin of the time scale is transplant, whereas relapse is an intermediate state. An individual that received a transplant in 1993-1996 does not necessarily have a relapse in the same period. He can have a relapse in a later calendar period and can also die without relapse. If he had a relapse in 1993-1996, it means that he had relapse shortly after transplant. Hence, he does not represent all individuals that received a transplant in 1993-1996. The state occupation probabilities that we calculate can best be interpreted as what would have happened after transplant if the treatment of relapse had not improved after 1996.

We can use the `plot.probtrans` function to visualize the estimates of the state occupation probabilities, which are the transition probabilities from the first state. The argument *`type`* distinguishes between different types of plots. The overlaid format is obtained by using *`type="single"`*, as shown in Figure 23. Note that the first state is the default, so we could leave out *`from=1`*.

Perhaps a more informative figure is obtained using the stacked format. For our three states, we get a nice overview if we first plot the probability of a relapse, then stack the death probability, and finally transplant. The order relapse–death–transplant is not the same as the states defined in `tmat.mst`. Therefore we specify the argument *`ord`* as *c(2,3,1)*. The result is shown in Figure 24. The lowest curve of Figure 24 indicates the probability of being in state 2 (alive with relapse), the second curve the probability of being in state 2 plus the probability of being in state 3. The distance between these two curves is the probability of dying. The second

```
> plot(tpr.call1, from=1, type="single")
```

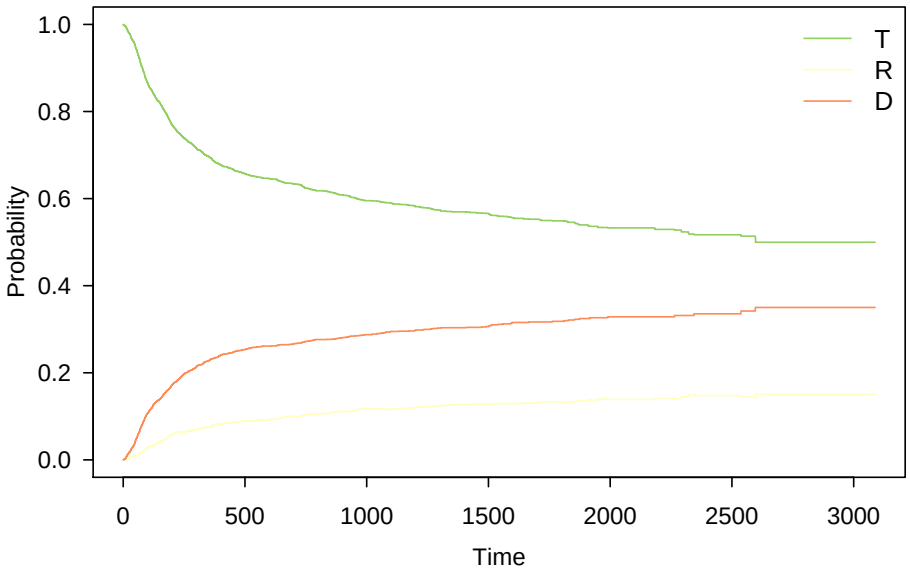

Figure 23: State occupation probabilities for a Low risk patient in 1993-1996, overlaid display format

```
> plot(tpr.call1, ord=c(2,3,1))
```

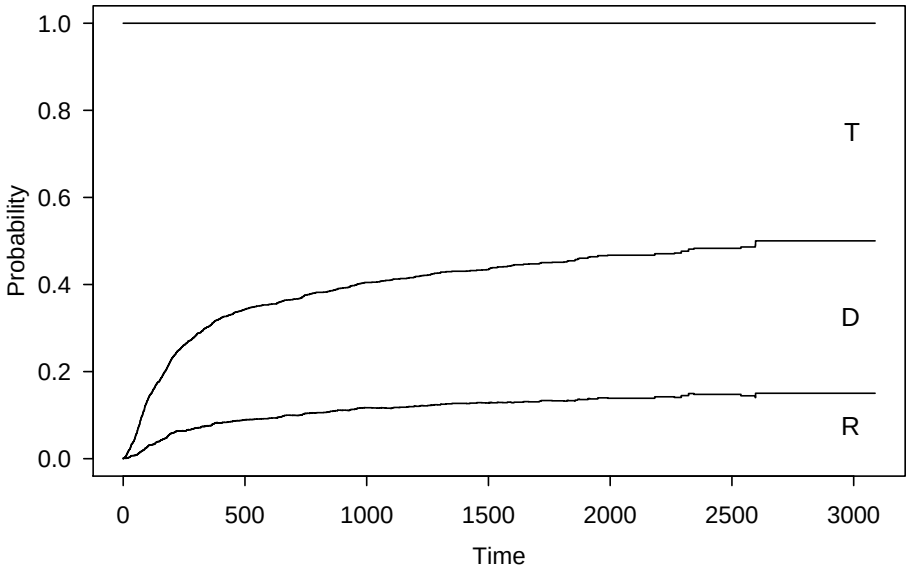

Figure 24: State occupation probabilities for a Low risk patient in 1993-1996, stacked format

curve thus indicates the probability of treatment failure (either relapse or death). The distance between the second curve and 1 is the relapse-free survival.

An even nicer picture is obtained when specifying `type="filled"` argument. In that case the spaces between two adjacent curves are colored. Figure 25 shows an example.

6. Impact of year of relapse on transition probability Now we will compare the transition probability to death that

```
> plot(tpr.cal1, ord=c(2,3,1), type="filled", cols=c("orange","red","green3"))
```

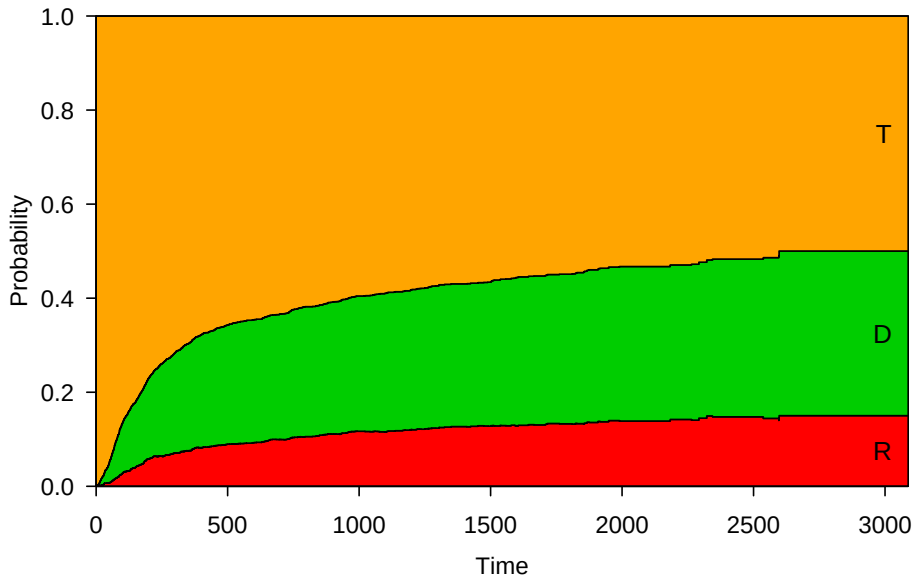

Figure 25: State occupation probabilities for a Low risk patient in 1993-1996, stacked (and filled) format

we obtained in the previous exercise (patients in 1993-1996) with those from patients in later calendar periods.

Compare the transition probabilities for patients having a relapse in 1993-1996 with those obtained from patients having a relapse in 1997-1999, or 2000+ (and with the same EBMT risk score, low risk). What is your conclusion?

In order to obtain these prediction probabilities we repeat what we have just done. We create new data frames in which `yrel1.3` and `yrel2.3` are changed according to the calendar period of interest. After this, we apply `msfit` and `probtrans` again. The code is tedious, but straightforward.

```
> new.data.copy <- new.data # make a copy for later
> # first dummy variable = 1 for transition 3
> new.data$yrel1.3[new.data$trans==3] <- 1
> new.data

  trans from to    score strata yrel1.3 yrel2.3 rel.srv
1     1    1  2 Low risk      1       0       0       0
2     2    1  3 Low risk      2       0       0       0
3     3    2  3 Low risk      2       1       0       1

> HvH2 <- msfit(fit.ebmt4, newdata=new.data, trans=tmat.mst)
> new.data <- new.data.copy # use the copy
> # second dummy variable = 1 for transition 3
> new.data$yrel2.3[new.data$trans==3] <- 1
> new.data

  trans from to    score strata yrel1.3 yrel2.3 rel.srv
1     1    1  2 Low risk      1       0       0       0
2     2    1  3 Low risk      2       0       0       0
3     3    2  3 Low risk      2       0       1       1
```

```
> HvH3 <- msfit(fit.ebmt4, newdata=new.data, trans=tmat.mst)
> tpr.cal2 <- probtrans(HvH2, predt=0, direction="forward")
> tpr.cal3 <- probtrans(HvH3, predt=0, direction="forward")
```

The result is shown in Figure 26. We see that the impact of calendar period on mortality is much less pronounced than on the transition hazard from relapse to death. This can be explained by the fact that more individuals die without relapse.

```
> plot(tpr.cal1[[1]]$time/365.25, tpr.cal1[[1]]$pstate3, type="s", ylim=c(0,1),
+     xlab="Years since transplant", ylab="Probability of death", lwd=2, col="red")
> lines(tpr.cal2[[1]]$time/365.25, tpr.cal2[[1]]$pstate3, type="s", lwd=2, col="orange")
> lines(tpr.cal3[[1]]$time/365.25, tpr.cal3[[1]]$pstate3, type="s", lwd=2, col="blue")
> legend("topleft", c("1993-1996","1997-1999","2000 or later"),
+        lwd=2, col=c("red","orange","blue"), bty="n")
```

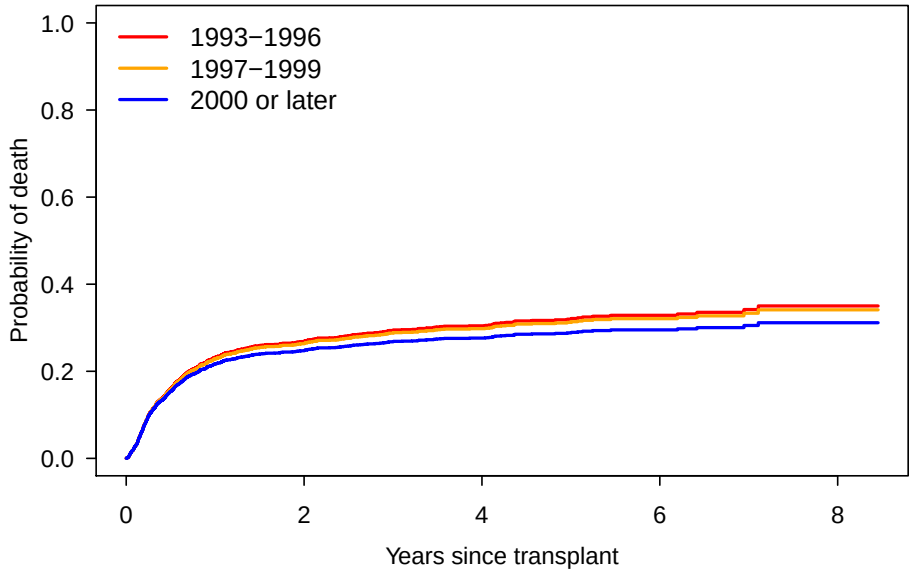

Figure 26: Cumulative death probability for a Low risk patient in different calendar periods

Would you expect the impact of period of relapse on survival after transplant to be stronger for a high risk patient? Repeat what you have done for a patient with a high EBMT risk score.

Again we repeat what we have done before, but now also with the value of EBMT risk score changed. Here is the code:

```
> new.data <- new.data.copy # use copy from last time
> new.data$score <- "High risk"
> new.data

  trans from to     score strata yrel1.3 yrel2.3 rel.srv
1     1    1  2 High risk      1       0       0       0
2     2    1  3 High risk      2       0       0       0
3     3    2  3 High risk      2       0       0       1

> new.data.copy <- new.data # make a copy for later
> HvH1 <- msfit(fit.ebmt4, newdata=new.data, trans=tmat.mst)
> new.data <- new.data.copy # used the copy
```

```
> # first dummy variable = 1 for transition 3
> new.data$yrel1.3[new.data$trans==3] <- 1
> new.data

  trans from to     score strata yrel1.3 yrel2.3 rel.srv
1     1    1  2 High risk      1       0       0       0
2     2    1  3 High risk      2       0       0       0
3     3    2  3 High risk      2       1       0       1

> HvH2 <- msfit(fit.ebmt4, newdata=new.data, trans=tmat.mst)
> new.data <- new.data.copy
> # second dummy variable = 1 for transition 3
> new.data$yrel2.3[new.data$trans==3] <- 1
> new.data

  trans from to     score strata yrel1.3 yrel2.3 rel.srv
1     1    1  2 High risk      1       0       0       0
2     2    1  3 High risk      2       0       0       0
3     3    2  3 High risk      2       0       1       1

> HvH3 <- msfit(fit.ebmt4, newdata=new.data, trans=tmat.mst)
> tpr.cal1 <- probtrans(HvH1, predt=0, direction="forward")
> tpr.cal2 <- probtrans(HvH2, predt=0, direction="forward")
> tpr.cal3 <- probtrans(HvH3, predt=0, direction="forward")
```

And the plot is shown in Figure 27.

```
> plot(tpr.cal1[[1]]$time/365.25, tpr.cal1[[1]]$pstate3, type="s", ylim=c(0,1),
+     xlab="Years since transplant", ylab="Probability of death", lwd=2, col="red")
> lines(tpr.cal2[[1]]$time/365.25, tpr.cal2[[1]]$pstate3, type="s", lwd=2, col="orange")
> lines(tpr.cal3[[1]]$time/365.25, tpr.cal3[[1]]$pstate3, type="s", lwd=2, col="blue")
> legend("topleft", c("1993-1996","1997-1999","2000 or later"),
+     lwd=2, col=c("red","orange","blue"), bty="n")
```

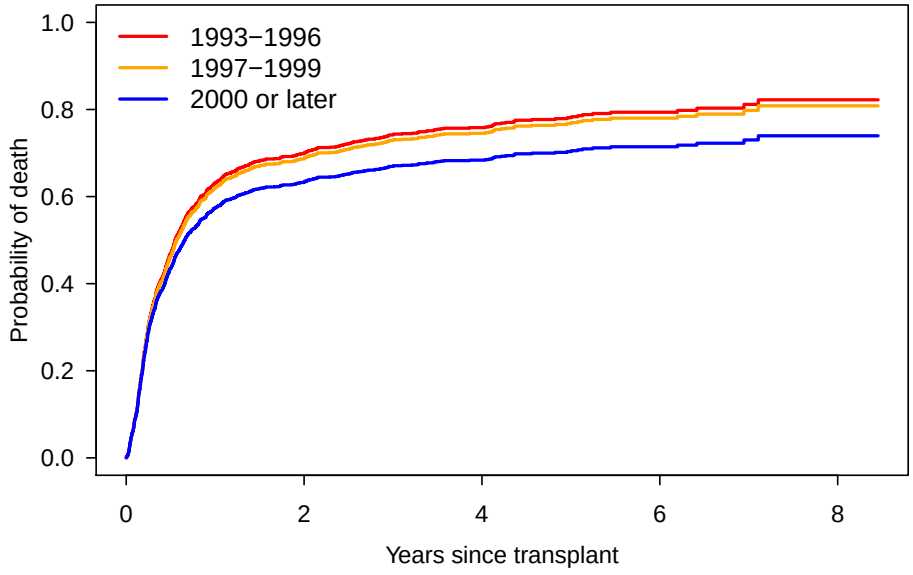

Figure 27: Cumulative death probability for a High risk patient in different calendar periods

Note that the effect of the EBMT risk score was assumed to be proportional on the transition hazards. Although is has no straightforward relation to the state occupation probabilities, yet we see that the main difference with a low score patient is that all three curves have shifted up. The relative position did not change.