

# Multi-state models and competing risks

Terry Therneau

Cynthia Crowson

Elizabeth Atkinson

April 28, 2016

## 1 Multi-state models

A multi-state model is used to model a process where subjects transition from one state to the next. For instance, a standard survival curve can be thought of as a simple multi-state model with two states (alive and dead) and one transition between those two states. A diagram illustrating this process is shown in the top left corner of figure 1. In these types of diagrams, each box is a state and each arrow is a possible transition. The top right diagram depicts a classic competing risk analysis, where all subjects start on the left and each subject can make a single transition to one of 3 terminal states. The bottom left diagram shows a common multi-state situation known as the illness-death model with recovery. Finally, the lower right diagram represents sequential events, such as repeated infections in the CGD study. In that case one subject had 8 events so there are 9 states corresponding to entry into the study (0 infections) and the first, second, ..., eighth events.

As will be shown below, there are often multiple choices for the state and transition diagram, and for some data sets it is revealing to look at a problem from multiple views. In addition to deciding the diagram that best matches the research questions, the two other primary decisions are the choice of time scale for the fits, e.g., time from entry in the study vs. time from entry in the state, and what covariates will be used.

## 2 Multi-state curves

### 2.1 Aalen-Johansen estimate

As a starting point for the analysis, it is important to compute and plot estimates of  $p(t)$ , which is a vector containing the probability of being in each of the states at time  $t$ . If there is no censoring then  $p$  becomes a simple tabulation at time  $t$  of all the states. For the general case, we compute this using the Aalen-Johansen estimate via the `survfit` function.

Mathematically the estimate is simple. For each unique time where an event occurs, form the transition matrix  $T(t)$  with elements or rates of  $\lambda_{ij}(t)$  = the fraction of subjects who transition from state  $i$  to  $j$  at time  $t$ , among those in state  $i$  just prior to time  $t$ . ( $T$  is equal to the identity matrix at any time point without an observed transition.) Then

$$p(t) = p(0) \prod_{s \leq t} T(s) \tag{1}$$

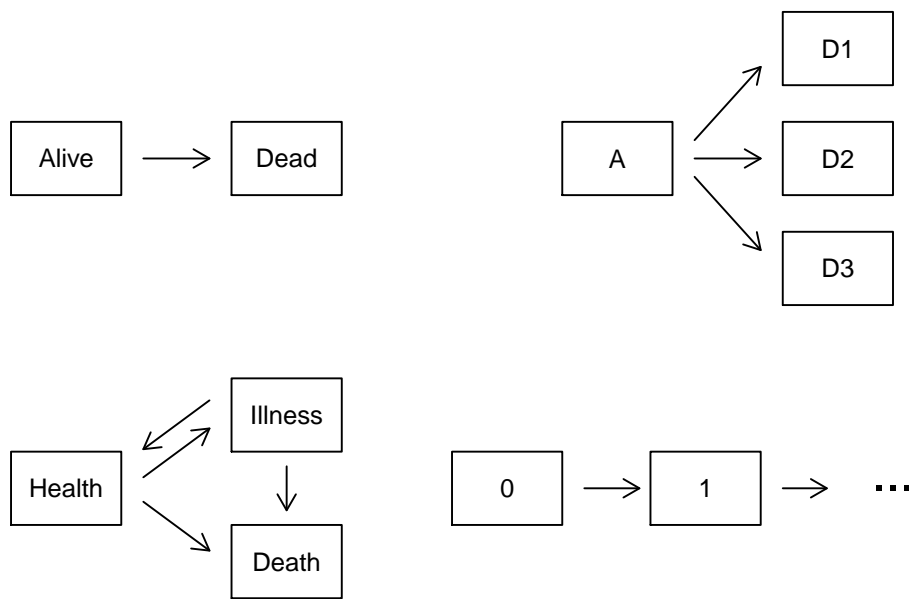


Figure 1: Four multi-state models. The upper left panel depicts simple survival, the upper right is an example of competing risks, the lower left panel is a multi-state illness-death model, and the lower right panel depicts sequential events.

where  $p(0)$  is the initial distribution of subjects.

Let's work this out for the simple two-state alive  $\rightarrow$  death model. Let  $n(t)$  be the number of subjects still at risk at time  $t$  and  $d(t)$  the number of deaths at that time point. All subjects start in the alive state and thus  $p(0) = (1, 0)$  and the transition matrix is

$$T(s) = \begin{pmatrix} \frac{n(s)-d(s)}{n(s)} & \frac{d(s)}{n(s)} \\ 0 & 1 \end{pmatrix}$$

The second row corresponds to the fact that death is an absorbing state.

Writing out the matrices for the first few transitions and multiplying them leads to

$$p_1(t) = \prod_{s \leq t} [n(s) - d(s)] / n(s) \quad (2)$$

which we recognize as the Kaplan-Meier estimate of survival. For the two state alive-dead model the Aalen-Johansen (AJ) estimate has reprised the KM. In the competing risks case  $p(t)$  has an alternate form known as the *cumulative incidence* (CI) function

$$CI_k(t) = \int_0^t \lambda_k(u) S(u-) du \quad (3)$$

where  $\lambda_k$  is the incidence function for outcome  $k$ , and  $S$  is the overall survival curve for “time to any endpoint”. (The label “cumulative incidence” is one of the more unfortunate ones in the survival lexicon, since we normally use ‘incidence’ and ‘hazard’ as interchangeable synonyms but the CI is *not* a cumulative hazard.) Repeating the same matrix exercise for the competing risks, i.e. writing out the Aalen-Johansen computation, exactly recovers the CI formula. The CI is also a special case of the Aalen-Johansen. The AJ estimate is very flexible; subjects can visit multiple states during the course of a study, subjects can start after time 0 (delayed entry), and they can start in any of the states. The `survfit` function implements the AJ estimate and will handle all these cases.

The standard error of the estimates is computed using an infinitesimal jackknife. Let  $D(t)$  be a matrix with one row per subject and one column per state. Each row contains the *change* in  $p(t)$  corresponding to subject  $i$ , i.e., the derivative of  $p$  with respect to the  $i$ th subject's case weight  $dp(t)/dw_i$ . Then  $V(t) = D'D$  is the estimated variance-covariance matrix of the estimates at time  $t$ .

The  $p(t)$  vector obeys the obvious constraint that its sum at any time is equal to one; each person has to be somewhere. I prefer the label *current prevalence* estimate, since it estimates what fraction of the subjects are in any given state across time. However the word “prevalence” is likely to generate confusion whenever death is one of the states, due to its traditional use as the fraction of living subjects who have a particular condition. We will use the phrase *probability in state* or simply  $p$  from this point forward.

In the simple two state model  $\text{Pr}(\text{alive})$  is the usual KM survival estimate, and we have  $p_1(t) = 1 - p_2(t)$ ,  $\text{Pr}(\text{alive}) = 1 - \text{Pr}(\text{dead})$ . Plots for the 2 state case sometimes choose to show  $\text{Pr}(\text{alive})$  and sometimes  $\text{Pr}(\text{dead})$ . Which one is used often depends on a historical whim of the disease specialty; cardiology journals for instance quite often use  $\text{Pr}(\text{event})$  resulting in curves that rise starting from zero, while oncology journals invariably use  $\text{Pr}(\text{alive})$  giving curves that

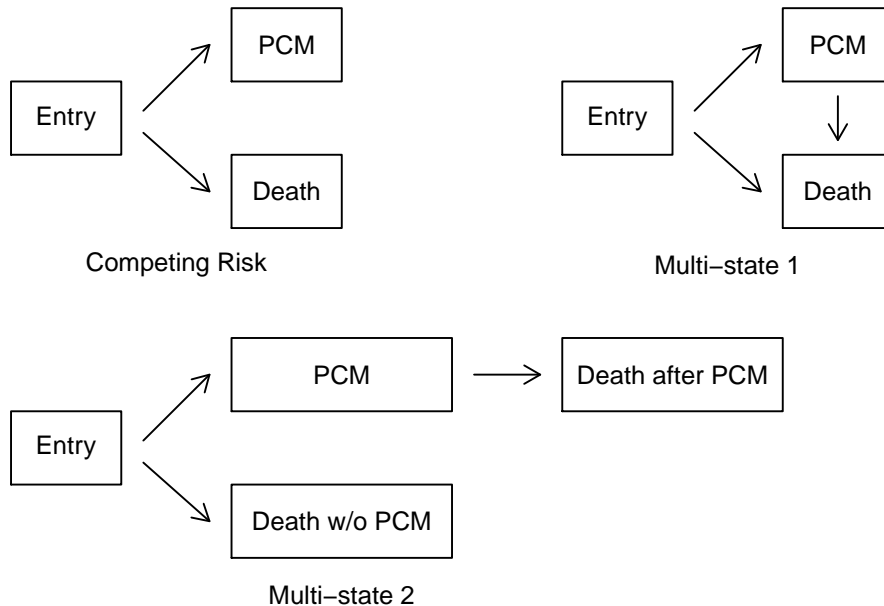


Figure 2: Three models for the MGUS data.

fall downhill from 1. The `survfit` routine's historical default for the 2 state case is to print and plot  $\Pr(\text{alive}) = p_1(t)$ , which reflects that the author of the routine was working primarily in cancer trials at the time said default was chosen.

For simple survival we have gotten used to the idea of using  $\Pr(\text{dead})$  and  $\Pr(\text{alive})$  interchangeably, but that habit needs to be left behind for multi-state models, as curves of  $1 - p_k(t) = \text{probability}(\text{any other state than } k)$  are not useful. In the multi-state case, individual curves can go both up and down. For competing risks the curve for the initial state (leftmost in the diagram) is rarely included in the final plot. Since the curves sum to 1, the full set is redundant.  $\Pr(\text{nothing yet})$  is usually the least interesting of the set and so it is left off to make the plot less busy. The remaining curves in the competing risks case rise from 0. (This bothers some researchers as it 'just looks wrong' to them.)

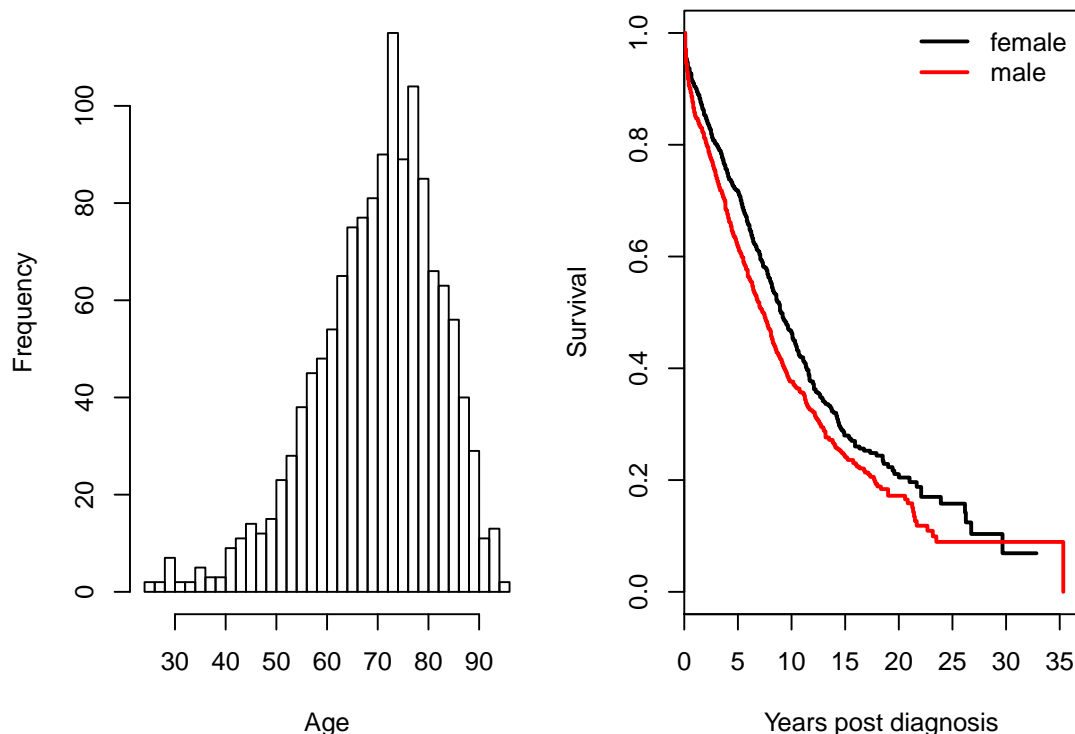
## 2.2 Examples

Start with a simple competing risks problem as illustrated in the first diagram of figure 2. The `mgus2` data set contains the time to plasma cell malignancy (PCM) and/or death for 1384 subjects diagnosed with monoclonal gammopathy of undetermined significance (MGUS). Survival and progression time are in months. The code below creates an ordinary Kaplan-Meier curve of post-diagnosis survival for these subjects, along with a histogram of age at diagnosis. The mean

age at diagnosis is just over 70 years.

```
> oldpar <- par(mfrow=c(1,2))
> hist(mgus2$age, nclass=30, main='', xlab="Age")
> with(mgus2, tapply(age, sex, mean))
      F      M
71.32171 69.67065
> mfit1 <- survfit(Surv(futime, death) ~ sex, data=mgus2)
> mfit1
Call: survfit(formula = Surv(futime, death) ~ sex, data = mgus2)

      n events median 0.95LCL 0.95UCL
sex=F 631    423   108    100   121
sex=M 753    540    88     79    97
> plot(mfit1, col=c(1,2), xscale=12, mark.time=FALSE, lwd=2,
      xlab="Years post diagnosis", ylab="Survival")
> legend("topright", c("female", "male"), col=1:2, lwd=2, bty='n')
> par(oldpar)
```

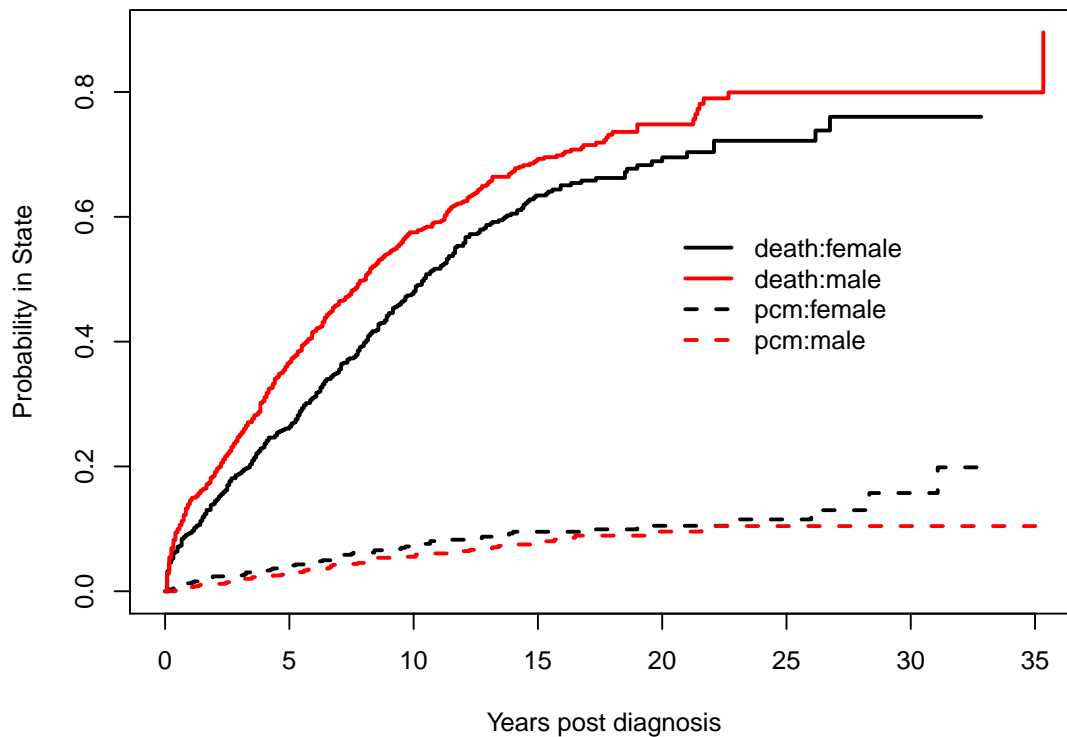


The `xscale` and `yscale` arguments to `plot.survfit` affect only the axis labels, not the data. Further additions to the plot region such as `legend`, `lines`, or `text` remain in the original scale. This simplifies programmatic additions such as adding another curve to the plot, while making interactive additions such as a legend somewhat less simple.

As a second model for these subjects we will use competing risks with PCM and death without malignancy as the two terminal states, as shown in the upper left of figure 2. For this model we are only interested in the first event for each subject. Formally we are treating progression to a PCM as an *absorbing state*, i.e., one that subjects never exit. We create a variable `etime` containing the time to the first of progression, death, or last follow-up along with an event variable that contains the outcome. The starting data set `mgus2` has two pairs of variables (`ptime`, `pstat`) that contain the time to progression and (`fuptime`, `status`) that contain the time to death or last known alive. The code below creates the necessary `etime` and `event` variables, then computes and plots the competing risks estimate.

```
> etime <- with(mgus2, ifelse(pstat==0, fuptime, ptime))
> event <- with(mgus2, ifelse(pstat==0, 2*death, 1))
> event <- factor(event, 0:2, labels=c("censor", "pcm", "death"))
> table(event)
event
censor    pcm  death
  409     115    860
> mfit2 <- survfit(Surv(etime, event) ~ sex, data=mgus2)
> print(mfit2, rmean=240, scale=12)
Call: survfit(formula = Surv(etime, event) ~ sex, data = mgus2)

              n nevent      mean*
sex=F, pcm    631      59  1.323284
sex=M, pcm    753      56  1.064693
sex=F, death  631     370  8.823108
sex=M, death  753     490 10.260294
sex=F,        631       0  9.853608
sex=M,        753       0  8.675012
  *mean time in state, restricted (max time = 20 )
> mfit2$transitions
      to
from   pcm death
pcm    0     0
death  0     0
      115    860
> plot(mfit2, col=c(1,2,1,2), lty=c(2,2,1,1),
       mark.time=FALSE, lwd=2, xscale=12,
       xlab="Years post diagnosis", ylab="Probability in State")
> legend(240, .6, c("death:female", "death:male", "pcm:female", "pcm:male"),
       col=c(1,2,1,2), lty=c(1,1,2,2), lwd=2, bty='n')
```



The `mfit2` call is nearly identical to that for an ordinary Kaplan-Meier, with the exception of the `event` variable.

1. The event variable was created as a *factor*, whereas for ordinary single state survival the status is either 0/1 or TRUE/FALSE. The first level of the factor must be censoring, which is the status code for those whose follow-up terminated without reaching a new endpoint. Codes for the remaining states can be in any order. The labels for the states are unrestricted, e.g., the first one does not have to be “censor”. (It will however be treated as censoring, whatever the name.)
2. A simple print of the `mfit2` object shows the order in which the curves will be displayed. This information was used to choose the line types and colors for the curves.
3. The `mfit2` object contains curves for all the states, but by default the entry state will not be plotted. The remaining curves all start at 0.
4. The transitions component of the result is useful as a data check, e.g., if it showed a transition from death to PCM.
5. Each subject’s initial state is specified by the `istate` argument. When this is omitted all subjects are assumed to start from an entry state named “ ” (the empty string), as seen in the printout above.

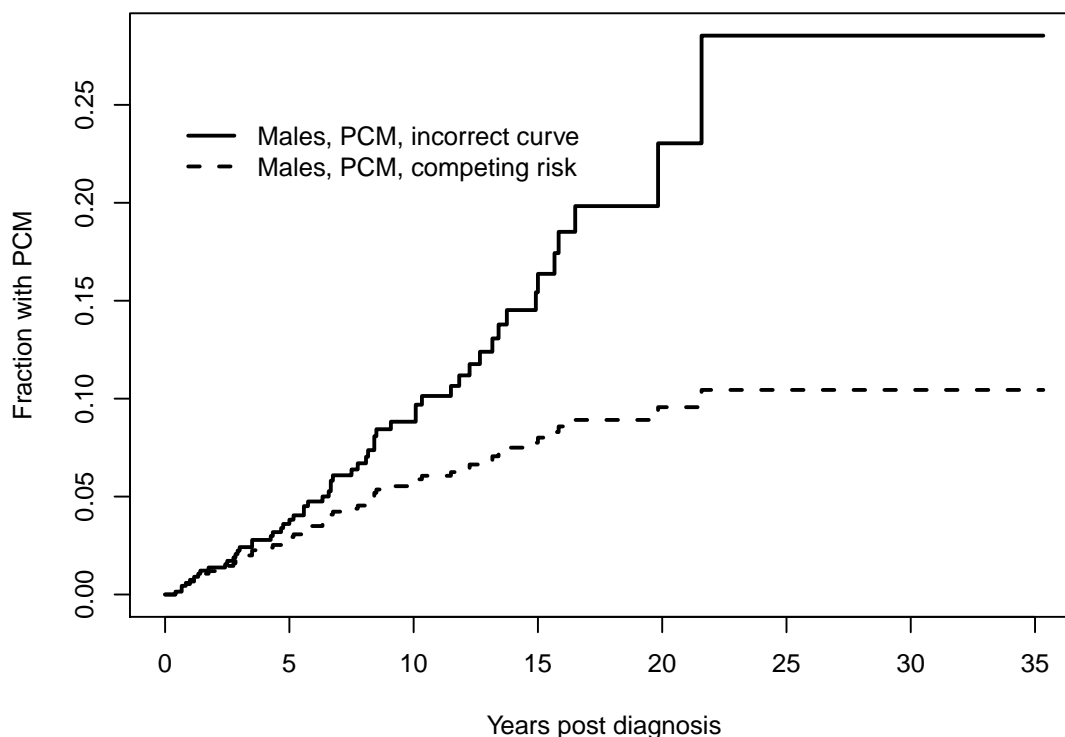
The printout shows that a male subject will spend, on average, 8.7 of his first 20 years post diagnosis in the entry state, 1.1 years in the PCM state and 10.3 of those 20 in the death state.

If a cutoff time is not given the default is to use the maximum observed time for all curves, which is 424 months in this case.

The result of a multi-state `survfit` is a matrix of probabilities with one row per time and one column per state. First are the states found in the event variable (excluding censoring) and then the states found in the `istate` variable, removing any duplicates. By default any unnamed state is not plotted – point 3 above – for the simple reason that multiple event curves can very quickly get overcrowded with all the multiple lines. Since the three MGUS states of entry/pcm/death must sum to 1 at any given time (everyone has to be somewhere), one of the three curves is redundant and the “fraction still in the entry state” curve is normally the least interesting. One can easily add this last state to the plot if desired, e.g., `lines(mfit2[,3], col=4, lty=1:2)`, since entry is the third state in the printout. (One can use, e.g., `mfit2[, 'pcm']` to select a state as well, but an empty string does not work as the subscript.)

A common mistake with competing risks is to use the Kaplan-Meier separately on each event type while treating other event types as censored. The next plot is an example of this for the PCM endpoint.

```
> pcmbad <- survfit(Surv(etime, pstat) ~ sex, data=mgus2)
> plot(pcmbad[2], mark.time=FALSE, lwd=2, fun="event", conf.int=FALSE, xscale=12,
      xlab="Years post diagnosis", ylab="Fraction with PCM")
> lines(mfit2[2,1], lty=2, lwd=2, mark.time=FALSE, conf.int=FALSE)
> legend(0, .25, c("Males, PCM, incorrect curve", "Males, PCM, competing risk"),
      col=1, lwd=2, lty=c(1,2), bty='n')
```





There are two problems with the `pcmbad` fit. The first is that it attempts to estimate the expected rate of plasma cell malignancy (PCM) if all other causes of death were to be disallowed. In this hypothetical world it is indeed true that many more subjects would progress to PCM (the incorrect curve is higher), but it is also not a world that any of us will ever inhabit. This author views the result in much the same light as discussions of survival after the zombie apocalypse. The second problem is that the computation for this hypothetical case is only correct if all of the competing endpoints are independent, a situation which is almost never true. We thus have an unreliable estimate of an uninteresting quantity. The competing risks curve, on the other hand, estimates the fraction of MGUS subjects who *will experience* PCM, a quantity sometimes known as the lifetime risk, and one which is actually observable.

The last example chose to plot only a subset of the curves, something that is often desirable in competing risks problems to avoid a “tangle of yarn” plot that simply has too many elements. This is done by subscripting the `survfit` object. For subscripting, multi-state curves behave as a matrix with the outcomes as the second subscript. The columns are in order of the levels of `event`, i.e., as displayed by our earlier call to `table(event)`. The first subscript indexes the groups formed by the right hand side of the model formula, and will be in the same order as simple survival curves. Thus `mfit2[2,1]` corresponds to males (2) and the PCM endpoint (1). Curves are listed and plotted in the usual matrix order of R.

A third example using the MGUS data treats it as a multi-state model and it shown in the upper right of figure 2. In this version a subject can have multiple transitions and thus multiple rows in the data set. In this case it is necessary to identify which data rows go with which subject via the `id` argument of `survfit`; valid estimates of the curves and their standard errors both depend on this. Our model looks like the illness-death model of figure 1 but with “PCM” as the upper state and no arrow for a return from that state to health. The necessary data set will have two rows for any subject who has further follow-up after a PCM and a single row for all others. The data set is created below using the `tmerge` function, which is discussed in detail in another vignette.

We need to decide what to do with the 9 subjects who have PCM and death declared at the same month. (Some of these were cancer cases discovered at autopsy.) They slipped through without comment in the earlier competing risks analysis; only when setting up this second data set did we notice the ties. Looking back at the code, the prior example counted these subjects as a progression. In retrospect this is a defensible choice: even though undetected before death, the disease must have been present for some amount of time previous and so progression did occur first. For the multi-state model we need to be explicit in how this is coded since a sojourn time of 0 within a state is not allowed. Below we push the progression time back by .1 month when there is a tie, but that amount is entirely arbitrary.

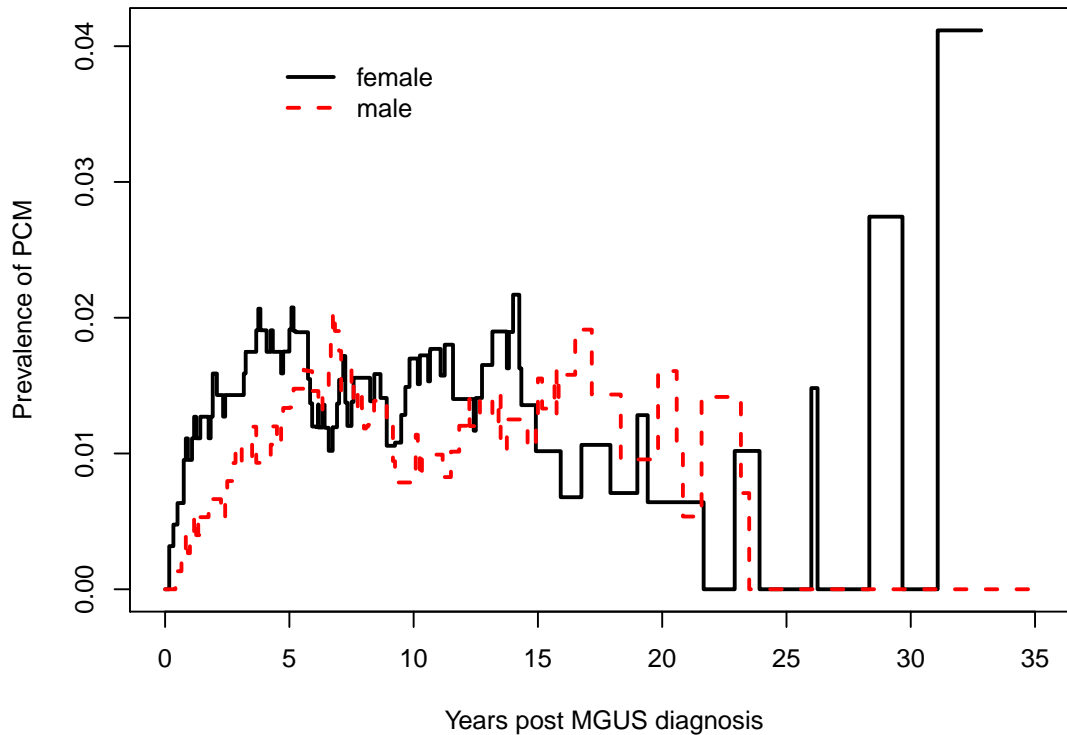
```
> ptemp <- with(mgus2, ifelse(ptime==fuptime & pstat==1, ptime-.1, ptime))
> newdata <- tmerge(mgus2, mgus2, id=id, death=event(fuptime, death),
  pcm = event(ptemp, pstat))
> newdata <- tmerge(newdata, newdata, id, enum=cumtdc(tstart))
> with(newdata, table(death, pcm))
```

	pcm	
death	0	1
0	421	115
1	963	0

The table above shows that there are no observations in `newdata` that have both a PCM and death, i.e., the ties have been resolved. The last `tmerge` line above creates a variable `enum` which simply counts rows for each person; it will be used later.

```
> temp <- with(newdata, ifelse(death==1, 2, pcm))
> newdata$event <- factor(temp, 0:2, labels=c("censor", "pcm", "death"))
> mfit3 <- survfit(Surv(tstart, tstop, event) ~ sex, data=newdata, id=id)
> print(mfit3, rmean=240, digits=2)
Call: survfit(formula = Surv(tstart, tstop, event) ~ sex, data = newdata,
              id = id)

              n nevent mean*
sex=F, pcm    690      59   3.2
sex=M, pcm    809      56   2.7
sex=F, death  690     423 118.6
sex=M, death  809     540 133.2
sex=F,        690       0 118.2
sex=M,        809       0 104.1
      *mean time in state, restricted (max time = 240 )
> mfit3$transitions
      to
from   pcm death
pcm     0   103
death  0     0
      115   860
> plot(mfit3[,1], mark.time=FALSE, col=1:2, lty=1:2, lwd=2,
       xscale=12,
       xlab="Years post MGUS diagnosis", ylab="Prevalence of PCM")
> legend(48, .04, c("female", "male"), lty=1:2, col=1:2, lwd=2, bty='n')
```



This plot is quite different in that it shows the fraction of subjects *currently* in the PCM state. Looking at our multi-state diagram this is the fraction of subjects in the upper right PCM box. The curve goes up whenever someone enters the box (progression) and down when they leave (death). Myeloma survival was quite short during the era of this study and the proportion currently in the PCM state rarely rises above 2 percent.

The result of `print(mfit3)` reveals, as expected, less time spent in the PCM state. In the prior `mfit2` model, subjects who enter that state remain there for the duration; in this one they quickly pass through. It is worthwhile to check the `transitions` table in the output simply as a data check. In this case it shows subjects going from the entry (unnamed) state to PCM and death along with transitions from PCM to death. This is as expected. An error in creating the input data can lead to surprising counts and an even more surprising curve.

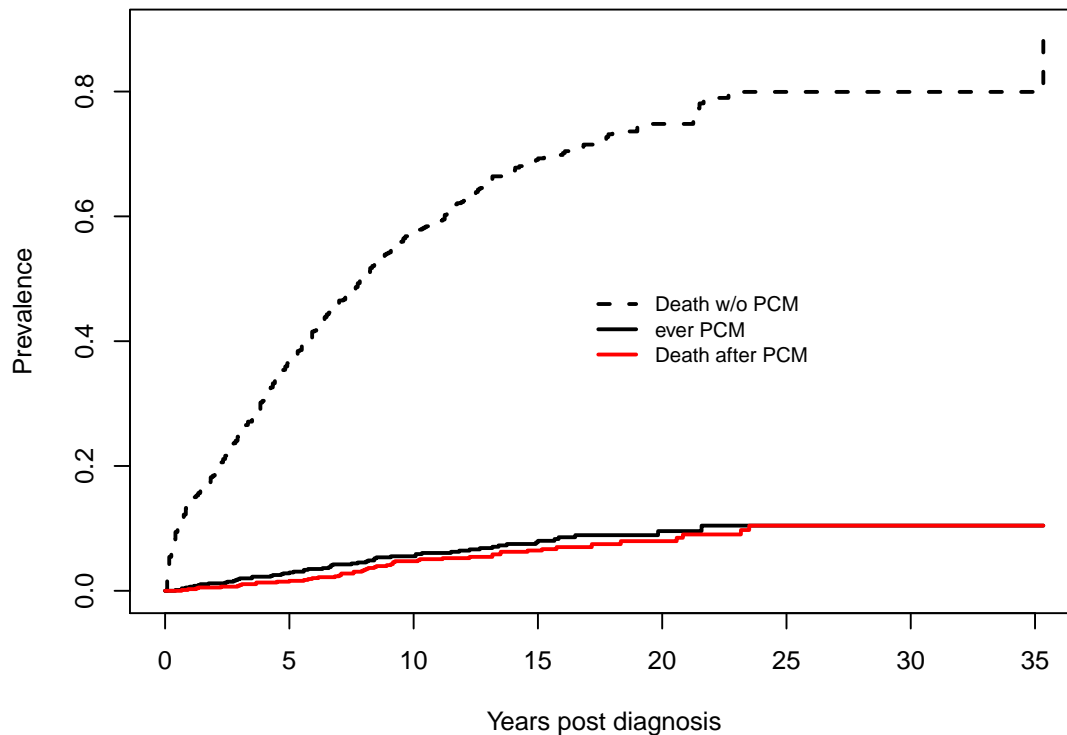
We have often found the three curve display below useful in the case of a transient state. It combines the results from competing risk model used above along with a second fit that treats death after PCM as a separate state from death before progression, the *multi-state 2* model of figure 2. In this plot the fraction of subjects currently in the PCM state is shown by the distance between the two curves. Only males are shown in the plot to minimize overlap.

```
> # Death after PCM will correspond to data rows with
> #  enum = 2 and event = death
> d2 <- with(newdata, ifelse(enum==2 & event=='death', 4, as.numeric(event)))
> e2 <- factor(d2, labels=c("censor", "pcm", "death w/o pcm",
                           "death after pcm"))
> mfit4 <- survfit(Surv(tstart, tstop, e2) ~ sex, data=newdata, id=id)
```

```

> plot(mfit2[2,], lty=c(1,2),
      xscale=12, mark.time=FALSE, lwd=2,
      xlab="Years post diagnosis", ylab="Prevalence")
> lines(mfit4[2,3], mark.time=FALSE, col=2, lty=1, lwd=2,
      conf.int=FALSE)
> legend(200, .5, c("Death w/o PCM", "ever PCM",
                    "Death after PCM"), col=c(1,1,2), lty=c(2,1,1),
      lwd=2, bty='n', cex=.82)

```



## 2.3 Further notes

The Aalen-Johansen method used by `survfit` does not account for interval censoring, also known as panel data, where a subject's current state is recorded at some fixed time such as a medical center visit but the actual times of transitions are unknown. Such data requires further assumptions about the transition process in order to model the outcomes and has a more complex likelihood. The `msm` package, for instance, deals with data of this type. If subjects reliably come in at regular intervals then the difference between the two results can be small, e.g., the `msm` routine estimates time until progression *occurred* whereas `survfit` estimates time until progression was *observed*.

- When using multi-state data to create Aalen-Johansen estimates, individuals are not allowed to have gaps in the middle of their time line. An example of this would be a data

set with (0, 30, pcm] and (50,70, death] as the two observations for a subject where the time from 30-70 is not accounted for.

- Subjects must stay in the same group over their entire observation time, i.e., variables on the right hand side of the equation cannot be time-dependent.
- A transition to the same state is allowed, e.g., observations of (0,50, 1], (50, 75, 3], (75, 89, 4], (89, 93, 4] and (93, 100, 4] for a subject who goes from entry to state 1, then to state 3, and finally to state 4. However, a warning message is issued for the data set in this case, since stuttering may instead be the result of a coding mistake. The same result is obtained if the last three observations were collapsed to a single row of (75, 100, 4].

### 3 Rate models

For simple two-state survival, the Cox model leads to three relationships

$$\lambda(t) = \lambda_0(t)e^{X\beta} \quad (4)$$

$$\Lambda(t) = \Lambda_0(t)e^{X\beta} \quad (5)$$

$$S(t) = \exp(-\Lambda(t)) \quad (6)$$

where  $\lambda$ ,  $\Lambda$  and  $S$  are the hazard, cumulative hazard and survival functions, respectively. There is a single linear predictor which governs both the rate  $\lambda$  (the arrow in figure 1) and probability of residing in the left hand box of the figure. For multi-state models this simplicity no longer holds; proportional hazards does not lead to proportional  $p(t)$  curves. The task before us is more complex.

The analysis of multi-state data has four key steps. In order of importance:

1. Draw a box and arrow figure describing the model.
2. Think through the rates (arrows).
  - (a) Which covariates should be attached to each rate? Sometimes a covariate is important for one transition, but not for another.
  - (b) For which transitions should one or more of the covariates be constrained to have the same coefficient? Sometimes there will be a biologic rationale for this. For other studies an equivalence is forced simply because we have too many unknowns and cannot accommodate them all. (This is the often the same reason that models contain very few interaction terms).
  - (c) Which, if any, of the transitions should share the same baseline hazard? Most of the time the baseline rates are all assumed to be different.
  - (d) Should there be random effects, and if so what is an appropriate correlation structure? Do some pairs of transitions have a shared effect, some pairs separate effects and others no random effect? Mixed effects Cox models tend to need larger sample size — does the data set have enough events?
3. Build an appropriate data set.

4. Fit the data. Examine multiple summaries of the model fit, including the predicted occupancy curves.

Step 1 is key to the entire endeavor. We saw in figure 2 and the examples above that multiple views of a multi-state process can be useful, and this will hold for modeling as well. Step 3 will often be the one that demands the most attention to detail.

### 3.1 MGUS example

Start with the simplest model for the MGUS data: a competing risks model (upper left diagram of figure 2), distinct baseline hazards for the two rates, no shared coefficients, and three covariates.

```
> options(show.signif.stars = FALSE) # display intelligence
> cfit2 <- coxph(Surv(etime, event=="death") ~ age + sex + mspike, mgus2)
> summary(cfit2, scale=c(10, 1, 1)) # scale age in decades

Call:
coxph(formula = Surv(etime, event == "death") ~ age + sex + mspike,
      data = mgus2)

n= 1373, number of events= 854
(11 observations deleted due to missingness)

              coef exp(coef)  se(coef)      z Pr(>|z|)
age      0.065200  1.067373  0.003649 17.869 < 2e-16
sexM     0.388899  1.475355  0.069928  5.561 2.68e-08
mspike   -0.059260  0.942461  0.063869 -0.928  0.353

              exp(coef) exp(-coef) lower .95 upper .95
age      1.9194      0.5210    1.7869    2.062
sexM     1.4754      0.6778    1.2864    1.692
mspike   0.9425      1.0611    0.8316    1.068

Concordance= 0.673 (se = 0.011 )
Rsquare= 0.247 (max possible= 1 )
Likelihood ratio test= 389.1 on 3 df, p=0
Wald test               = 333.4 on 3 df, p=0
Score (logrank) test = 334.8 on 3 df, p=0
>
```

The effect of age and sex on non-PCM mortality is profound, which is not a surprise given the median starting age of 72. Risk rises 1.9 fold per decade of age and the death rate for males is 1.5 times as great as that for females. The size of the serum monoclonal spike has almost no impact on non-PCM mortality. A 1 unit increase changes mortality by only 6%.

```
> cfit1 <- coxph(Surv(etime, event=="pcm") ~ age + sex + mspike, mgus2)
> cfit1
```

```
Call:
coxph(formula = Surv(etime, event == "pcm") ~ age + sex + mspike,
      data = mgus2)
```

	coef	exp(coef)	se(coef)	z	p
age	0.01635	1.01649	0.00837	1.95	0.051
sexM	-0.00503	0.99498	0.18845	-0.03	0.979
mspike	0.88408	2.42075	0.16521	5.35	8.7e-08

```
Likelihood ratio test=30.3 on 3 df, p=1.22e-06
```

```
n= 1373, number of events= 115
```

```
(11 observations deleted due to missingness)
```

```
> quantile(mgus2$mspike, na.rm=TRUE)
```

```
0% 25% 50% 75% 100%
```

```
0.0 0.6 1.2 1.5 3.0
```

The mspike size has a major impact on progression, however; each 1 gram change increases risk by 2.4 fold. The interquartile range of mspike is 0.9 gram so this risk increase is clinically important. The effect of age on the progression rate is much less pronounced, with a coefficient only 1/4 that for mortality, while the effect of sex on progression is completely negligible.

The effect of sex on the *lifetime* probability of PCM is not zero, however. Because of a longer lifetime, a female with MGUS will on average spend more total years at risk for PCM than the average male, and so has a larger lifetime risk of PCM. The average rate of progression is about 1% per year, as shown below, while the mean post diagnosis lifetime is 19 months longer for females. The overall effect is a 1.6% increase in lifetime risk.

```
> pfit1 <- pyears(Surv(ptime, pstat) ~ sex, mgus2, scale=12)
```

```
> round(100* pfit1$event/pfit1$pyears, 1) # PCM rate per year
```

```
sex
```

```
F M
```

```
1.1 1.0
```

```
> temp <- summary(mfit1, rmean="common") #print the mean survival time
```

```
> round(temp$table[,1:6], 1)
```

	records	n.max	n.start	events	*rmean	*se(rmean)
sex=F	631	631	631	423	142.4	6.1
sex=M	753	753	753	540	123.7	5.4

Notice that each `coxph` fit essentially ignores the other event type(s). In the figure, each rate (arrow) depends only on the box from which it originates and the events which it enumerates. Rates are instantaneous quantities, and depend only on the set of subjects who are at risk at a given moment; if someone is not at risk it really does not matter why.

When computing  $p(t)$ , on the other hand, all the rates must be considered at once. The Aalen-Johansen estimate applies as before, but now the individual entries  $\lambda_{ij}(t)$  in each cell of the transition matrix are taken from the relevant fit. As is also the case with predicted survival curves from a simple Cox model, predicted prevalence curves correspond to a set of prespecified covariate values. As an example we will generate the curves for four hypothetical subjects: male

and female, age 60 and 80, and serum m-spike of 1.2 grams. These are the approximate quartiles of age, and the median mspike.

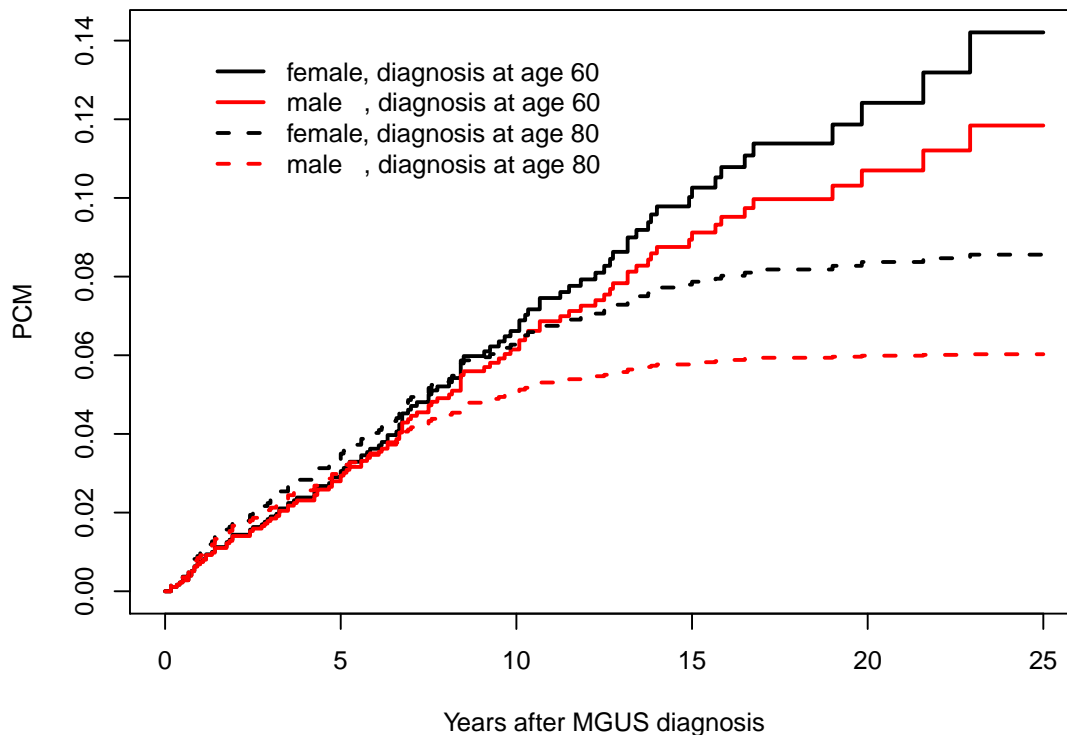
The Aalen-Johansen estimate for this simple 3-state competing risks setup works with a matrix of this form:

$$\begin{pmatrix} & \lambda_{12}t & \lambda_{13}(t) \\ 0 & & 0 \\ 0 & 0 & \end{pmatrix}$$

As before, the diagonal elements are chosen so that each row sums to 1. Standard survival curve calculations for a Cox model can be used to obtain  $\lambda_{12}$ , the rate of transition to the PCM state for our four subjects, and  $\lambda_{13}$  = the rate of transition to the “death before PCM” state. These are placed into a matrix and combined using a third call.

```
> newdata <- expand.grid(sex=c("F", "M"), age=c(60, 80), mspike=1.2)
> newdata
  sex age mspike
1  F  60    1.2
2  M  60    1.2
3  F  80    1.2
4  M  80    1.2
> temp <- matrix(list(), 3,3)
> dimnames(temp) <- list(from=c("Entry", "PCM", "Death"),
  to =c("Entry", "PCM", "Death"))
> temp[1,2] <- list(survfit(cfit1, newdata, std.err=FALSE))
> temp[1,3] <- list(survfit(cfit2, newdata, std.err=FALSE))
> csurv <- survfit(temp, p0 =c(1,0,0))
> plot(csurv[,2], xmax=25*12, xscale=12,
  xlab="Years after MGUS diagnosis", ylab="PCM",
  col=1:2, lty=c(1,1,2,2), lwd=2)
> legend(10, .14, outer(c("female", "male"),
  c("diagnosis at age 60", "diagnosis at age 80"),
  paste, sep=", "),
  col=1:2, lty=c(1,1,2,2), bty='n', lwd=2)
```





The individual survival curves that result from `survfit(cfit1)` and `survfit(cfit2)` are not actually of interest, since each is a Cox model analog of the `pcmbad` curve we criticized earlier. The cumulative hazard portion of the results is what is used to build an Aalen-Johansen estimate. (Calling `survfit` on a set of `survfit` objects is, I admit, a bit confusing. It would perhaps have been better to name the second routine “AalenJohansen”, but we use this often and didn’t want to type that long a name.)

Sex has nearly no effect on the hazard of PCM, i.e., on any given day the risk of conversion for a male is essentially the same as for a female of the same age. Yet we see above that the fitted Cox models predict a higher lifetime risk for females, and an age effect on lifetime risk that is far from proportional. Very few subjects acquire PCM more than 15 years after a MGUS diagnosis at age 80 for the obvious reason: very few of them will still be alive.

Creating the ‘list’ form matrix above is quite easy, in particular we do not need to fill in elements on the diagonal, nor those for which no transitions occur, e.g., from death back to the entry state. The resulting `survfit` object is easy to plot or print using standard calls. The approach has a number of caveats, however.

- It does not produce standard errors for the curves, as a consequence of being two steps removed from the data.
- It is easy to “fool” the program. For instance if you were to get curves for females and males from `cfit1`, but the curves from `cfit2` were in the reverse order of male then female, results will still be produced but they would not be valid. The user is responsible for setting the problem up correctly.

- The R syntax for a matrix of lists is rather fussy, e.g., you can't leave the `list` function out of the lines that assign elements to `temp` above.

The `mstate` package addresses these issues, at the price of a somewhat more complex syntax.

```
> # Print out a M/F results at 20 years
> temp <- summary(csurv, time=20*12)$prev
> cbind(newdata, PCM= round(100*temp[,2], 1))
```

	sex	age	mspike	PCM
1	F	60	1.2	12.4
2	M	60	1.2	10.7
3	F	80	1.2	8.4
4	M	80	1.2	6.0

The above table shows that females are modeled to have a higher risk of 20 year progression, even though their hazard at any given moment is nearly identical to males. The difference at 20 years is on the order of our “back of the napkin” person-years estimate of 1% progression per year \* 1.7 more years of life for the females, but the progression fraction varies substantially by group.

## 4 Fine-Gray model

For the competing risk case the Fine-Gray model provides an alternate way of looking at the data. As we saw above, the impact of a particular covariate on the final prevalence values  $P$  can be complex, even if the models for the hazards are relatively simple. The primary idea of the Fine-Gray approach is to turn the multi-state problem into a collection of two-state ones. In the upper right diagram of figure 1, draw a circle around all of the states except the chosen endpoint and collapse them into a single meta-state. For the MGUS data these are

- Model 1
  - left box: All subjects in the entry or “death first” state
  - right box: PCM
- Model 2
  - left box: All subjects in the entry or “PCM first” state
  - right box: Death (without PCM)

An interesting aspect of this is that the fit can be done as a two stage process: the first stage creates a special data set while the second fits a weighted `coxph` or `survfit` model to the data. The data set can be created using the `finegray` command.

```
> # (first three lines are identical to an earlier section)
> etime <- with(mgus2, ifelse(pstat==0, futime, ptime))
> event <- with(mgus2, ifelse(pstat==0, 2*death, 1))
> event <- factor(event, 0:2, labels=c("censor", "pcm", "death"))
```

```

> pcmdat <- finegray(Surv(etime, event) ~ ., data=mgus2,
                     etype="pcm")
> pcmdat[1:4, c(1:3, 11:14)]
   id age sex fgstart fgstop fgstatus   fgwt
1  1  88  F      0     35         0 1.0000000
2  1  88  F     35     44         0 0.9990449
3  1  88  F     44     47         0 0.9980368
4  1  88  F     48     52         0 0.9959629
> deathdat <- finegray(Surv(etime, event) ~ ., data=mgus2,
                      etype="death")
> dim(pcmdat)
[1] 41775    14
> dim(deathdat)
[1] 12910    14
> dim(mgus2)
[1] 1384    10

```

The `finegray` command has been used to create two data sets: one for the PCM endpoint and one for the death endpoint. In each, four new variables have been added containing a survival time (`fgstart`, `fgstop`, `fgstatus`) with an ‘ordinary’ status of 0/1, along with a case weight and a large number of new rows. We can use this new data set as yet another way to compute multi-state survival curves, though there is no good reason to use this rather roundabout approach instead of the simpler `survfit(Surv(etime, event) ~ sex)`.

```

> # The PCM curves of the multi-state model
> pfit2 <- survfit(Surv(fgstart, fgstop, fgstatus) ~ sex,
                  data=pcmdat, weight=fgwt)
> # The death curves of the multi-state model
> dfit2 <- survfit(Surv(fgstart, fgstop, fgstatus) ~ sex,
                  data=deathdat, weight=fgwt)

```

The two new curves are almost identical to the prior estimates, and in fact would be identical if we had accounted for the slightly different censoring patterns in males and females (by adding `strata(sex)` to the right hand side of the `finegray` formulas).

A Cox model fit to the constructed data set yields the Fine-Gray models for PCM and for death.

```

> fgfit1 <- coxph(Surv(fgstart, fgstop, fgstatus) ~ sex, data=pcmdat,
                 weight= fgwt)
> summary(fgfit1)
Call:
coxph(formula = Surv(fgstart, fgstop, fgstatus) ~ sex, data = pcmdat,
      weights = fgwt)

n= 41775, number of events= 115

```

```

      coef exp(coef) se(coef)      z Pr(>|z|)
sexM -0.2290    0.7953    0.1866 -1.228    0.22

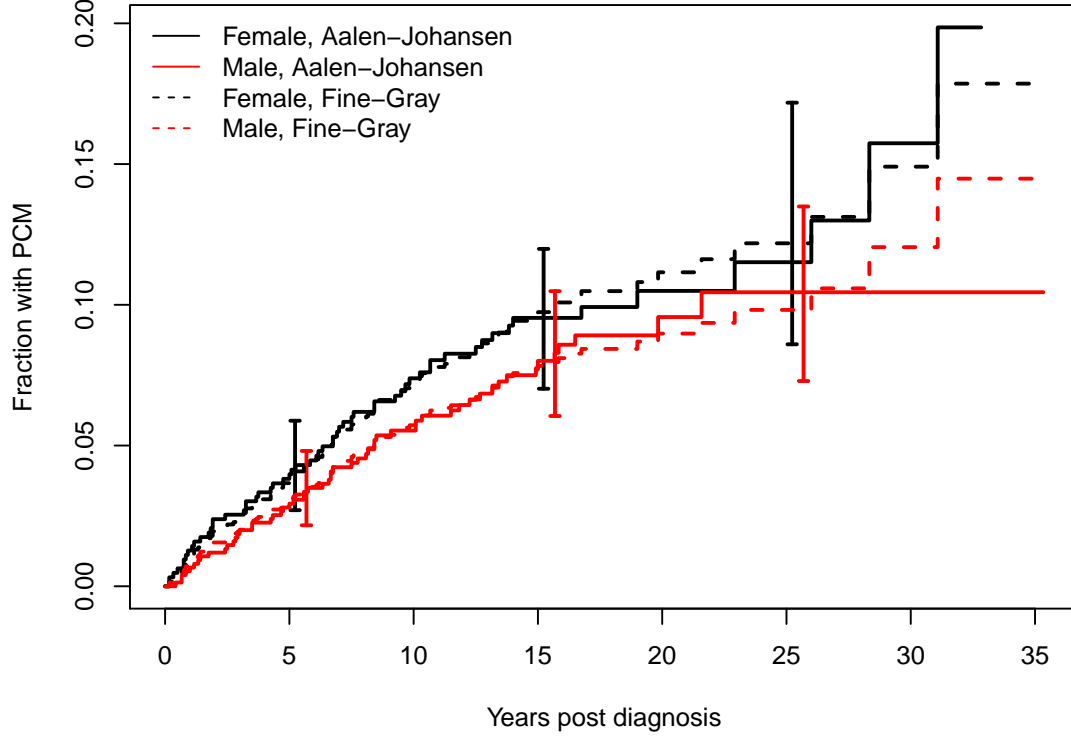
      exp(coef) exp(-coef) lower .95 upper .95
sexM    0.7953      1.257    0.5517    1.146

Concordance= 0.528 (se = 0.024 )
Rsquare= 0 (max possible= 0.037 )
Likelihood ratio test= 1.51 on 1 df, p=0.2196
Wald test = 1.51 on 1 df, p=0.2196
Score (logrank) test = 1.51 on 1 df, p=0.2186
> fgfit2 <- coxph(Surv(fgstart, fgstop, fgstatus) ~ sex, data=deathdat,
                  weight= fgwt)
> fgfit2
Call:
coxph(formula = Surv(fgstart, fgstop, fgstatus) ~ sex, data = deathdat,
      weights = fgwt)

      coef exp(coef) se(coef)      z      p
sexM 0.2335    1.2630    0.0689 3.39 0.00071

Likelihood ratio test=11.6 on 1 df, p=0.00067
n= 12910, number of events= 860
> mfit2 <- survfit(Surv(etime, event) ~ sex, data=mgus2) #reprise the AJ
> plot(mfit2[,1], col=1:2,
      lwd=2, xscale=12,
      conf.times=c(5, 15, 25)*12,
      xlab="Years post diagnosis", ylab="Fraction with PCM")
> ndata <- data.frame(sex=c("F", "M"))
> fgssurv1 <- survfit(fgfit1, ndata)
> lines(fgssurv1, fun="event", lty=2, lwd=2, col=1:2)
> legend("topleft", c("Female, Aalen-Johansen", "Male, Aalen-Johansen",
                    "Female, Fine-Gray", "Male, Fine-Gray"),
      col=1:2, lty=c(1,1,2,2), bty='n')
> # rate models with only sex
> pfitr <- coxph(Surv(etime, event=="pcm") ~ sex, mgus2)
> dfitr <- coxph(Surv(etime, event=="death") ~ sex, mgus2)
> temp <- matrix(list(), 3,3)
> temp[1,2] <- list(survfit(pfitr, ndata, std.err=FALSE))
> temp[1,3] <- list(survfit(dfitr, ndata, std.err=FALSE))
> rcurve <- survfit(temp, p0=c(entry=1, pcm=0, death=0))

```



The FG model states that males have a lower rate of *observed* PCM, by a factor of 0.8, and that this hazard ratio is constant over time. An overlaid plot of the non-parametric Aalen-Johansen estimates for the PCM state (from `survfit`) along with predicted curves from the Fine-Gray model show that proportional hazards is not unreasonable for this particular fit. The predicted values from the rate model, computed just above but not plotted on the curve, also fit well with the data.

When there is only a single categorical 0/1 covariate the Fine-Gray model reduces to Gray's test of the subdistribution function, in the same way that a `coxph` fit with a single categorical predictor is equivalent to the log-rank test.

The mathematics behind the Fine-Gray estimate starts with the functions  $F_k(t) = p_k(t)$ , where  $p$  is the probability in state function estimated by the AJ estimate. This can be thought of as the distribution function for the improper random variable  $T^* = I(\text{event type} = k)T + I(\text{event type} \neq k)\infty$ . Fine and Gray refer to  $F_k$  as a subdistribution function. In analogy to the survival probability in the two state model define

$$\gamma_k(t) = -d \log[1 - F_k(t)]/dtI \quad (7)$$

and assume that  $\gamma_k(t; x) = \gamma_{k0}(t) \exp(X\beta)$ . In a 2 state alive  $\rightarrow$  death model,  $\gamma$  becomes the usual hazard function  $\lambda$ . In the same way that our multivariate Cox model `cfit2` made the simplifying assumption that the impact of male sex is to increase the hazard for death by a factor of 1.48, independent of the subject's age or serum mspike value, the Fine-Gray model assumes that each covariate's effect on  $\log(1 - F)$  is a constant, independent of other variables. Both model's assumptions are wonderfully simplifying with respect to understanding a covariate,

since we can think about each one separately from the others. This is, of course, under the assumption that the model is correct: additivity across covariates, linearity, and proportional hazards all hold. In a multi-state model, however, these assumptions cannot hold for both the per-transition and Fine-Gray models formulations at the same time; if it is true for one, it will not be true for the other.

Now consider a multivariate fit on age, sex, and serum m-spike.

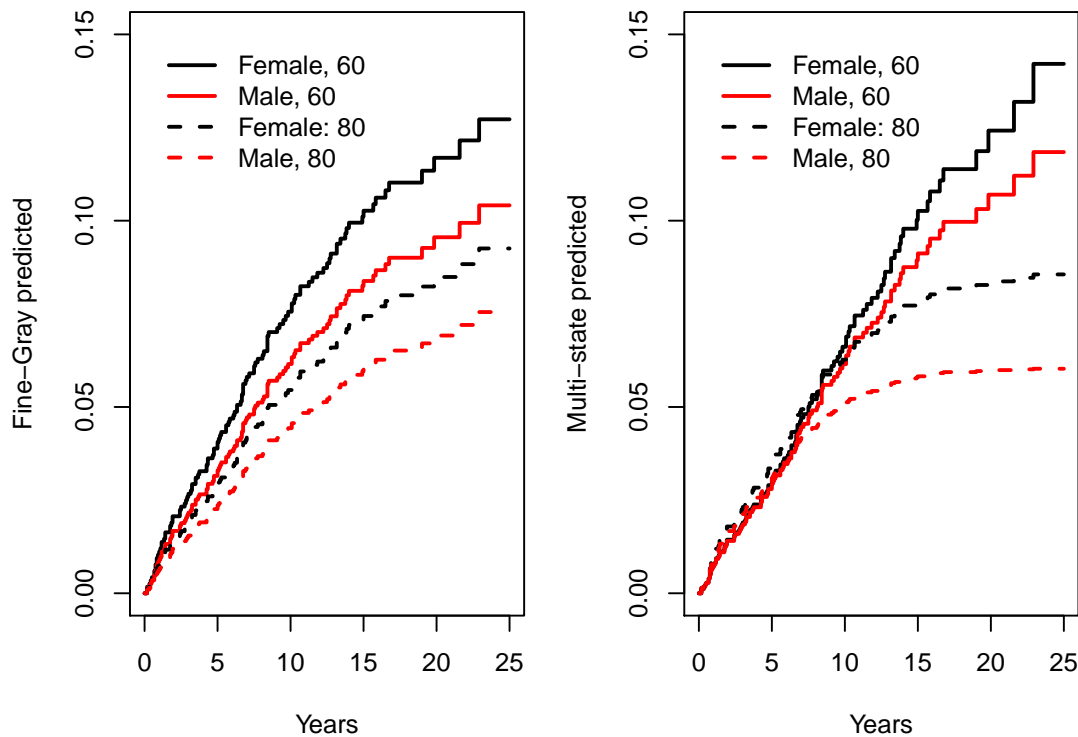
```
> fgfit2a <- coxph(Surv(fgstart, fgstop, fgstatus) ~ age + sex + mspike,
  data=pcmdat, weight=fgwt)
> fgfit2b <- coxph(Surv(fgstart, fgstop, fgstatus) ~ age + sex + mspike,
  data=deathdat, weight=fgwt)
> round(rbind(PCM= coef(fgfit2a), death=coef(fgfit2b)), 3)

      age  sexM mspike
PCM   -0.017 -0.213  0.888
death  0.059  0.369 -0.154
```

The Fine-Gray fits show an effect of all three variables on the subdistribution rates. Males have a lower lifetime risk of PCM before death and a higher risk of death before PCM, while a high serum m-spike works in the opposite direction. The Cox models showed no effect of sex on the instantaneous hazard of PCM and none for serum m-spike on the death rate. However, as shown in the last section, the Cox models do predict a greater lifetime risk for females. We had also seen that older subjects are less likely to experience PCM due to the competing risk of death; this is reflected in the FG model as a negative coefficient for age.

Now compute predicted survival curves for the model, and show them alongside the predictions from the multi-state model.

```
> oldpar <- par(mfrow=c(1,2))
> newdata <- expand.grid(sex= c("F", "M"), age=c(60, 80), mspike=1.2)
> fsurv1 <- survfit(fgfit2a, newdata) # time to progression curves
> plot(fsurv1, xscale=12, col=1:2, lty=c(1,1,2,2), lwd=2, fun='event',
  xlab="Years", ylab="Fine-Gray predicted",
  xmax=12*25, ylim=c(0, .15))
> legend(1, .15, c("Female, 60", "Male, 60", "Female: 80", "Male, 80"),
  col=c(1,2,1,2), lty=c(1,1,2,2), lwd=2, bty='n')
> plot(csurv[,2], xscale=12, col=1:2, lty=c(1,1,2,2), lwd=2,
  xlab="Years", ylab="Multi-state predicted",
  xmax=12*25, ylim=c(0, .15))
> legend(1, .15, c("Female, 60", "Male, 60", "Female: 80", "Male, 80"),
  col=c(1,2,1,2), lty=c(1,1,2,2), lwd=2, bty='n')
> par(oldpar)
```



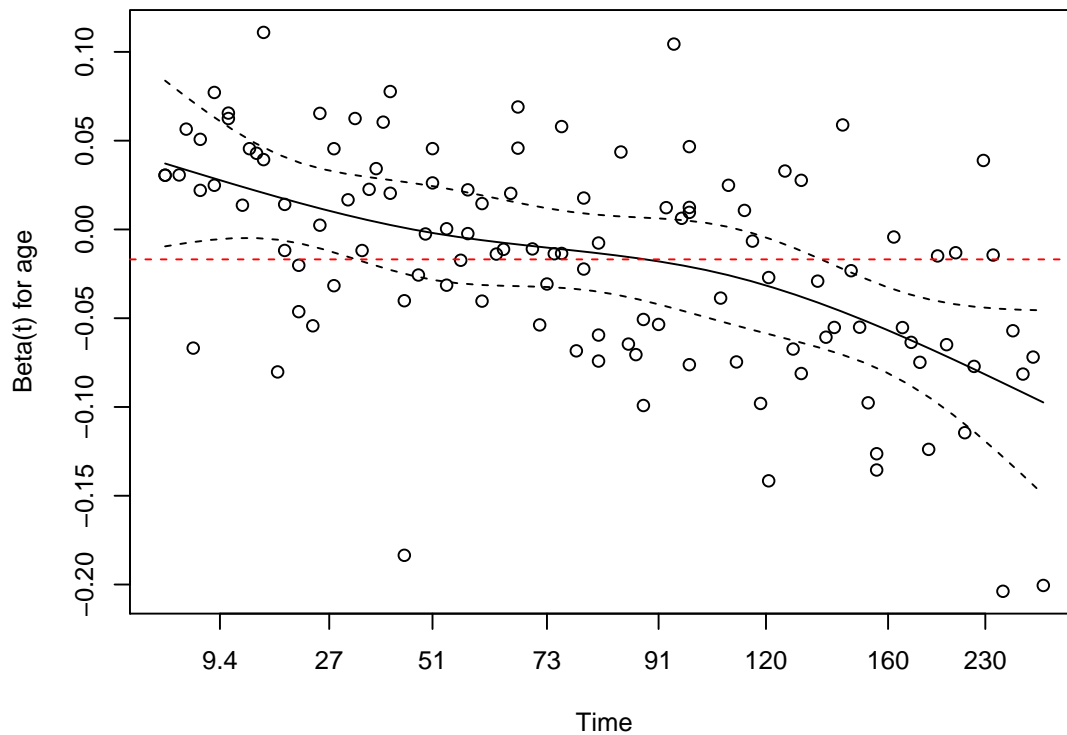
The predictions as a function of age group are quite different for the Fine-Gray model: new PCM cases are predicted 20+ years after diagnosis in both the old and young age groups, while they are predicted to cease in the multi-state fit. The average of the curves is nearly the same at each age, but the global proportional hazards assumption of the FG model forces the curves to remain parallel.

We can check the proportional hazards assumption of the models using the `cox.zph` function, linearity of the continuous variables age and mspike by using non-linear terms such as `pspline` or `ns`, and additivity by exploring interactions. All are obvious and important next steps. For instance, the proportional hazards assumption for age shows clear violations.

```
> zph.fgfit2a <- cox.zph(fgfit2a)
> zph.fgfit2a
```

	rho	chisq	p
age	-0.5306	20.9939	4.61e-06
sexM	0.0215	0.0528	8.18e-01
mspike	-0.0267	0.0788	7.79e-01
GLOBAL	NA	21.3108	9.07e-05

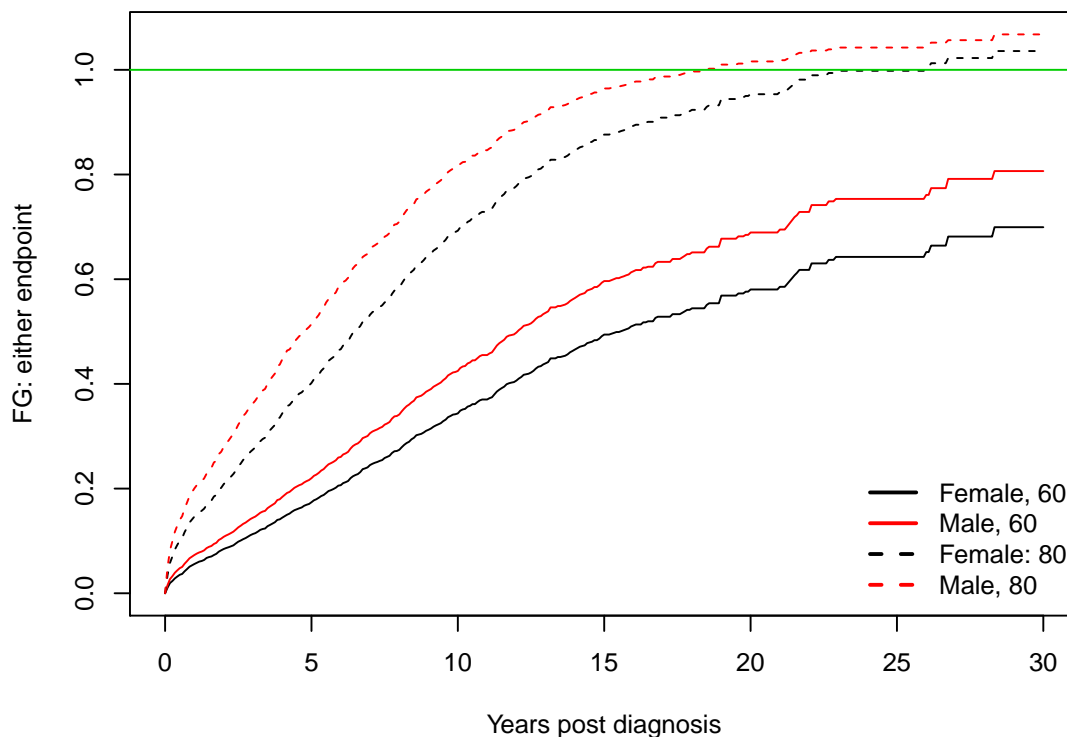
```
> plot(zph.fgfit2a[1])
> abline(h=coef(fgfit2a)[1], lty=2, col=2)
```



A weakness of the Fine-Gray approach is that since the two endpoints are modeled separately, the results do not have to be consistent. Below is a graph of the predicted fraction who have experienced neither endpoint. For subjects diagnosed at age 80 the Fine-Gray models predict that more than 100% will either progress or die by 30 years. Predictions based on the Aalen-Johansen approach do not have this issue.

```
> fsurv2 <- survfit(fgfit2b, newdata) # time to progression curves
> xtime <- 0:(30*12) #30 years
> y1a <- 1 - summary(fsurv1, times=xtime)$surv #predicted pcm
> y1b <- 1 - summary(fsurv2, times=xtime)$surv #predicted deaths before pcm
> y1 <- (y1a + y1b) #either
> matplot(xtime/12, y1, col=1:2, lty=c(1,1,2,2), type='l',
          xlab="Years post diagnosis", ylab="FG: either endpoint")
> abline(h=1, col=3)
> legend("bottomright", c("Female, 60", "Male, 60", "Female: 80", "Male, 80"),
        col=c(1,2,1,2), lty=c(1,1,2,2), lwd=2, bty='n')
```





The primary strength of the Fine-Gray model with respect to the Cox model approach is that if lifetime risk is a primary question, then the model has given us a simple and digestible answer to that question: “females have a 1.2 fold higher chance of PCM, after adjustment for age and serum m-spike”. This simplicity is not without a price, however, and these authors are not proponents of the approach. There are five issues.

1. The attempt to capture a complex process as a single value is grasping for a simplicity that does not exist for many (perhaps most) data sets. The necessary assumptions in a multivariate Cox model of proportional hazards, linearity of continuous variables, and no interactions are strong ones. For the FG model these need to hold for a combined process — the mixture of transition rates to each endpoint — which turns out to be a more difficult barrier.
2. The sum of predictions need not be consistent.
3. From the per-transition models one can work forward and compute  $p(t)$ , the occupancy probabilities for each state over time; both the hazard ratios and  $p$  are useful summaries of the data. We don’t have tools to work backwards from a Fine-Gray fit to the per transition hazards.
4. The approach is viable only for competing risks and not for other multi-state models.
5. The risk sets are odd.

The last of these is perhaps the most frequently listed issue with the Fine-Gray model, but it is actually a minor complaint. The state probabilities  $p(t)$  in a multi-state model are implicitly fractions of the total population we started with: someone who dies in month 1 is still a part of the denominator for the fraction of subjects with PCM at 20 years. In the Fine-Gray formulas this subject explicitly appears in risk set denominators at a later time, which looks odd but is more of an artifact.

The first issue is substantial, however, and checking the model assumptions of a Fine-Gray fit is mandatory. The second point is alarming, but it does not have a practical impact unless there is long follow-up.

## 5 Stacked data sets

How does one fit risk models that have shared coefficients or baseline hazards? One approach is to fit the set of Cox models for the rates ‘all at once’ on a combined data set. For the simple competing risks MGUS fit above, assume that we wanted to add hemoglobin to the fit, with a common coefficient for both the PCM and death endpoint. (Anemia is a feature of both PCM and old age.) Create a stacked data set with  $2n$  observations. The first  $n$  rows are the data set we would use for a time to PCM analysis, with a simple 0/1 status variable encoding the PCM outcome. The second  $n$  rows are the data set we would have used for the ‘death before PCM’ fits, with status encoding the death-before-PCM endpoint. A last variable, `group`, is ‘pcm’ for the first  $n$  observations and ‘death’ for the remainder. Then fit a model

```
> temp1 <- data.frame(mgus2, time=etime, status=(event=="pcm"), group='pcm')
> temp2 <- data.frame(mgus2, time=etime, status=(event=="death"), group="death")
> stacked <- rbind(temp1, temp2)
> allfit <- coxph(Surv(time, status) ~ hgb + (age + sex)*strata(group),
                  data=stacked)
```

This fits a common effect for hemoglobin (hgb) but separate age and sex effects for the two endpoints, along with separate baseline hazards.

## 6 Reproducing the output in “Tutorial in biostatistics: Competing risks and multi-state models”

An excellent introduction to multi-state models is found in Putter, Fiocco and Geskus, *Tutorial in biostatistics: Competing risks and multi-state models* [1]. In this section we recreate the graphs and tables from the paper; it parallels a similar document that is a vignette in the `mstate` package. The following text will make the most sense if it is read in parallel with the paper.

The tutorial uses two data sets which are included in the `mstate` package. The first is data from 329 homosexual men from the Amsterdam Cohort Studies on HIV infection and AIDS. A competing risks analysis is done with the appearance of syncytium inducing (SI) phenotype and AIDS diagnosis as the endpoints. The second data set is from the European Blood and Marrow Transplant (EBMT) registry, and follows 2204 subjects who had a transplant forward to relapse or death, with platelet recovery as an intermediate state.

```

> require(mstate, quietly = TRUE)
> data(aidssi) # AIDS data set
> data(ebmt3)  # transplant data set

```

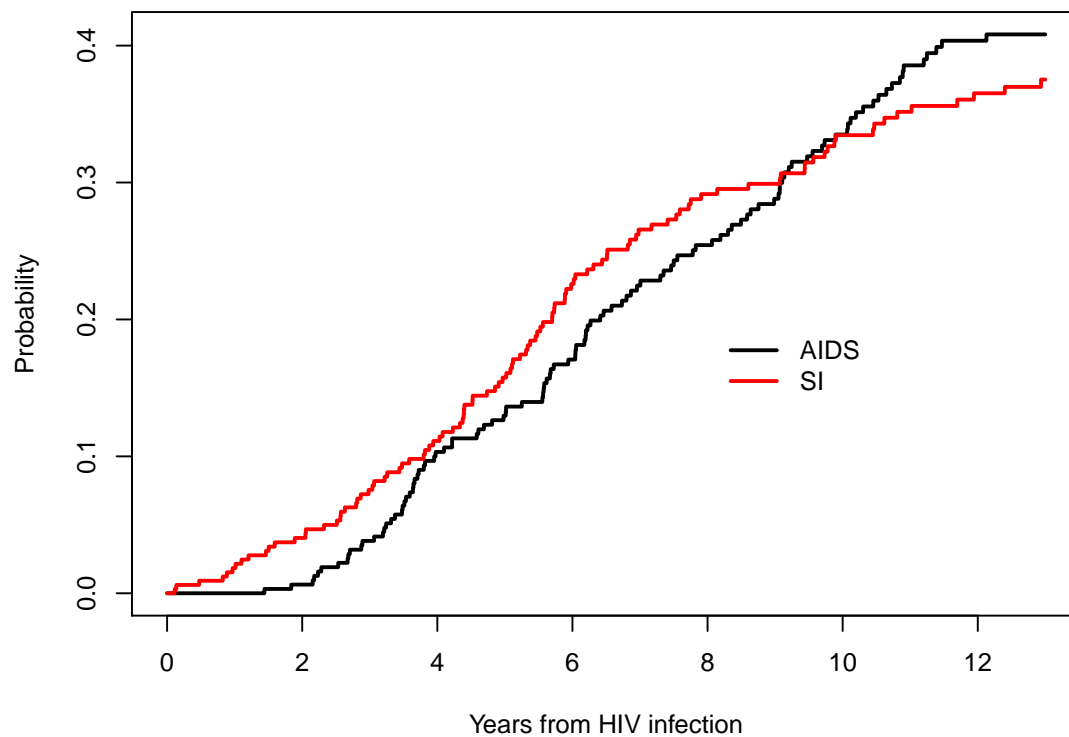
## 6.1 Competing risk

We first create the multi-state status variable and use it to plot the competing risk curves for the outcome.

```

> adata <- aidssi
> adata$event <- factor(adata$status, 0:2, c("censored", "AIDS", "SI"))
> # KM curves that censor the other endpoint (a bad idea)
> bad1 <- survfit(Surv(time, event=="AIDS") ~ 1, adata)
> bad2 <- survfit(Surv(time, event=="SI") ~1, adata)
> # The correct Aalen-Johansen curves
> ajfit <- survfit(Surv(time, event) ~1, adata)
> ajfit$transitions
      to
from  AIDS  SI
  AIDS    0   0
   SI     0   0
      114 108
> plot(ajfit, xmax=13, col=1:2, lwd=2,
       xlab="Years from HIV infection", ylab="Probability")
> legend(8, .2, c("AIDS", "SI"), lty=1, lwd=2, col=1:2, bty='n')

```



We will use "Txx" to stand for figures or page numbers in the tutorial. figure T2 shows the two Kaplan-Meier curves, with one going uphill and the other downhill. The fact that they cross is used to emphasize the inconsistency of the two estimates, i.e., that they add to more than 1.0.

```
> # re-create figure T2
> plot(bad1, conf.int=FALSE, xmax=13,
       xlab="Years from HIV infection", ylab="Probability")
> lines(bad2, conf.int=FALSE, fun='event', xmax=13)
> text(c(8,8), c(.8, .22), c("AIDS", "SI"))
```

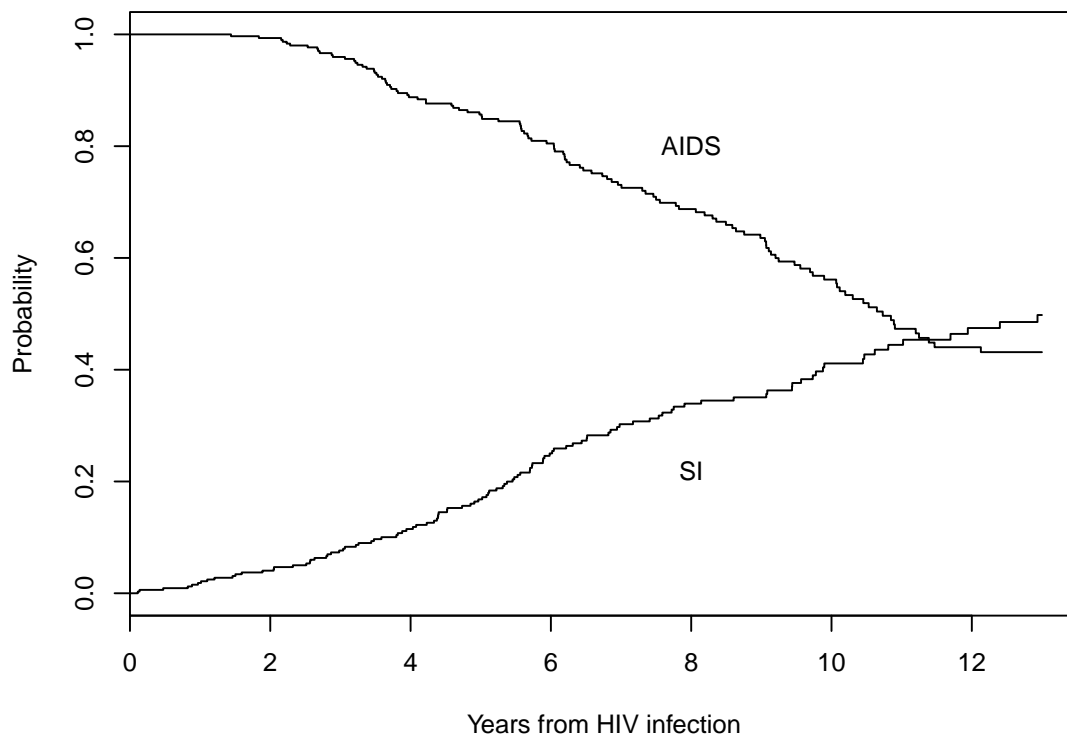


Figure T3 adds the Aalen-Johansen curves on top of this. Plotting the competing risk curve for AIDS “down from the top” is something we do not recommend, since it is not “survival free of AIDS” which is what a naive reader of such a curve would expect.

```
> # re-create figure T3
> plot(bad1, conf.int=FALSE, xmax=13, col="lightgray",
       xlab="Years from HIV infection", ylab="Probability")
> lines(bad2, conf.int=FALSE, fun='event', xmax=13, col='lightgray')
> text(c(8,8), c(.8, .22), c("AIDS", "SI"))
> lines(ajfit[,2], conf.int=FALSE, lwd=2, xmax=13)
> lines(ajfit[,1], conf.int=FALSE, lwd=2, xmax=13, fun=function(x) x)
```

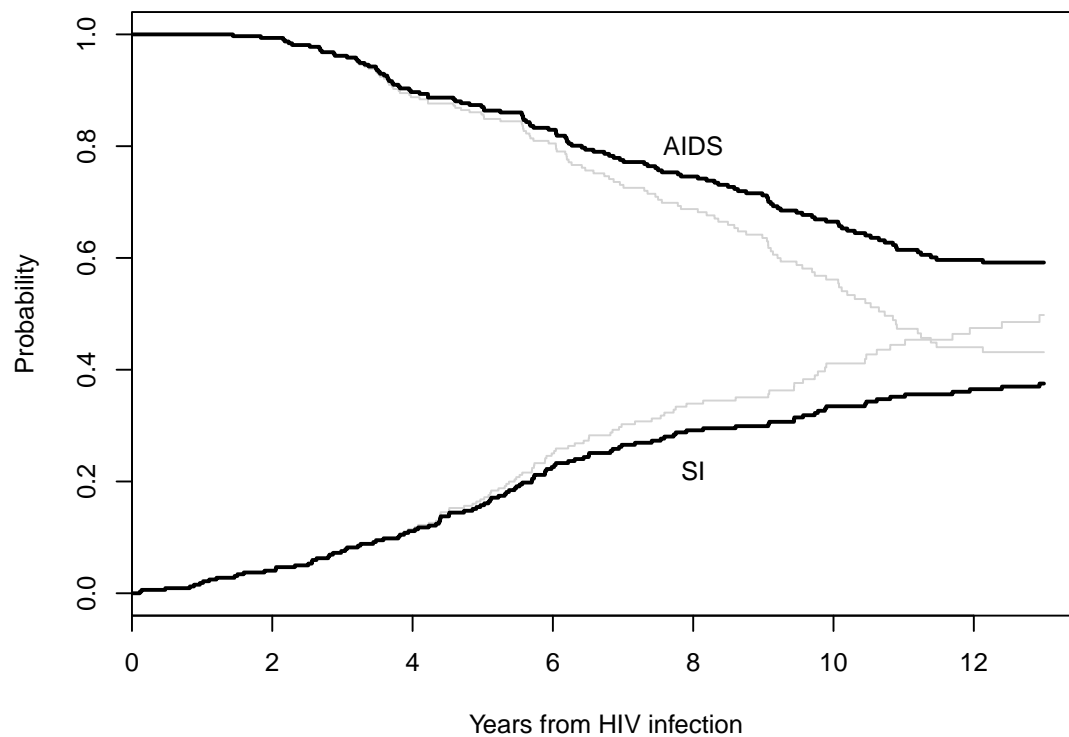
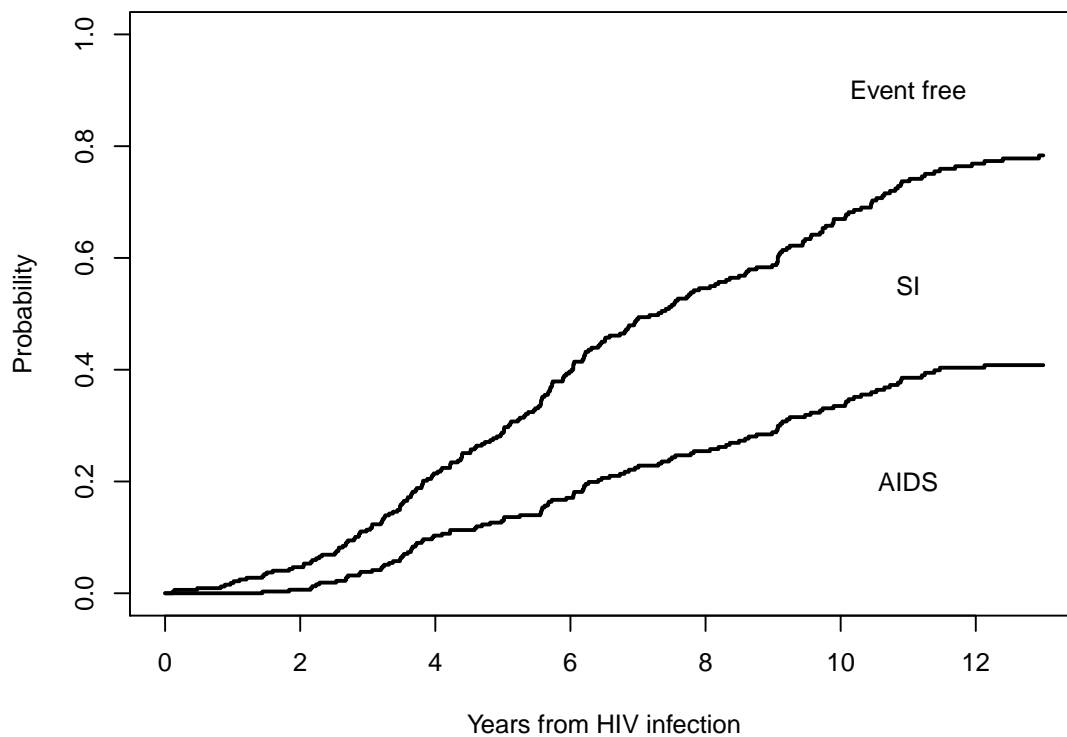


Figure T4 is a stacked plot of the competing risks outcome. We can modify the `ajfit` object by replacing each row of the `prev` component with the cumulative sum, and then plot the result.

```
> # re-create figure T4
> temp <- ajfit
> temp$prev <- t(apply(temp$prev, 1, cumsum)) # apply() transposes
> plot(temp, xmax=13, lwd=2, col=1, ylim=c(0,1),
       xlab="Years from HIV infection", ylab="Probability")
> text(c(11, 11, 11), c(.2, .55, .9), c("AIDS", "SI", "Event free"))
```



The tutorial then fits separate Cox models to the two rates, page T2404.

```
> cfit1 <- coxph(Surv(time, event=="AIDS") ~ ccr5, adata)
> cfit1
Call:
coxph(formula = Surv(time, event == "AIDS") ~ ccr5, data = adata)
```

	coef	exp(coef)	se(coef)	z	p
ccr5WM	-1.236	0.291	0.307	-4.02	5.7e-05

Likelihood ratio test=22 on 1 df, p=2.76e-06

n= 324, number of events= 113

(5 observations deleted due to missingness)

```
> cfit2 <- coxph(Surv(time, event=="SI") ~ ccr5, adata)
> cfit2
```

```
Call:
coxph(formula = Surv(time, event == "SI") ~ ccr5, data = adata)
```

	coef	exp(coef)	se(coef)	z	p
ccr5WM	-0.254	0.776	0.238	-1.07	0.29

Likelihood ratio test=1.19 on 1 df, p=0.275

```
n= 324, number of events= 107
  (5 observations deleted due to missingness)
```

We can achieve the same result using a stacked data set, which will have 329 rows = number of subjects for the AIDS endpoint, followed by 329 rows for the SI endpoint. We had to be a bit cautious since the tutorial uses `cause` for the event type and the data set `aidsii` already has a variable by that name; hence the initial subset call.

```
> temp <- subset(adata, select=c(time, ccr5))
> temp1 <- data.frame(temp, status= 1*(adata$event=="AIDS"), cause="AIDS")
> temp2 <- data.frame(temp, status= 1*(adata$event=="SI"), cause="SI")
> stack <- rbind(temp1, temp2)
> cfit3 <- coxph(Surv(time, status) ~ ccr5 * strata(cause), data=stack)
> cfit3
```

Call:

```
coxph(formula = Surv(time, status) ~ ccr5 * strata(cause), data = stack)
```

	coef	exp(coef)	se(coef)	z
ccr5WM	-1.236	0.291	0.307	-4.02
ccr5WM:strata(cause)SI	0.982	2.669	0.389	2.53

	p
ccr5WM	5.7e-05
ccr5WM:strata(cause)SI	0.012

Likelihood ratio test=23.2 on 2 df, p=9.29e-06

```
n= 648, number of events= 220
```

```
(10 observations deleted due to missingness)
```

```
> sum(coef(cfit3))
```

```
[1] -0.2542042
```

The use of an interaction term gives a different form for the coefficients; the second is now the difference in CCR-5 effect between the two endpoints. Which form one prefers is a matter of taste. In the tutorial, the stacked data set also has individual indicator variables for `ccr` within type. We can reprise that approach instead of using the interaction to get yet a third equivalent fit.

```
> stack$ccr5.1 <- (stack$ccr5=="WM") * (stack$cause == "AIDS")
> stack$ccr5.2 <- (stack$ccr5=="WM") * (stack$cause == "SI")
> coxph(Surv(time, status) ~ ccr5.1 + ccr5.2 + strata(cause), stack)
```

Call:

```
coxph(formula = Surv(time, status) ~ ccr5.1 + ccr5.2 + strata(cause),
      data = stack)
```

	coef	exp(coef)	se(coef)	z	p
ccr5.1	-1.236	0.291	0.307	-4.02	5.7e-05
ccr5.2	-0.254	0.776	0.238	-1.07	0.29



```

Likelihood ratio test=23.2 on 2 df, p=9.29e-06
n= 648, number of events= 220
(10 observations deleted due to missingness)

```

The model `cfit4` assumes that the baseline hazards for the two endpoints are proportional.

```

> cfit4 <- coxph(Surv(time, status) ~ ccr5.1 + ccr5.2 + cause, stack)
> cfit4

```

Call:

```

coxph(formula = Surv(time, status) ~ ccr5.1 + ccr5.2 + cause,
      data = stack)

```

	coef	exp(coef)	se(coef)	z	p
ccr5.1	-1.166	0.311	0.306	-3.81	0.00014
ccr5.2	-0.332	0.718	0.237	-1.40	0.16112
causeSI	-0.184	0.832	0.148	-1.25	0.21201

```

Likelihood ratio test=21.5 on 3 df, p=8.12e-05
n= 648, number of events= 220
(10 observations deleted due to missingness)

```

```

> summary(cfit4)$coefficients

```

	coef	exp(coef)	se(coef)	z
ccr5.1	-1.1663939	0.3114882	0.3063380	-3.807539
ccr5.2	-0.3315902	0.7177814	0.2366264	-1.401324
causeSI	-0.1842985	0.8316875	0.1476677	-1.248062

Pr(>|z|)

ccr5.1	0.0001403566
ccr5.2	0.1611173377
causeSI	0.2120082240

```

> cfit4b <- coxph(Surv(time, status) ~ ccr5*cause, stack) #same result

```

We can now generate the Aalen-Johansen curves for `cfit4`. This starts with obtaining the predictions for the two levels of CCR-5 and for the two endpoints, which results in four curves. The first two are for AIDS and the second two for SI. The temporary matrix `smat` can have the states in any order we wish (as long as we plug our curves into the right position!); we have chosen the order found above in the non-parametric curves `fit2`.

```

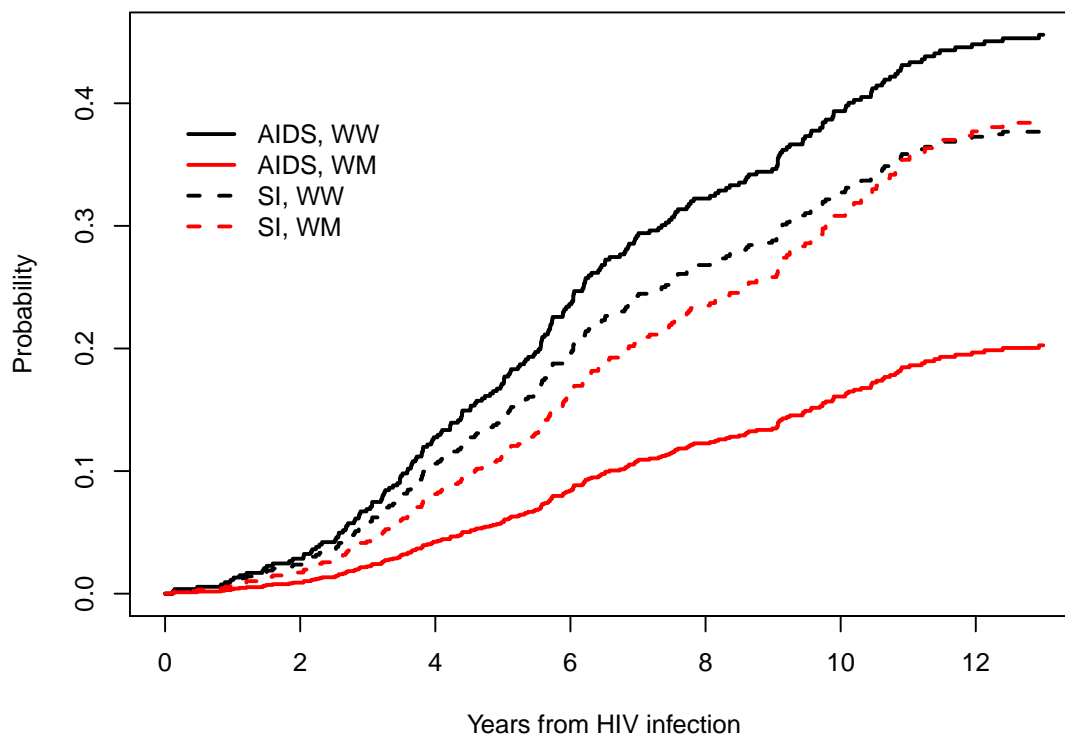
> # re-create figure T5 in a single panel
> tdata <- expand.grid(ccr5=c("WW","WM"), cause=c("AIDS", "SI"))
> tdata
  ccr5 cause
1   WW  AIDS
2   WM  AIDS
3   WW   SI
4   WM   SI

```

```

> tsurv <- survfit(cfit4b, newdata=tdata)
> smat <- matrix(list(), 3, 3,
                  dimnames=list(from= c("AIDS", "SI", "entry"),
                                to = c("AIDS", "SI", "entry")))
> smat[3,1] <- list(tsurv[1:2])
> smat[3,2] <- list(tsurv[3:4])
> smat      #did we put things in the right place?
      to
from  AIDS    SI    entry
AIDS  NULL    NULL  NULL
SI    NULL    NULL  NULL
entry List,14 List,14 NULL
> csurv1 <- survfit(smat, p0=c(0,0,1))
> plot(csurv1[,1:2], col=1:2, lty=c(1,1,2,2), xmax=13, lwd=2,
       xlab="Years from HIV infection", ylab="Probability")
> legend(0, .4, c("AIDS, WW", "AIDS, WM", "SI, WW", "SI, WM"),
       col=1:2, lty=c(1,1,2,2), lwd=2, bty='n')

```



Because the entry state has a name, the plot command would have shown all 3 curves by default; the 1:2 subscript restricts it to the first two.

### 6.1.1 Fine-Gray fits

We can also fit Fine-Gray models for AIDS and SI appearance.

```
> fdata1 <- finegray(Surv(time, event) ~ ., adata, etype='AIDS')
> fgfit1 <- coxph(Surv(fgstart, fgstop, fgstatus) ~ ccr5, fdata1,
                  weight = fgwt)
> fgfit1
Call:
coxph(formula = Surv(fgstart, fgstop, fgstatus) ~ ccr5, data = fdata1,
      weights = fgwt)

              coef exp(coef) se(coef)      z      p
ccr5WM -1.004      0.366    0.306 -3.28 0.001

Likelihood ratio test=13.9  on 1 df, p=0.000189
n= 2668, number of events= 113
(21 observations deleted due to missingness)
> fdata2 <- finegray(Surv(time, event) ~., adata, etype="SI")
> fgfit2 <- coxph(Surv(fgstart, fgstop, fgstatus) ~ ccr5, fdata2,
                  weight = fgwt)
> fgfit2
Call:
coxph(formula = Surv(fgstart, fgstop, fgstatus) ~ ccr5, data = fdata2,
      weights = fgwt)

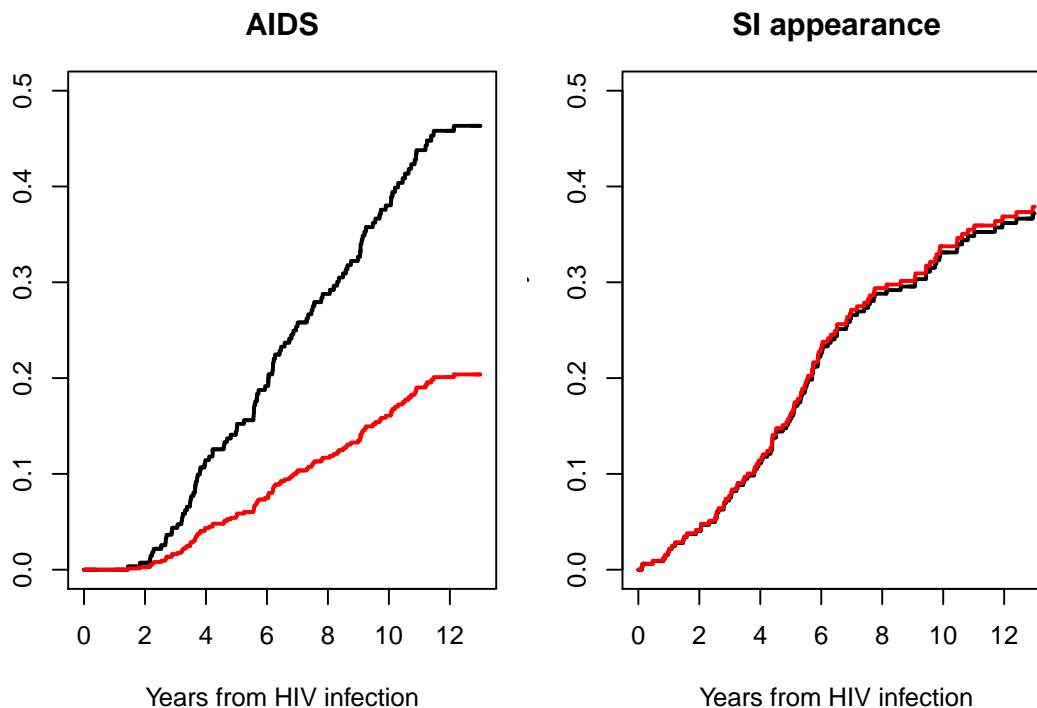
              coef exp(coef) se(coef)      z      p
ccr5WM 0.0237      1.0239    0.2355 0.1 0.92

Likelihood ratio test=0.01  on 1 df, p=0.92
n= 2195, number of events= 107
(31 observations deleted due to missingness)
```

The predicted curves based on the Fine-Gray model use the ordinary survival tools (not Aalen-Johansen), since they are ordinary Cox models on a special data set.

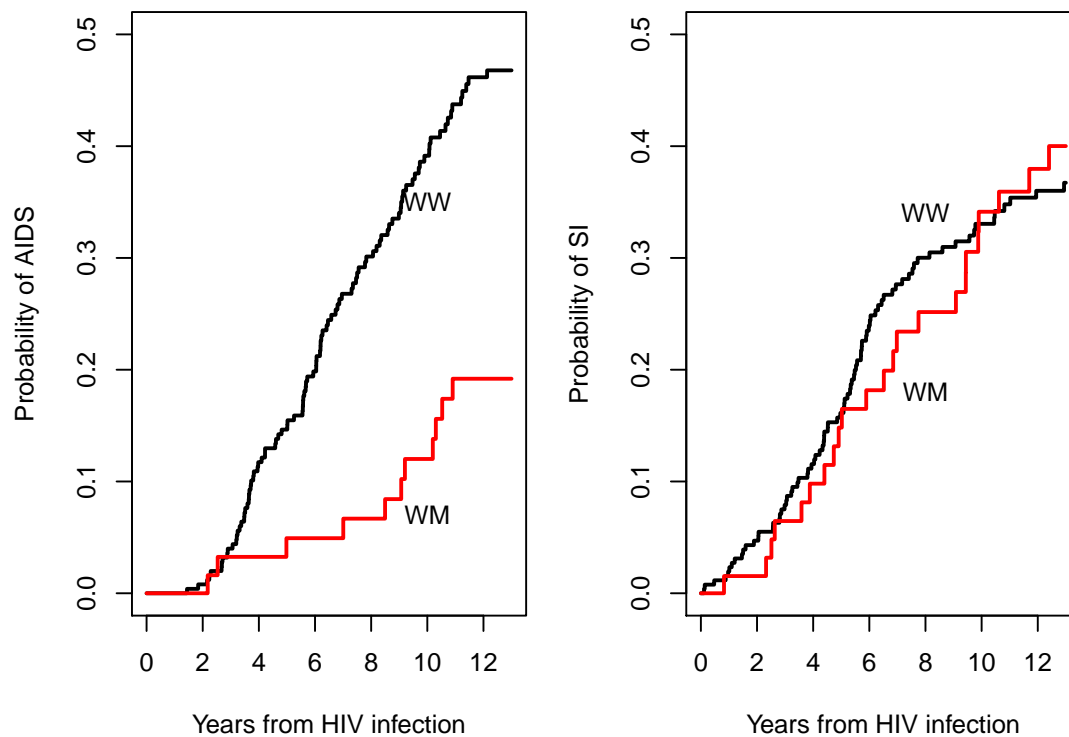
```
> # re-create figure T8: Fine-Gray curves
> fgsurv1<-survfit(fgfit1,newdata=tdata)
> fgsurv2<-survfit(fgfit2,newdata=tdata)
> oldpar <- par(mfrow=c(1,2), mar=c(4.1, 3.1, 3.1, 1)) #leave room for title
> plot(fgsurv1, col=1:2, lty=c(1,1,2,2), lwd=2, xmax=13,
      ylim=c(0, .5),fun='event',
      xlab="Years from HIV infection", ylab="Probability")
> title("AIDS")
> plot(fgsurv2, col=1:2, lty=c(1,1,2,2), lwd=2, xmax=13,
      ylim=c(0, .5), fun='event',
      xlab="Years from HIV infection", ylab="Probability")
```

```
> title("SI appearance")
> par(oldpar)
```



The last plot in this section of the tutorial contains the non-parametric fits.

```
> # re-create figure T9: curves by CCR type
> aj2 <- survfit(Surv(time, event) ~ ccr5, adata)
> oldpar <- par(mfrow=c(1,2))
> plot(aj2[,1], xmax=13, col=1:2, lwd=2, ylim=c(0, .5),
       xlab="Years from HIV infection", ylab="Probability of AIDS")
> text(c(10, 10), c(.35, .07), c("WW", "WM"))
> plot(aj2[,2], xmax=13, col=1:2, lwd=2, ylim=c(0, .5),
       xlab="Years from HIV infection", ylab="Probability of SI")
> text(c(8, 8), c(.34, .18), c("WW", "WM"))
> par(oldpar)
```



## 6.2 Multi-state models

The multi-state model is based on patients from the European Blood and Marrow Transplant registry. The initial state for each subject is bone marrow transplant after which they may have platelet recovery (PR); the end stage is relapse or death. Important covariates are the disease classification of AML, ALL or CML, age at transplant (3 groups), whether T-cell depletion was done, and whether donor and recipient are sex matched.

We first reprise table T2 to verify that we have the same data set.

```
> table(ebmt3$dissub)
AML ALL CML
853 447 904
> table(ebmt3$drmatch)
No gender mismatch    Gender mismatch
           1648              556
> table(ebmt3$tcd)
No TCD    TCD
 1928    276
> table(ebmt3$age)
<=20 20-40 >40
  419  1057  728
```

Next create the analysis data set `edata`. The `tmerge` function creates the basic time course data set that tracks a subject from state to state using (`tstart`, `tstop`) intervals of time. The `enum` variable is 1 for a subject who has had platelet recovery. We also shorten one of the factor labels so as to better fit the printout on a page. Printout of a subset of rows shows that subjects 8 and 11 achieve PR, subject 9 is censored at 3.5 years (1264/365), and subject 10 dies at about 1 year. We then generate the multi-state  $P(t)$  curves, a plot that does not appear in the tutorial. It shows the rapid onset of platelet recovery followed by a slow but steady conversion of these patients to relapse or death.

```
> temp <- subset(ebmt3, select = -c(prtime, prstat, rfstime, rfsstat))
> edata <- tmerge(temp, ebmt3, id,
                  rstat = event(rfstime, rfsstat),
                  pstat = event(prtime, prstat),
                  enum = tdc(prtime))
> print(edata[15:20,-(3:5)])
```

	id	dissub	tstart	tstop	rstat	pstat	enum
15	8	ALL	0	35	0	1	0
16	8	ALL	35	1448	0	0	1
17	9	AML	0	1264	0	0	0
18	10	CML	0	338	1	0	0
19	11	AML	0	50	0	1	0
20	11	AML	50	84	1	0	1

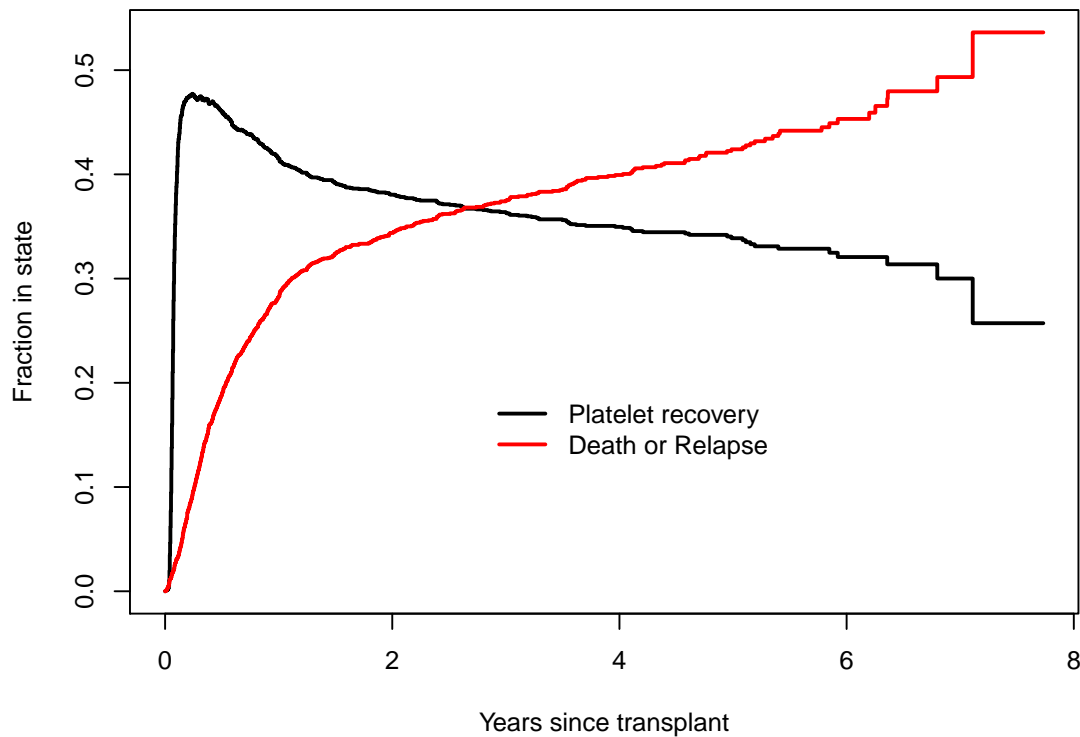
```
> # Check that no one has recovery and death on the same day
> with(edata, table(rstat, pstat))
```

		pstat	
		0	1
rstat	0	1363	1169
	1	841	0

```
> # Create the factor outcome
> edata$event <- with(edata, factor(pstat + 2*rstat, 0:2,
                                   labels = c("censor", "PR", "RelDeath")))
> levels(edata$drmatch) <- c("Match", "Mismatch")
> surv1 <- survfit(Surv(tstart, tstop, event) ~ 1, edata, id=id)
> surv1$transitions # matches the Frequencies on page C5
```

		to	
from		PR	RelDeath
PR	PR	0	383
	RelDeath	0	0
		1169	458

```
> plot(surv1, col=1:2, xscale=365.25, lwd=2,
       xlab="Years since transplant", ylab="Fraction in state")
> legend(1000, .2, c("Platelet recovery", "Death or Relapse"),
       lty=1, col=1:2, lwd=2, bty='n')
```



Many of the fits in the paper make use of a stacked data set. Subjects can go from baseline to platelet recovery (`temp1`), baseline to death/relapse (`temp2`) or from PR to the death/relapse state (`temp3`).

```
> temp1 <- with(edata, data.frame(edata[enum==0,], status=pstat[enum==0],
  trans="1->2", from=1, to=2)) # baseline to PR
> temp2 <- with(edata, data.frame(edata[enum==0,], status=rstat[enum==0],
  trans="1->3", from=1, to=3)) # baseline to relapse/death
> temp3 <- with(edata, data.frame(edata[enum==1,], status=rstat[enum==1],
  trans="2->3", from=2, to=3)) # PR to relapse/death
> edata2 <- rbind(temp1, temp2, temp3) # the stacked data set
```

The first fit, found in the left columns of table T3, has separate strata and covariate effects for each of the three transitions. We can fit the model in multiple ways: first as 3 separate fits (`efit1.1`, `efit1.2`, `efit1.3`), then using the stacked data set and a strata by covariate interaction (`efit1b`) and then rewriting the model to get a slightly different variable coding (`efit1c`). In the first two fits below everyone has `tstart=0` and so `tstart` can be left out of the model statement. (Or left in: it has no effect.)

```
> efit1.2 <- coxph(Surv(tstop, event=='PR') ~
  dissub + age + drmatch + tcd, ties='breslow',
  data=edata, subset = (enum==0))
> efit1.3 <- coxph(Surv(tstop, event=='RelDeath') ~
  dissub + age + drmatch + tcd, ties='breslow',
```

```

data=edata, subset = (enum==0))
> efit2.3 <- coxph(Surv(tstart, tstop, event=='RelDeath') ~
  dissub + age + drmatch + tcd, ties='breslow',
  data=edata, subset = (enum==1))
> round(cbind('1->2'= coef(efit1.2) , '1->3'= coef(efit1.3),
  '2->3' =coef(efit2.3)), 3)

```

	1->2	1->3	2->3
dissubALL	-0.044	0.256	0.136
dissubCML	-0.297	0.017	0.247
age20-40	-0.165	0.255	0.062
age>40	-0.090	0.526	0.581
drmatchMismatch	0.046	-0.075	0.173
tcdTCD	0.429	0.297	0.201

These coefficients match those in the left column of table T3, which lists all six coefficients for the 1.2 transition, then the six for the 1.3 transition, and finally the six for the 2.3 transition. This is a rational way to present the results to a reader.

The other fits result in a different order.

```

> efit1b <- coxph(Surv(tstart, tstop, status) ~ (dissub + age + drmatch + tcd)
  *strata(trans), data=edata2, ties="breslow")
> efit1c <- coxph(Surv(tstart, tstop, status) ~
  strata(trans)/(dissub + age + drmatch + tcd),
  data=edata2, ties="breslow")
> # Rearrange the order so as to match the paper
> mycoef <- summary(efit1c)$coefficients[, c(1,3)] # coef and se
> index <- as.vector(matrix(1:18, ncol=3, byrow=T))
> round(mycoef[index,], 3)

```

	coef	se(coef)
strata(trans)1->2:dissubALL	-0.044	0.078
strata(trans)1->2:dissubCML	-0.297	0.068
strata(trans)1->2:age20-40	-0.165	0.079
strata(trans)1->2:age>40	-0.090	0.086
strata(trans)1->2:drmatchMismatch	0.046	0.067
strata(trans)1->2:tcdTCD	0.429	0.080
strata(trans)1->3:dissubALL	0.256	0.135
strata(trans)1->3:dissubCML	0.017	0.108
strata(trans)1->3:age20-40	0.255	0.151
strata(trans)1->3:age>40	0.526	0.158
strata(trans)1->3:drmatchMismatch	-0.075	0.110
strata(trans)1->3:tcdTCD	0.297	0.150
strata(trans)2->3:dissubALL	0.136	0.148
strata(trans)2->3:dissubCML	0.247	0.117
strata(trans)2->3:age20-40	0.062	0.153
strata(trans)2->3:age>40	0.581	0.160
strata(trans)2->3:drmatchMismatch	0.173	0.115
strata(trans)2->3:tcdTCD	0.201	0.126



```

> # Compare the log-likelihoods of the fits
> matrix(c(efit1.2$loglik + efit1.3$loglik + efit2.3$loglik,
           efit1b$loglik, efit1c$loglik), nrow=2,
          dimnames=list(c("Initial LL", "Final LL"),
                        c("fit a", "fit b", "fit c")))
           fit a      fit b      fit c
Initial LL -14168.02 -14168.02 -14168.02
Final LL   -14109.19 -14109.19 -14109.19

```

The coefficients of `efit1b` are focused on contrasts: first the six coefficients for the 1.2 transition and then the *changes* in those coefficients for the 1.3 and 2.3 transitions. In its favor, this makes it easy to see if there are significant differences in, e.g., the disease effect for transition 1.2 and 1.3. The last fit gives the same coefficients as the separate fits, but in a different order. We can see from the log-likelihoods that the three fits are equivalent.

Unlike '\*' and '+', the '/' operator is not symmetric. An equivalent form to `a/b` is `a %in% b`, which is supposed to help us remember which should be first. However, the natural interpretation (at least to us) of these as coefficients *within* strata is still reversed.

The next fit assumes that the baseline hazards for the two transitions to relapse-death are proportional, but retains separate coefficients for each transition. Again there are multiple ways to do this. Since the model leaves the coefficients and strata for the transplant to PR transition unchanged the first approach is to replace `efit1.3` and `efit2.3` with a single model `efit2a`, the second approach is to again fit all three at once.

```

> efit2a <- coxph(Surv(tstart, tstop, status) ~
                  factor(from)*(dissub + age + drmatch + tcd),
                  data=edata2, ties="breslow", subset= (to==3))
> efit2b <- coxph(Surv(tstart, tstop, status) ~
                  factor(from)/(dissub + age + drmatch + tcd),
                  data=edata2, ties="breslow", subset= (to==3))
> mycoef <- summary(efit2a)$coefficients[, c(1,3)]
> round(mycoef, 3)[c(2,4,6,8,10,12, 3,5,7,9,11,13,1),]

```

	coef	se(coef)
dissubALL	0.261	0.135
age20-40	0.251	0.151
drmatchMismatch	-0.072	0.110
factor(from)2:dissubALL	-0.121	0.200
factor(from)2:age20-40	-0.195	0.215
factor(from)2:drmatchMismatch	0.241	0.159
dissubCML	0.004	0.108
age>40	0.526	0.158
tcdTCD	0.319	0.150
factor(from)2:dissubCML	0.247	0.159
factor(from)2:age>40	0.037	0.225
factor(from)2:tcdTCD	-0.108	0.196
factor(from)2	-0.379	0.212

```

> efit2c <- coxph(Surv(tstart, tstop, status) ~ strata(to) +
  trans/(dissub + age + drmatch + tcd),
  data=edata2, ties="breslow")
> round(summary(efit2b)$coefficients[, c(1,3)], 3)

```

	coef	se(coef)
factor(from)2	-0.379	0.212
factor(from)1:dissubALL	0.261	0.135
factor(from)2:dissubALL	0.140	0.148
factor(from)1:dissubCML	0.004	0.108
factor(from)2:dissubCML	0.250	0.117
factor(from)1:age20-40	0.251	0.151
factor(from)2:age20-40	0.056	0.153
factor(from)1:age>40	0.526	0.158
factor(from)2:age>40	0.562	0.160
factor(from)1:drmatchMismatch	-0.072	0.110
factor(from)2:drmatchMismatch	0.169	0.114
factor(from)1:tcdTCD	0.319	0.150
factor(from)2:tcdTCD	0.211	0.126

This matches the second column of table T3, though again the model coefficients are not in the same order as the table. (In table T3 the first 6 coefficients in columns 2 and 3 are left blank since they are unchanged from column 1; this confused me at first.) The NA in position 2 of `efit2c` is due to the fact that the model as written has both a separate stratum for the 1.2 transition and a coefficient for it. These are confounded and as a result `coxph` fits one more coefficient than it needs, 2 instead of 1. In such a situation the last coefficient is the one set to NA.

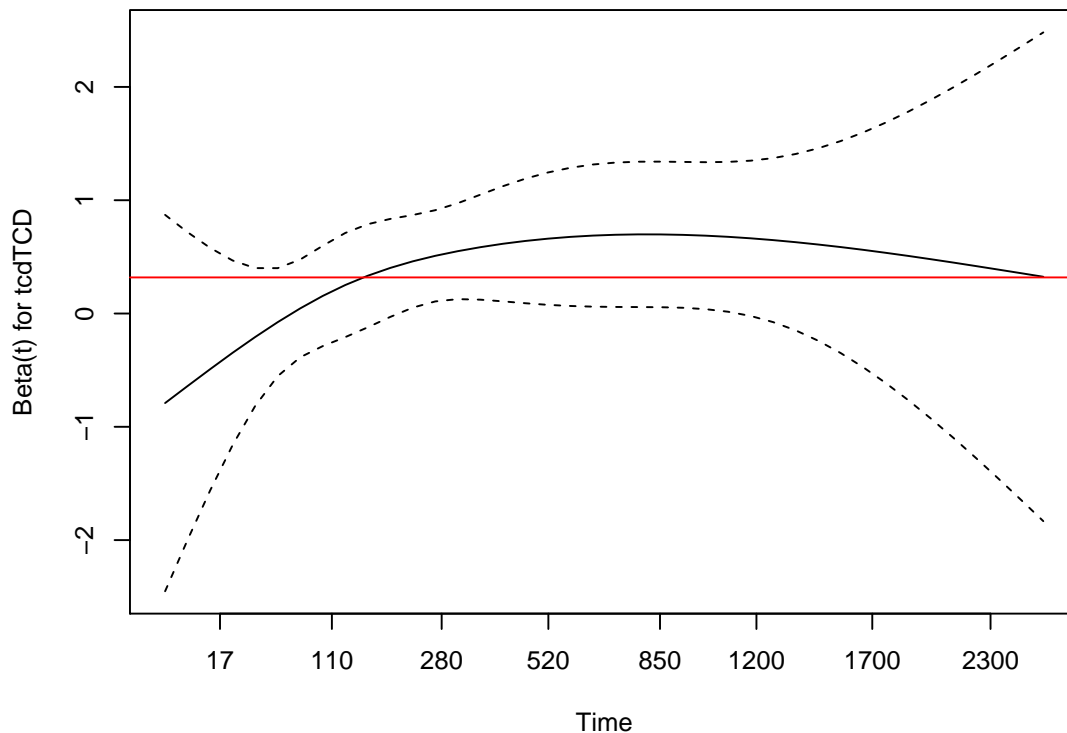
Rather than using the plot of figure T14 along with a “they look parallel” argument, we can more directly assess the proportional hazards assumption in model `efit2a` by using the `cox.zph` routine. We chose to plot on the  $\sqrt{t}$  scale to emphasize the first year, where more events occur. There is little formal evidence for non-proportional hazards, with a p-value of 0.27, though the confidence limits are very wide.

```

> z2 <- cox.zph(efit2a, transform=function(x) sqrt(x))
> plot(z2[7], resid=FALSE)
> abline(h=coef(efit2a)[7], col=2)
> z2[7]

```

	rho	chisq	p
tcdTCD	0.0553	2.61	0.106



The third column of table S3 adds the time of platelet recovery as a covariate, i.e., we simply add `tstart` as a variable in the model. We had not changed the data set from days to years, as the tutorial did; to match their coefficient this extra variable needs to be in years.

```
> # The extra variable for column 3 of table 3
> efit3 <- coxph(Surv(tstart, tstop, status) ~ I(tstart/365.25) +
               factor(from)/(dissub + age + drmatch + tcd),
               data=edata2, ties="breslow", subset= (to==3))
> coef(efit3)["I(tstart/365.25)"]
I(tstart/365.25)
0.2952259
```

The Cox model fit for `efit3` produces a warning message that a time variable appears on both sides of the equation. This is a warning that should not be ignored without careful thought, since it is often associated with an invalid model. In short, a Cox model may not include a time-dependent covariate whose value (at any given event time) is dependent on the future. In this case we are okay. Because all the observations are on a common time scale, the value of `tstart` does not become “known” for a subject until all of the subjects have passed that time point.

Table T4 reruns all the models using a “clock reset” time scale. These are the same models as before but with `tstop-tstart` as the time value. The code below gives the first column of the table.

```

> efit4 <- coxph(Surv(tstop - tstart, status) ~
  strata(trans)/(dissub + age + drmatch + tcd),
  data=edata2, ties="breslow")
> mycoef4 <- summary(efit4)$coefficients[, c(1,3)] # coef and se
> index4 <- index[7:18] # leave off transition 1->2
> round(mycoef4[index4,], 3)

```

	coef	se(coef)
strata(trans)1->3:dissubALL	0.256	0.135
strata(trans)1->3:dissubCML	0.017	0.108
strata(trans)1->3:age20-40	0.255	0.151
strata(trans)1->3:age>40	0.526	0.158
strata(trans)1->3:drmatchMismatch	-0.075	0.110
strata(trans)1->3:tcdTCD	0.297	0.150
strata(trans)2->3:dissubALL	0.120	0.148
strata(trans)2->3:dissubCML	0.252	0.117
strata(trans)2->3:age20-40	0.065	0.153
strata(trans)2->3:age>40	0.582	0.160
strata(trans)2->3:drmatchMismatch	0.170	0.115
strata(trans)2->3:tcdTCD	0.197	0.126

We now predict the future state of a patient, using as our reference set two subjects who are  $\leq 20$  years old, gender matched, AML, with and without T-cell depletion. We will use the fit from column 2 of table T3, which has proportional hazards for the two transitions to Relapse/Death and a separate baseline hazard for the PR transition. We can predict the hazard for the 1  $\rightarrow$  2 transition from model `efit1.2`, the hazards for the other two transitions come from `efit2a`.

```

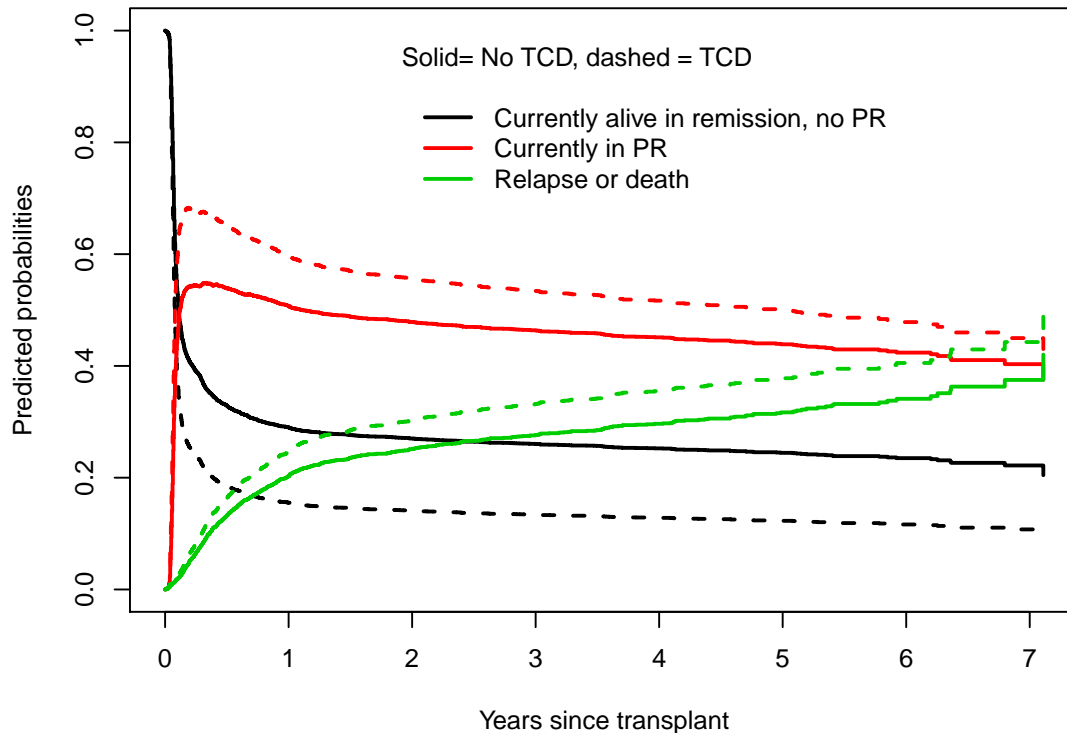
> newdata1 <- expand.grid(age="<=20", dissub="AML", drmatch="Mismatch",
  tcd=c("No TCD", "TCD"))
> newdata2 <- cbind(newdata1[c(1,2,1,2),], from=c(1,1,2,2))
> newdata2
  age dissub drmatch tcd from
1  <=20    AML Mismatch No TCD  1
2  <=20    AML Mismatch  TCD  1
1.1 <=20    AML Mismatch No TCD  2
2.1 <=20    AML Mismatch  TCD  2
> tcurve1 <- survfit(efit1.2, newdata1, se.fit=FALSE)
> tcurve2 <- survfit(efit2a, newdata2, se.fit=FALSE)
> tmat <- matrix(list(), 3, 3,
  dimnames=list(from=c("Tx", "PR", "R/D"),
    to =c("Tx", "PR", "R/D")))
> tmat[1,2] <- list(tcurve1)
> tmat[1,3] <- list(tcurve2[1:2])
> tmat[2,3] <- list(tcurve2[3:4])
> ecurve <- survfit(tmat, p0=c(1,0,0))
> plot(ecurve, col=c(1,1,2,2,3,3), lty=1:2, lwd=2, xscale=365.25,

```

```

      xlab="Years since transplant", ylab="Predicted probabilities")
> legend(700, .9, c("Currently alive in remission, no PR", "Currently in PR",
      "Relapse or death"), col=1:3, lwd=2, bty='n')
> text(700, .95, "Solid= No TCD, dashed = TCD", adj=0)

```



The predicted effect of TCD is to increase the occupancy of both the PR and remission/death states, at the expense of the unchanged state.

Figure T15 separates the remission/death state into two portions, those who had prior PR and those who did not. To create this set of curves we re-package the above calculation as a four state model which has two relapse/death endpoints. In order to make it easy to do the stacked plot that they show, the states matrix is put in a particular order. Note that `p0` changes, since everyone now starts in state number 4.

```

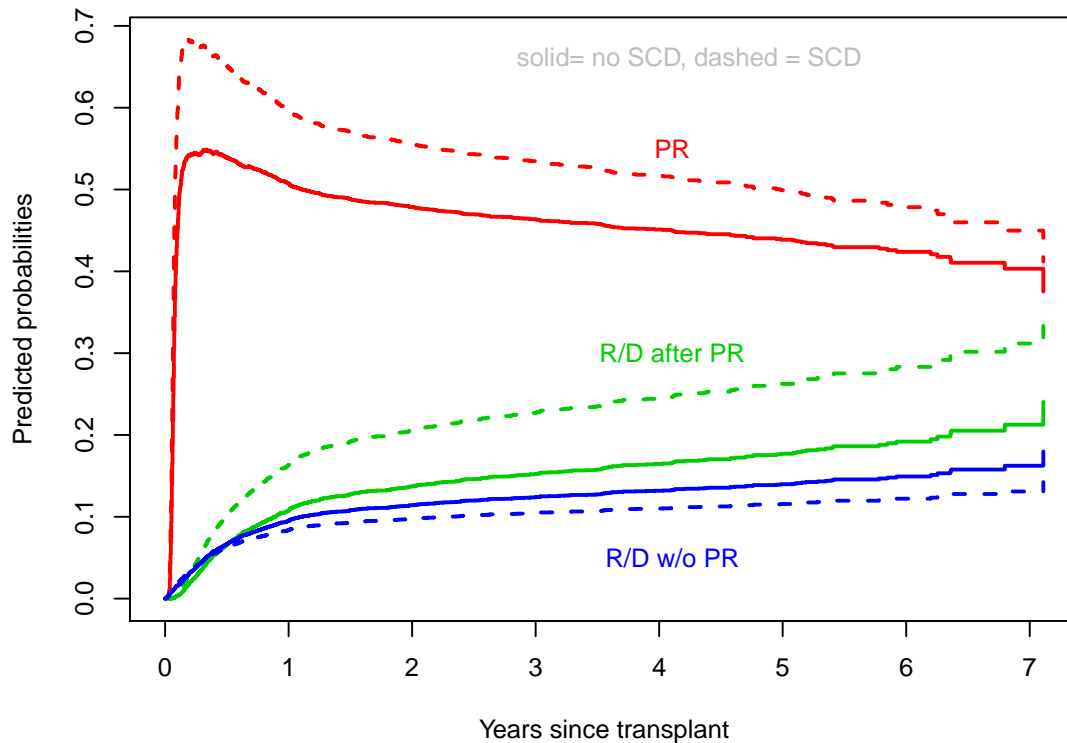
> dtemp <- c("PR", "R/D after PR", "R/D without PR", "Tx")
> xmat <- matrix(list(), 4, 4, dimnames=list(from=dtemp, to=dtemp))
> xmat[4,1] <- list(tcurve1)
> xmat[4,3] <- list(tcurve2[1:2] )
> xmat[1,2] <- list(tcurve2[3:4] )
> ecurve2 <- survfit(xmat, p0=c(0,0,0,1))
> dim(ecurve2)  # rows are the two "subjects", cols are the states
[1] 2 4
> plot(ecurve2[, 1:3], lwd=2, lty=1:2, col=c(2,2,3,3,4,4,1,1),
      xscale=365.25,

```

```

      xlab="Years since transplant", ylab="Predicted probabilities")
> text(c(1500, 1500, 1500), c(.55, .3, .05),
      c("PR", "R/D after PR", "R/D w/o PR"), col=2:4)
> text(1550, .66, "solid= no SCD, dashed = SCD", col='gray')

```



We have left the transplant state off this second plot to reduce the total number of lines. (The sum of state probabilities is 1, so one state can always be left off without loss.) The suggestion from this plot is that TCD increases the total number of PR, but at a cost of the durability of the PR; essentially all of the “extra” failures under TCD have passed through the PR state.

Last, we redraw this figure as a stacked diagram. The authors are not themselves enamored of the stacked form, at least partly because the impression it gives can depend on the order of the states. We split it as two figures because the version with both TCD and no TCD together had too many crossing lines. Figure T15 corresponds to the left panel.

```

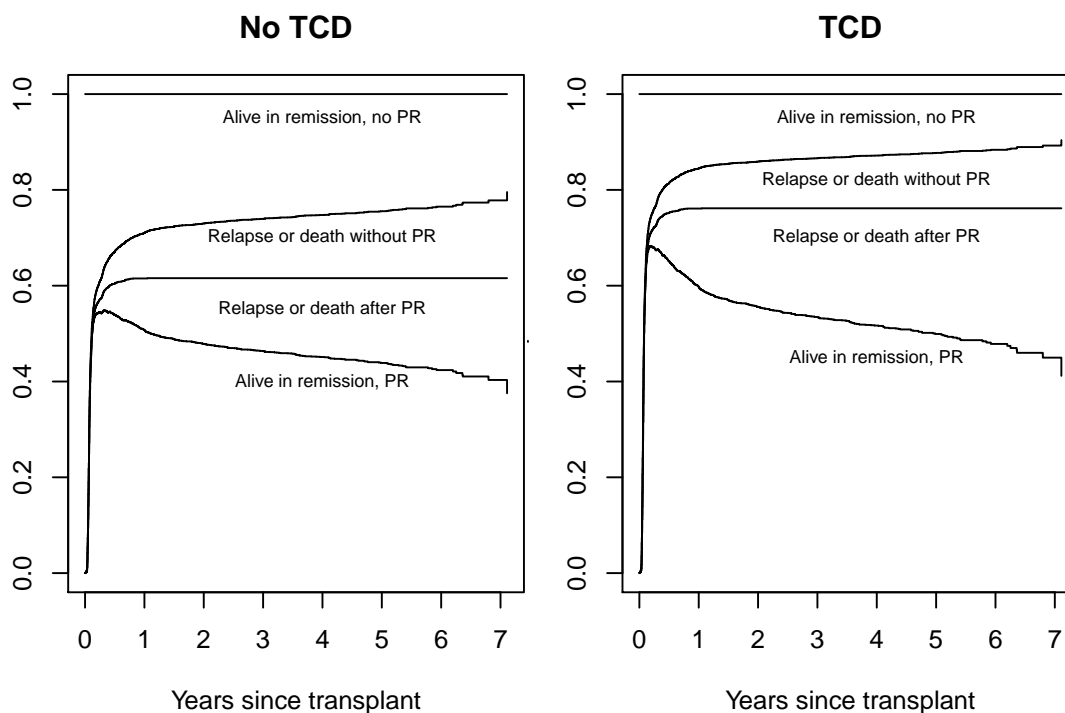
> tcurve <- ecurve2
> tcurve$prev <- t(apply(tcurve$prev, 1, cumsum))
> oldpar <- par(mfrow=c(1,2), mar=c(4.1, 3.1, 3.1, .1))
> plot(tcurve[1,1:4], col=1, xscale=365.25, ylim=c(0,1),
      xlab="Years since transplant", ylab="Predicted probabilities")
> text(rep(4*365, 4), c(.4, .55, .7, .95), cex=.7,
      c("Alive in remission, PR", "Relapse or death after PR",
        "Relapse or death without PR", "Alive in remission, no PR"))
> title("No TCD")

```

```

> plot(tcurve[2,1:4], col=1, xscale=365.25, ylim=c(0,1),
      xlab="Years since transplant", ylab="Predicted probabilities")
> text(rep(4*365, 4), c(.45, .7, .82, .95), cex=.7,
      c("Alive in remission, PR", "Relapse or death after PR",
        "Relapse or death without PR", "Alive in remission, no PR"))
> title("TCD")
> par(oldpar)

```



A last task is to draw the curves using only the overall fits, e.g., `efit2b`. This will be necessary for models with shared coefficients across some strata, for instance, since they can only be fit using the stacked data form. At the moment the `survfit` routine gets confused by models that contain nesting (`%in%` or `/`) operators). However, since the predicted values are independent of exactly how factors are coded in the model this is only a nuisance: rerun the fit using dummy variables or `*` operators and obtain the curves from the latter.

## 7 Other software

### 7.1 The `mstate` package

As the number of states + transitions (arrows + boxes) gets larger then the ‘by hand’ approach used above for creating a stacked data set, labeling coefficients, and producing multi-state curves becomes a challenge. (It is still fairly easy to do, just not as easy to ensure it has been done

correctly.) The `mstate` package starts with a definition of the matrix of possible transitions and uses that to drive tools that build and analyze the stacked data set in a more automated fashion. We recommend it for more complex models. (The tutorial above is about at our personal threshold.) A second advantage of `mstate` is that all the Cox model fits are now in one well indexed object, which allows for calculation of proper confidence intervals for the state probabilities  $p(t)$ . (Since all of the steps used the same transition matrix template, the necessary computations are scripted and reliable.)

## 7.2 The `msm` package

There are two broad classes of multi-state data:

- Panel data arises when subjects have regular visits, with the current state assessed at each visit. We don't know when the transitions between states occur, or if other states may have been visited in the interim — only the subject's state at specific times.
- Survival data arises when we observe the transition times; death, for example.

The overall model (boxes and arrows), the quantities of interest (transition rates and  $p(t)$ ), and the desired printout and graphs are identical for the two cases. Much of the work in creating a data set is also nearly the same. The underlying likelihood equations and resulting analytical methods for solving the problem are, however, completely different. The `msm` package addresses panel data, while `survival`, `mstate`, and a host of others are devoted to survival data.

## 8 Conclusions

When working with acute diseases, such as advanced cancer or end-stage liver disease, there is often a single dominating endpoint. Ordinary single event Kaplan-Meier curves and Cox models are then efficient and sufficient tools for much of the analysis. Such data was the primary use case for survival analysis earlier in the authors' careers. Data with multiple important endpoints is now common, and multi-state methods are an important addition to the statistical toolbox. As shown above, they are now readily available and easy to use.

It is sometimes assumed that the presence of competing risks *requires* the use of a Fine-Gray model (we have seen it in referee reports), but this is not correct. The model may often be useful, but is one available option among many. Grasping the big picture for a multi-state data set is always a challenge and we should make use of as many tools as possible. We are often reminded of the story of a centenarian on his 100th birthday proclaiming that he was looking forward to many more years ahead because “I read the obituaries every day, and you almost never see someone over 100 listed there”. It is not always easy to jump between observed deaths, hazard rates, and lifetime risk.

## References

- [1] H. Putter, M. Fiocco, and R. B. Geskus. Tutorial in biostatistics: Competing risks and multi-state models. *Stat. in Medicine*, 26:2389–2430, 2007.