

A Practitioners' User Guide for 'Multi-State Models for Event History Analysis' with Applications in R*

Gradon Nicholls *PhD Student, Statistics, University of Waterloo*

Event history analysis is the study of discrete events which occur over time. Standard survival analysis, for example, studies a single type of event: death (or more generically, 'failure'). In practice, a Statistician may encounter data involving more complex sequences of events. For instance, competing risks models involve 2 or more possible causes of failure. Illness-death models involve an intermediary event ('illness') which may occur before failure. In recurrent events analysis, the same type of event may occur multiple times. These and other models can all be described within the framework of multi-state models. The objective of this report is to provide a practitioners' user guide to multi-state modeling based on [Andersen and Keiding \(2002\)](#). This user guide aims first to reiterate the theory underlying multi-state modeling, providing more details and context for the non-expert. Secondly, this user guide supplements their work by illustrating how to estimate such models using data and packages in R which are freely available.

1 Introduction

What is **event history analysis**? "History" implies that we follow individuals over time. "Event" implies that we are interested in discrete events that happen to these individuals. More precisely, we will think of an event as a point in time where an individual **transitions** from one **state** to another. For instance, death is an event where one transitions from the state "healthy" to the state "dead", while illness is an event where one transitions from "healthy" to "sick". We see then that event history analysis is the study of individuals, followed over continuous time, who may transition between a finite (or countably infinite) number of states over that time.

It turns out that many real-world problems can be modelled in this same general framework—called **multi-state models**. [Andersen and Keiding \(2002\)](#) provide a great overview of these methods, and provide a thorough literature review. However, while their overview is good for experts in the field, it is difficult to follow for someone new to the topic. Namely, the authors gloss over steps in their treatment of the theory, they only provide one data-driven example of one type of multi-state process, and they do not provide code or algorithms for estimating the models.

The objective of this report is thus to convert the content of [Andersen and Keiding \(2002\)](#) into a "practitioners' user guide". First, I reiterate the theory of multi-state processes, providing more mathematical details or conceptual context in certain areas. Next, I go through a series of practical examples using data and packages in R. The examples include survival analysis, competing risks, illness-death, and repeated events models. For each example, I describe the dataset, visually represent the process in terms of its state diagram, specify the model, and estimate the model using R. Due to time constraints, I focus only on estimating each model and interpreting the coefficients. A theoretical revision to this report would include data visualization and analysis of model fit.

*This report was prepared for STAT935 (Survival Analysis) taught by Professor Richard Cook in the Winter 2023 semester. Theoretical content here is based on [Andersen and Keiding \(2002\)](#) along with my own interpolations. Any and all mistakes are my fault only.

2 Multi-state Modeling: Theory

2.1 The Setup

A **multi-state process** is a stochastic process $(X(t), t \in \mathcal{T})$. $X(t)$ represents the current **state** of the process at time t . The process can take on one of a finite (or countably infinite) number of states $h = 1, \dots, p$. We call $\mathcal{S} = \{1, \dots, p\}$ the **state space**. \mathcal{T} represents the set of all time over which the process is observed. Generally, $\mathcal{T} = [0, \tau]$ or $\mathcal{T} = [0, \tau)$ where 0 is the **origin** and τ is the last moment of time at which we observe the process.

The process starts in some initial state $X(0)$. Oftentimes, we will assume all subjects start in the same initial state. For instance, in survival analysis we will assume that all subjects are initially alive (or have initially not failed). More generally, we may specify an **initial distribution** $\pi_h(0) = \mathbf{P}(X(0) = h); h \in \mathcal{S}$.

As time proceeds, the state is governed by **transition probabilities** $P_{h \rightarrow j}(s, t); h, j \in \mathcal{S}$ —that is, the probability that the process enters state j at time t , given that the process was in state h as of time s . Let \mathcal{X}_s represent the **history** of the process prior to time s (i.e. over the interval $[0, s)$). Then transition probabilities are given by:

$$P_{h \rightarrow j}(s, t) = \mathbf{P}(X(t) = j \mid X(s) = h, \mathcal{X}_s); \quad t > s.$$

Notice that $P_{h \rightarrow h}(s, t)$ is a valid probability, and represents the probability that the process *remains* in state h between times s and t .

As we will see, multi-state models are most often specified in terms of **transition intensities** given by the time derivative of the transition probabilities:

$$\alpha_{hj}(t) = \lim_{\Delta t \rightarrow 0} \frac{P_{h \rightarrow j}(t, t + \Delta t) - P_{h \rightarrow j}(t, t)}{\Delta t}.$$

Note that these are often called “hazard functions”, particularly when state j represents death/failure.

We call $h \in \mathcal{S}$ an **absorbing state** if $\alpha_{hj}(t) = 0 \quad \forall j, t$. That is, h is an absorbing state if you will never leave state h , even after infinite time. (We could alternatively define an absorbing state as one where $P_{h \rightarrow h}(s, t) = 1 \quad \forall t > s$). If h is not an absorbing state, then we say it is a **transient state**.

2.2 The Likelihood

Now imagine we collect data on n individuals $i = 1, \dots, n$. In particular, for each individual, we observe “snapshots” of their multi-state process $X_i(t)$ over the time interval $[0, \tau_i]$. We can think of τ_i either as some fixed time at which we stop the study, or as a right-censoring time.

We study the multi-state process as a **counting process**. Specifically, we can count the number of times we observe $X_i(t)$ transition from state h to state j (denoted $h \rightarrow j$) up to some time t . Let's call this count $N_{h \rightarrow j}^i(t)$. Further, let $T_{h \rightarrow j}^{ik}$ denote the k^{th} time that person i experiences an $h \rightarrow j$ event. As individuals are observed only up to time τ_i , we can denote the total number of events observed to be $N_{h \rightarrow j}^i(\tau_i)$, that is, $k = 1, \dots, N_{h \rightarrow j}^i(\tau_i)$.

Constructing likelihoods is generally a matter of taking a product of density functions. However, because multi-state models are generally specified in terms of the transition intensity, we'd like to re-write the likelihood in terms of α . To do so, we take advantage of two results which may be familiar from survival analysis. First, $\alpha(t) = \frac{f(t)}{S(t)}$ —i.e. the intensity/hazard function is equal to the density function divided by the survival function. Second, $S(t) = \mathbf{P}(T > t) = \exp\left(-\int_0^t \alpha(u) du\right)$ —i.e. the survival function exponentiates the “cumulative hazard/intensity”. Thus, $f(t) = \alpha(t) \exp\left(-\int_0^t \alpha(u) du\right)$.

We (heuristically) make two modifications:

- Let α_{hj}^i be individual-specific intensities for each type of $h \rightarrow j$ transition.
- $h \rightarrow j$ transitions are only relevant during periods of time where individual i is currently in state h . Thus, let $Y_h^i(t) = I(X_i(t) = h)$ denote instances of time where i is observed to be in state h . (Notice then that $Y_h^i(t) = 0 \quad \forall t > \tau_i$, as i is not observed beyond this point.)

We thus have density $f_{hj}^i(t) = \alpha_{hj}^i(t) \exp\left(-\int_0^\infty \alpha_{hj}^i(u) Y_h^i(u) du\right)$. To obtain the overall likelihood we take the product over all individuals i , all types of transitions $h \rightarrow j$, and all $N_{h \rightarrow j}^i(\tau_i)$ times we observe a transition of that type, evaluating each density at time $T_{h \rightarrow j}^{ik}$ when the event is observed:

$$L \propto \prod_{i=1}^n \prod_{\substack{h \rightarrow j \\ h \neq j}} \prod_{k=1}^{N_{h \rightarrow j}^i(\tau_i)} \alpha_{hj}^i(T_{h \rightarrow j}^{ik}) \exp\left(-\int_0^\infty \alpha_{hj}^i(u) Y_h^i(u) du\right)$$

By using ‘ \propto ’ rather than ‘ $=$ ’ above, we are acknowledging some implicit assumptions. Namely, we are conditioning on the initial state of the process $X_i(0)$, any potential covariates appearing in $\alpha_{hj}^i(T_{h \rightarrow j}^{ik})$, and any potential (right) censoring. However, we will use ‘ $=$ ’ from here on out.

2.3 The Model

We obtain different models by specifying different functional forms for $\alpha_{hj}^i(t)$. Regression models for multi-state models most often take the ‘multiplicative’ form:

$$\alpha_{hj}^i(t) = \alpha_{hj0}(t)e^{\beta_{hj}^T Z_i}$$

where $\alpha_{hj0}(t)$ is the **baseline intensity** for transition $h \rightarrow j$, β_{hj} is a vector of transition-specific parameters, and Z_i is a vector of individual-specific (and time-independent) covariates. From here, models differ in how they specify $\alpha_{hj0}(t)$. Two common approaches are:

- **piecewise constant:** $\alpha_{hj0}(t) = \bar{\alpha}_{hj0}^{(l)}$ for $t \in (b_{hj}^{(l-1)}, b_{hj}^{(l)}]$ —that is, the baseline intensity takes on the constant value $\bar{\alpha}_{hj0}^{(l)}$ within the l^{th} time interval
- **cox proportional hazard:** make no assumptions about $\alpha_{hj0}(t)$ —this is called **semi-parametric** as it contains the non-parametric part, $\alpha_{hj0}(t)$, and the parametric part, $e^{\beta_{hj}^T Z_i}$

In fact, we do not expect severe differences in these two approaches. While the cox model is always maximally flexible, the piecewise constant model can be made equally flexible by increasing the number of intervals. On the other hand, the piecewise constant model can be more “parsimonious” (and with a smoother survival function) by specifying fewer intervals. In the examples in this report, I will always use the cox proportional hazard model.

2.4 Separable Likelihoods

As long as $\alpha_{hj0}(t), \beta_{hj}$ differ for each $h \rightarrow j$ transition, the likelihood is **separable**. To see this, first let $\Theta(\alpha_{hj0}(t), \beta_{hj}, Z_i) \equiv \alpha_{hj}^i(t) \exp\left(-\int_0^\infty \alpha_{hj}^i(u) Y_h^i(u) du\right)$. Then,

$$\begin{aligned} \log L &= \sum_{i=1}^n \sum_{\substack{h \rightarrow j \\ h \neq j}} \sum_{k=1}^{N_{h \rightarrow j}^i(\tau_i)} \log \Theta(\alpha_{hj0}(T_{h \rightarrow j}^{ik}), \beta_{hj}, Z_i) \\ &= \sum_{\substack{h \rightarrow j \\ h \neq j}} \sum_{i=1}^n \sum_{k=1}^{N_{h \rightarrow j}^i(\tau_i)} \log \Theta(\alpha_{hj0}(T_{h \rightarrow j}^{ik}), \beta_{hj}, Z_i) \\ \implies \frac{\partial \log L}{\partial \beta_{hj}} &= \sum_{i=1}^n \sum_{k=1}^{N_{h \rightarrow j}^i(\tau_i)} \frac{\partial \log \Theta(\alpha_{hj0}(T_{h \rightarrow j}^{ik}), \beta_{hj}, Z_i)}{\partial \beta_{hj}}. \end{aligned}$$

We see then that the maximum likelihood estimator for β_{hj} for a particular $h \rightarrow j$ event does not depend on the other events. The implication is that we can separately estimate models for each type of transition, allowing us to focus on specific events of interest while excluding others.

There may be instances where we desire to reduce the number of parameters we need to estimate. For instance, perhaps prior research shows that $\beta_{hj} = \beta \forall h, j$ is a reasonable assumption. In cases such as this, separability is no longer possible. We will see such an example in Section 3.

2.5 Markov Processes and Beyond

Recall that we defined transition probabilities as

$$P_{h \rightarrow j}(s, t) = \mathbf{P}(X(t) = j \mid X(s) = h, \mathcal{X}_s); \quad t > s,$$

where \mathcal{X}_s was the entire history of the process up to time s . In many cases, however, we will make the simplifying assumption that:

$$P_{h \rightarrow j}(s, t) = \mathbf{P}(X(t) = j \mid X(s) = h); \quad t > s,$$

that is, the only relevant information about the history of the process is the state of the process as of time s —anything that occurred prior to s is not informative. These are called **markov processes**.

In some cases we get the markov assumption for free—for instance, in survival analysis, where there are just two states—alive or dead—the fact that you were alive at time s immediately implies you were alive at all times previous to s , and so the entire history \mathcal{X}_s is not needed. The same is true for competing risks models.

There are two important cases where we will relax the markov assumption. First, in illness-death models, the probability of death may depend on whether you got sick first, as well as *how long* the illness lasted. Second, in recurrent events analysis, an individual may enter the same state multiple times. It is possible, then, that transition intensities depend on the number of previous recurrences and/or the amount of time spent in each recurrence.

We will look at examples of both these models in the next section, and we will see that we can use the same modelling and estimation approach as in markov processes, but with the addition of **time-dependent variables** into the transition intensities. While other violations of the markov assumption are certainly possible, we do not consider them here.

3 Multi-State Modeling: Examples

In this section we cover a series of examples of multi-state models. For each example, we introduce the data and show how it's structured, show how to represent the multi-state process diagrammatically, specify a model (or models), estimate the model in R, and interpret the results. As I ran out of time, I focus on estimating the models and interpreting coefficients, and do not plot survival curves, do robustness checks, etc.

We will make use of the following packages:

```
library(survival)
library(mstate)
```

3.1 Survival Analysis

3.1.1 Data

The `survival` package contains a number of real datasets which can be used to experiment with multi-state models. The first we consider is the `diabetic` dataset as sourced from [Huster, Brookmeyer and Self \(1989\)](#). In this study, 197 patients with diabetic retinopathy in both eyes and visual acuity of 20/100 or better in both eyes were chosen. Each patient had one eye randomized to the treatment (laser treatment) while the other eye received no treatment. "Failure" here was defined as the patient's visual acuity dropping below 5/200 for two consecutive appointments. The primary objective of the study was to test whether laser treatment had an effect on failure. Patients could be censored by death, dropout, or end of study.

Let's take a quick look at the data:

```
data(diabetic); head(diabetic)
```

```
##   id laser age  eye trt risk  time status
## 1  5 argon 28 left  0   9 46.23     0
## 2  5 argon 28 right 1   9 46.23     0
## 3 14 xenon 12 left  1   8 42.50     0
## 4 14 xenon 12 right 0   6 31.30     1
## 5 16 xenon  9 left  1  11 42.27     0
## 6 16 xenon  9 right 0  11 42.27     0
```

Notice first that there are two observations per patient—one for each eye. For each eye we have associated covariates (some omitted):

- `age`: age when diagnosed with diabetes
- `trt`: 1 if treatment, 0 otherwise
- `time`: time of last follow-up (if `status=0`) or time of event (if `status=1`)
- `status`: 0 if censored, 1 if event occurred

3.1.2 State Diagram

Figure 1 summarizes the multi-state process $X_i(t)$. Our state space is $S = \{0, 1\}$ where 0 represents the baseline state while 1 represents “failure”—in this case loss of vision. The transition intensity between these states is $\alpha_{01}(t)$. Individuals may also be censored, as noted above.

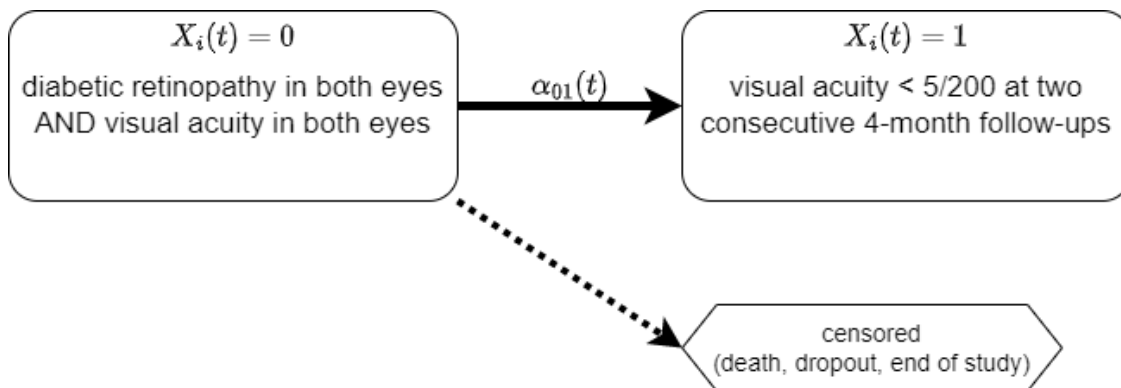


Figure 1: State Diagram: Example 1

3.1.3 Model and Analysis

We take a cox proportional hazards approach, and start by including only one covariate, the treatment indicator:

$$\alpha_{01}^i(t) = \alpha_{01,0}(t)e^{\beta \text{trt}_i}$$

We estimate the model¹ using `coxph` from the `survival` package. Notice we specify the model as $\text{Surv}(t, s) \sim x$ where t is the time-to-event variable, s is an indicator for whether there is censoring, and x could be a series of covariates.

```
fit = coxph(Surv(time, status) ~ trt, data= diabetic)
```

Table 1: Vision Loss in Diabetes Patients

	Hazard Ratio	95% CI		p-value
		Lower	Upper	
trt	0.46	0.33	0.64	4.2e-06

Table 1 gives the results.² The table reports the $e^{\hat{\beta}}$ rather than $\hat{\beta}$ itself. This is because we interpret $e^{\hat{\beta}}$ as the hazard ratio between an eye treated with the lase and an eye not treated. Given such

¹I generate tables using the ‘kableExtra’ R package. See the source Rmd file for the exact syntax.

²Note, because there are two observations per person, we should include the option ‘cluster=id’ in the ‘coxph’ function to obtain robust standard errors. In this case, however, robust standard errors are actually larger than the ones shown here.

a small p-value, we have strong evidence to suggest that the treatment reduces the risk of vision loss.

Perhaps we wish to control for age at time of diagnosis. Then our model is:

$$\alpha_{01}^i(t) = \alpha_{01,0}(t)e^{\beta_1 \text{trt}_i + \beta_2 \text{age}_i}$$

and we estimate it:

```
fit = coxph(Surv(time, status) ~ trt + age, data= diabetic)
```

Table 2: Vision Loss in Diabetes Patients II

	Hazard Ratio	95% CI		p-value
		Lower	Upper	
trt	0.457	0.328	0.637	0.000004
age	1.004	0.993	1.015	0.460988

We see in Table 2 that age is not significant (p-value ≈ 0.461) and so we stick with our original model and findings.

3.2 Competing Risks

3.2.1 Data

The next dataset we consider is the `transplant` dataset from the `survival` package as sourced from [Kim et al. \(2006\)](#). This study consists of 815 patients aged 18 or older registered on the liver transplant waiting list at Mayo Clinic, Rochester, Minnesota over a 10-year period between 1990 and 1999. There were three possible outcomes: the patient could die while waiting on the list, the patient could receive the liver transplant they were waiting for, or the patient could withdraw from the list. If the patient was still waiting at the end of the study, censoring occurred.

Let's take a quick look at the data:

```
data(transplant); head(transplant)
```

```
##   age sex abo year futime   event
## 1  47  m   B 1994   1197  death
## 2  55  m   A 1991    28    ltx
## 3  52  m   B 1996    85    ltx
## 4  40  f   0 1995   231    ltx
## 5  70  m   0 1996  1271 censored
## 6  66  f   0 1996    58    ltx
```

Each row represents one patient with the following covariates:

- `age`: age when entered the waiting list
- `sex`: male or female
- `abo`: blood type
- `year`: year they entered the waiting list
- `ftime`: time of final disposition
- `event`: 1 if censored, 2 if death, 3 if transplant, 4 if withdraw from list

3.2.2 State Diagram

Figure 2 summarizes the multi-state process $X_i(t)$. Our state space is $S = \{0, 1, 2, 3\}$ where 0 represents the baseline state while the other states represent alternative outcomes: 1 represents death, 2 represents lung transplant, and 3 represents the patient withdrawing from the waiting list. There are thus three transition intensities of interest: $\alpha_{01}(t)$, $\alpha_{02}(t)$, and $\alpha_{03}(t)$. Individuals may also be censored, as noted above.

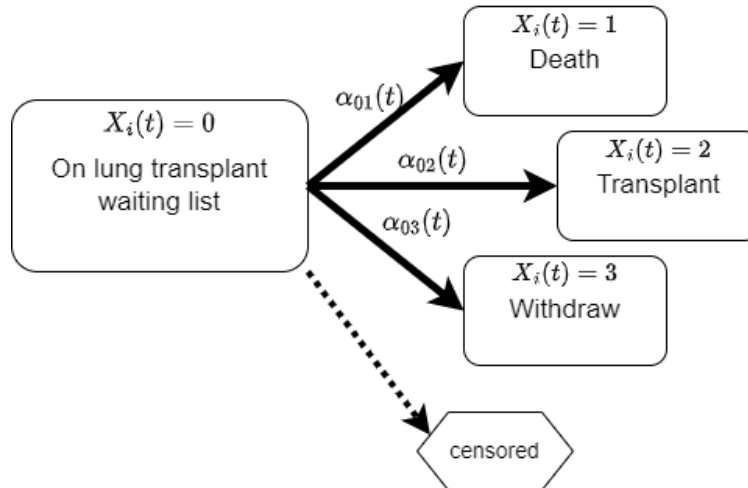


Figure 2: State Diagram: Example 2

3.2.3 Model and Analysis

We now have a model for each of the three intensities:

$$\alpha_{0j}^i(t) = \alpha_{0j,0}(t)e^{\beta_j^T Z_i}; \quad j = 1, 2, 3$$

We now have a vector of coefficients β_j associated with each possible outcome j . Estimating the model is, however, as straightforward as before:

```

fit=coxph(Surv(futime, event) ~ age + sex + abo + year,
          data = transplant,
          id = rownames(transplant))
  
```

Table 3 shows the results of this model for the three absorbing states death, transplant, and withdrawal. Most notably, we see that the year at which the person entered the waiting list reduces the “risk” of all three outcomes. But in fact, this is not strictly a positive thing. In particular, it implies that as time has gone on, the waiting time to obtain a transplant has increased substantially, even after controlling for various factors. We also see that having an O blood type increases one’s expected wait time to obtain a transplant. This has to do with blood type compatibility resulting in a disadvantage for O blood type recipients.

Table 3: Liver Transplant

	Hazard Ratio	95% CI		p-value
		Lower	Upper	
Death				
age_1:2	1.020	0.997	1.044	0.095
sexf_1:2	0.635	0.387	1.042	0.073
aboB_1:2	1.181	0.559	2.493	0.663
aboAB_1:2	1.277	0.401	4.066	0.679
aboO_1:2	1.011	0.577	1.772	0.969
year_1:2	0.833	0.757	0.916	0.000
Transplant				
age_1:3	0.997	0.988	1.006	0.512
sexf_1:3	0.942	0.762	1.164	0.580
aboB_1:3	0.726	0.514	1.026	0.070
aboAB_1:3	1.093	0.628	1.902	0.753
aboO_1:3	0.556	0.444	0.697	0.000
year_1:3	0.738	0.690	0.789	0.000
Withdraw				
age_1:4	0.981	0.946	1.017	0.287
sexf_1:4	1.378	0.733	2.593	0.319
aboB_1:4	1.620	0.542	4.843	0.388
aboAB_1:4	3.495	0.935	13.073	0.063
aboO_1:4	1.257	0.548	2.882	0.589
year_1:4	0.809	0.715	0.917	0.001

3.3 Illness-Death Models

3.3.1 Data

The next dataset we consider is the `colon` dataset from the `survival` package, as sourced from [Laurie et al. \(1989\)](#); [Moertel et al. \(1990\)](#). The goal of the study was to examine two treatments—levasimole and 5-FU—and their effect on colon cancer. Each patient at baseline has cancer. There are two possible outcomes: recurrence of the cancer and death, and they may die before recurrence occurs.

Let's take a quick look at the data:

```
data(colon); head(colon)
```

```
##   id study      rx sex age obstruct perfor adhere nodes status differ extent
## 1  1     1 Lev+5FU  1  43         0      0      0     5      1      2      3
## 2  1     1 Lev+5FU  1  43         0      0      0     5      1      2      3
## 3  2     1 Lev+5FU  1  63         0      0      0     1      0      2      3
## 4  2     1 Lev+5FU  1  63         0      0      0     1      0      2      3
```

```

## 5 3 1 Obs 0 71 0 0 1 7 1 2 2
## 6 3 1 Obs 0 71 0 0 1 7 1 2 2
## surg node4 time etype
## 1 0 1 1521 2
## 2 0 1 968 1
## 3 0 0 3087 2
## 4 0 0 3087 1
## 5 0 1 963 2
## 6 0 1 542 1

```

We see that each patient has two rows corresponding to the two possible events. The covariates represent (some omitted):

- rx: factor variable with three treatments (baseline, levasimole, levasimole and 5-FU)
- status: 0 if censored, 1 if event occurs
- time,stop: time of observation (or censoring time)
- etype: 1 for recurrence, 2 for death

3.3.2 State Diagram

Figure 3 summarizes the multi-state process $X_i(t)$. Our state space is $S = \{0, 1, 2\}$ where 0 represents the baseline state, 1 represents cancer recurrence, and 2 represents death. We see that there are three intensities associated with “baseline to recurrence”, “recurrence to death”, and “baseline to death”.

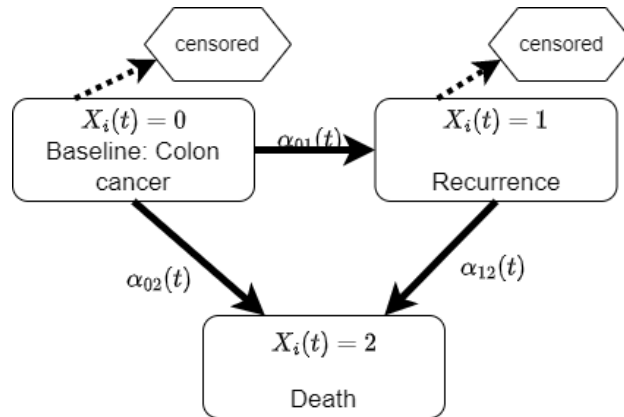


Figure 3: State Diagram: Example 3

3.3.3 Model and Analysis

We specify models for the three intensities:

$$\begin{aligned}
\alpha_{01}^i(t) &= \alpha_{01,0}(t)e^{\beta_{01}^T Z_i} \\
\alpha_{02}^i(t) &= \alpha_{02,0}(t)e^{\beta_{02}^T Z_i} \\
\alpha_{12}^i(t) &= \alpha_{12,0}(t)e^{\beta_{12}^T Z_i}
\end{aligned}$$

The example data from the survival package is already in the form we need to estimate such a model. The only thing we need is an indicator to distinguish whether or death occurred after a recurrence:

```
colon_v2 = colon %>%
  mutate(is_recur=as.numeric(etype==1 & status==1)) %>%
  group_by(id) %>%
  mutate(sum_recur = sum(is_recur)) %>%
  ungroup() %>%
  mutate(strat = case_when(etype==1 ~ 1,
                           etype==2 & sum_recur==1 ~ 2,
                           etype==2 & sum_recur==0 ~ 3))
coxph(Surv(time,status) ~ rx*factor(strat) + strata(strat),
      data = colon_v2, id = id)
```

```
## Call:
## coxph(formula = Surv(time, status) ~ rx * factor(strat) + strata(strat),
##       data = colon_v2, id = id)
##
##               coef exp(coef) se(coef) robust se      z      p
## rxLev          -0.01512   0.98499  0.10708   0.10726 -0.141   0.888
## rxLev+5FU      -0.51209   0.59924  0.11863   0.11813 -4.335 1.46e-05
## factor(strat)2         NA         NA  0.00000   0.00000    NA    NA
## factor(strat)3         NA         NA  0.00000   0.00000    NA    NA
## rxLev:factor(strat)2   0.06173   1.06367  0.15671   0.14465  0.427   0.670
## rxLev+5FU:factor(strat)2 0.64020   1.89686  0.17270   0.15967  4.009 6.09e-05
## rxLev:factor(strat)3  -0.28098   0.75504  0.43420   0.42713 -0.658   0.511
## rxLev+5FU:factor(strat)3 0.31711   1.37315  0.39739   0.39069  0.812   0.417
##
## Likelihood ratio test=25.91 on 6 df, p=0.0002318
## n= 1858, number of events= 920
```

Here I have coded 1 as the “baseline to recurrence” stratum, 2 as the “recurrence to death” stratum, and 3 as the “baseline to death” stratum. We thus see that the Levasimole + 5-FU treatment seems to decrease one’s likelihood of recurrence, but increase the likelihood of death once recurrence does occur.

Note that as mentioned previously, we could include time spent in recurrence as a time-dependent covariate affecting the probability of death. In the next example, we will do something similar, and so I leave it here for now.

3.4 Recurrent Events

3.4.1 Data

The next dataset we consider is the bladder dataset from the `survival` package as sourced from [Byar \(1980\)](#). This data consists of 118 subjects who had bladder tumors removed at the start of the study. There were then randomized to one of two treatments (pyrodoxine or thioteps) or to a placebo. The patients' recurrence of bladder cancer was tracked over time, with the tumors being removed at each occurrence.

Let's take a quick look at the data:

```
data(bladder1); head(bladder1)
```

```
##   id treatment number size recur start stop status rtumor rsize enum
## 1  1 placebo      1    1    0    0    0     3     .     .     1
## 2  2 placebo      1    3    0    0    1     3     .     .     1
## 3  3 placebo      2    1    0    0    4     0     .     .     1
## 4  4 placebo      1    1    0    0    7     0     .     .     1
## 5  5 placebo      5    1    0    0   10     3     .     .     1
## 6  6 placebo      4    1    1    0    6     1     1     1     1
```

Each row represents a recurrence of a patient. Thus, each patient has a varying number of rows. The covariates represent (some omitted):

- `treatment`: factor variable with the three treatments
- `recur`: total number of recurrences
- `start, stop`: the start and stop times of the interval
- `status`: 0 if censored, 1 if recurrence, 2 if death from bladder disease, 3 if death from other cause
- `enum`: observation number of the patient

3.4.2 State Diagram

Figure 4 summarizes the multi-state process $X_i(t)$. Our state space is $S = \{-1, 0, 1, 2, 3, \dots\}$ where 0 represents the baseline state, -1 represents death, and $1, 2, 3, \dots$ represent the number of occurrences of cancer. In theory, at least, there are a countably infinite number of transition intensities! We will deal with this in the next section. Note that individuals may also be censored. As well, we will combine the two causes of death in the data into one cause of death.

3.4.3 Model and Analysis

Given there are technically an infinite number of transition intensities, we will need to make assumptions about the process to make it tractable. We consider three options here³. In each case, let $\alpha_{r,r+1}^{iR}$ be the intensity between the r^{th} recurrence and the $r + 1^{\text{th}}$ recurrence. Further, let α_r^{iD} be the intensity between the r^{th} recurrence and death.

³Inspired by these slides: <https://myweb.uiowa.edu/pbreheny/7210/f15/notes/12-10.pdf>

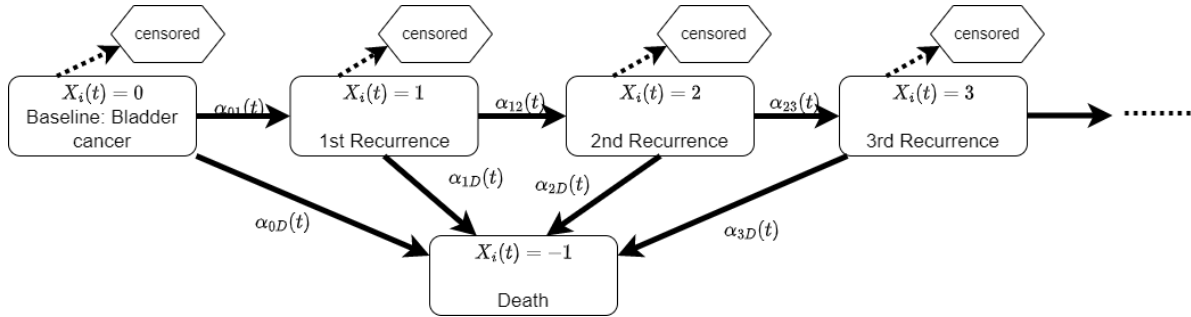


Figure 4: State Diagram: Example 3

OPTION 1: Behaviour at each recurrence is the same. Here we assume transition intensities are the same regardless of how many recurrences have occurred so far. That is,

$$\alpha_{r,r+1}^{iR}(t) = \alpha_0^R(t) e^{\beta_R^T Z_i}$$

$$\alpha_r^{iD}(t) = \alpha_0^D(t) e^{\beta_D^T Z_i}.$$

OPTION 2: Include the number of previous recurrences as a covariate. Here we allow for the possibility that an increase in the number of previous recurrences r is associated with the probability of further recurrence and/or death. That is,

$$\alpha_{r,r+1}^{iR}(t) = \alpha_0^R(t) e^{\beta_R^T Z_i + \gamma_R r}$$

$$\alpha_r^{iD}(t) = \alpha_0^D(t) e^{\beta_D^T Z_i + \gamma_D r}.$$

OPTION 3: Flexible baseline intensities. Here we allow the baseline intensity function to differ given the number of previous recurrences. This is equivalent to using the number of recurrences as a stratum variable in coxph. That is,

$$\alpha_{r,r+1}^{iR}(t) = \alpha_{r,0}^R(t) e^{\beta_R^T Z_i}$$

$$\alpha_r^{iD}(t) = \alpha_{r,0}^D(t) e^{\beta_D^T Z_i}.$$

You may notice that this example holds a strong resemblance to the illness-death model of the previous section. In fact, this is exactly an illness-death model only with multiple recurrences of the illness! However, unlike in the previous example, the data we are using is not already in the form we need! To estimate these models, we will first need to convert the data to a “long” format. That is, for each event (recurrence or death) I create a duplicate row with the event that did not occur:

```
bladder.expand = bladder1 %>%
  filter(!(start==0 & stop==0)) %>% # remove those apparently observed for 0 time
  mutate(status=ifelse(status==3,2,status)) %>%
  mutate(event=as.numeric(status>0))
df1 = bladder.expand %>%
```

```

  filter(status==2) %>%
  mutate(status=1) %>%
  mutate(event=0)
df2 = bladder.expand %>%
  filter(status==1) %>%
  mutate(status=2) %>%
  mutate(event=0)
df3 = bladder.expand %>%
  filter(status==0) %>%
  mutate(status=1) %>%
  mutate(event=0)
df4 = bladder.expand %>%
  filter(status==0) %>%
  mutate(status=2) %>%
  mutate(event=0)
bladder.expand = bladder.expand %>% filter(status>0)
bladder.expand = rbind(bladder.expand,df1,df2)
bladder.expand = bladder.expand[order(bladder.expand$id),]
head(bladder.expand)

```

```

##      id treatment number size recur start stop status rtumor rsize enum event
## 1    2 placebo      1    3     0    0    1     2     .     .     1     1
## 218  2 placebo      1    3     0    0    1     1     .     .     1     0
## 2    5 placebo      5    1     0    0   10     2     .     .     1     1
## 219  5 placebo      5    1     0    0   10     1     .     .     1     0
## 3    6 placebo      4    1     1    0    6     1     1     1     1     1
## 4    6 placebo      4    1     1    6   10     2     .     .     2     1

```

Notice that each time range from each individual in the original data now has two rows—one for recurrence and one for death. Further, the event indicator tell us which of these two (if any) occurred.

With that we can estimate each type of model:

```

# OPTION 1
bladder.expand = bladder.expand %>%
  mutate(event_recur=as.numeric(status==1)) %>%
  mutate(event_death=as.numeric(status==2)) %>%
  mutate(treatment=as.numeric(treatment)) %>%
  mutate(pyrid = as.numeric(treatment==2)) %>%
  mutate(thio = as.numeric(treatment==3))
coxph(Surv(start,stop,event) ~ pyrid:event_recur + thio:event_recur +
      pyrid:event_death + thio:event_death +
      strata(status),
      data = bladder.expand, id = id)

```

Call:


```
## coxph(formula = Surv(start, stop, event) ~ pyrid:event_recur +
##       thio:event_recur + pyrid:event_death + thio:event_death +
##       strata(status), data = bladder.expand, id = id)
##
##               coef exp(coef) se(coef) robust se      z      p
## pyrid:event_recur  0.1450    1.1560   0.1722    0.2288  0.633 0.526
## event_recur:thio -0.3246    0.7228   0.1845    0.2601 -1.248 0.212
## pyrid:event_death  0.1201    1.1276   0.4958    0.4993  0.240 0.810
## thio:event_death   0.3177    1.3739   0.4491    0.4394  0.723 0.470
##
## Likelihood ratio test=6.44  on 4 df, p=0.1688
## n= 434, number of events= 217
```

```
# OPTION 2a
bladder.expand = bladder.expand %>%
  mutate(r = enum-1)
coxph(Surv(start,stop,event) ~ pyrid:event_recur + thio:event_recur + r:event_recur +
      pyrid:event_death + thio:event_death + r:event_death +
      strata(status),
      data = bladder.expand, id = id)
```

```
## Call:
## coxph(formula = Surv(start, stop, event) ~ pyrid:event_recur +
##       thio:event_recur + r:event_recur + pyrid:event_death + thio:event_death +
##       r:event_death + strata(status), data = bladder.expand, id = id)
##
##               coef exp(coef) se(coef) robust se      z      p
## pyrid:event_recur  0.10980    1.11606  0.17328    0.17024  0.645 0.518921
## event_recur:thio -0.17310    0.84105  0.19008    0.21748 -0.796 0.426057
## event_recur:r      0.14697    1.15832  0.04538    0.04428  3.319 0.000903
## pyrid:event_death  0.12515    1.13332  0.49608    0.50036  0.250 0.802494
## thio:event_death   0.27693    1.31907  0.47157    0.48992  0.565 0.571901
## r:event_death     -0.03227    0.96825  0.11418    0.12443 -0.259 0.795392
##
## Likelihood ratio test=17.22  on 6 df, p=0.008504
## n= 434, number of events= 217
```

```
# OPTION 2b
coxph(Surv(start,stop,event) ~ r:event_recur + pyrid:event_recur + thio:event_recur + r:pyrid:ev
      r:event_death + pyrid:event_death + thio:event_death + r:pyrid:ev
      strata(status),
      data = bladder.expand, id = id)
```

```
## Call:
## coxph(formula = Surv(start, stop, event) ~ r:event_recur + pyrid:event_recur +
##       thio:event_recur + r:pyrid:event_recur + r:thio:event_recur +
```

```

##      r:event_death + pyrid:event_death + thio:event_death + r:pyrid:event_death +
##      r:thio:event_death + strata(status), data = bladder.expand,
##      id = id)
##
##              coef exp(coef) se(coef) robust se      z      p
## r:event_recur      0.06691   1.06920  0.06337   0.05164   1.296  0.19505
## event_recur:pyrid  -0.05788   0.94376  0.23269   0.29296  -0.198  0.84338
## event_recur:thio   -0.51171   0.59947  0.23950   0.27934  -1.832  0.06697
## r:event_death     -0.06890   0.93342  0.17309   0.16300  -0.423  0.67253
## pyrid:event_death  0.37853   1.46013  0.60951   0.63819   0.593  0.55310
## thio:event_death  -0.20602   0.81381  0.61058   0.59071  -0.349  0.72726
## r:event_recur:pyrid 0.08876   1.09282  0.07242   0.06780   1.309  0.19047
## r:event_recur:thio 0.25512   1.29061  0.10317   0.08425   3.028  0.00246
## r:pyrid:event_death -0.15885   0.85313  0.25033   0.28851  -0.551  0.58193
## r:thio:event_death  0.42468   1.52910  0.24983   0.20729   2.049  0.04049
##
## Likelihood ratio test=27.98 on 10 df, p=0.001819
## n= 434, number of events= 217

```

OPTION 3

```

bladder.expand = bladder.expand %>%
  mutate(num_recur_cat = pmin(r,4))
coxph(Surv(start,stop,event) ~ pyrid:event_recur + thio:event_recur +
      pyrid:event_death + thio:event_death +
      strata(status) + strata(num_recur_cat),
      data = bladder.expand, id = id)

```

```

## Call:
## coxph(formula = Surv(start, stop, event) ~ pyrid:event_recur +
##      thio:event_recur + pyrid:event_death + thio:event_death +
##      strata(status) + strata(num_recur_cat), data = bladder.expand,
##      id = id)
##
##              coef exp(coef) se(coef) robust se      z      p
## pyrid:event_recur  0.21889   1.24470  0.18605   0.16813   1.302  0.193
## event_recur:thio  -0.30806   0.73487  0.20918   0.19784  -1.557  0.119
## pyrid:event_death  0.08255   1.08606  0.50596   0.49219   0.168  0.867
## thio:event_death  0.19766   1.21855  0.50085   0.46907   0.421  0.673
##
## Likelihood ratio test=5.33 on 4 df, p=0.2549
## n= 434, number of events= 217

```

Out of these models, the most interesting seems to be Model 2b. We see that there appears to be an interaction effect between the treatment and the number of recurrences. Specifically, there appears to be some evidence that the thiotepa treatment causes an initial decline in recurrence, but that this effect disappears as the number of recurrences increases. In models without these interaction terms, the treatment indicators were not significant in either the recurrence or the death intensities.

3.5 Bonus: Stratification and Parameter Tying

In [Andersen and Keiding \(2002\)](#), Section 5.6, they provide an example of a model incorporating two ideas:

- **stratification**: where the baseline intensity is estimated separately within each stratum
- **parameter tying**: where the model is made parsimonious by fixing coefficients in different intensities to be the same

Here, I briefly show how to conduct each of these methods.

3.5.1 Stratification

Stratification is quite simple to implement using the `coxph` function. Take as an example the diabetic dataset from Section 3.1. We would like to consider the following model:

$$\alpha_{01}^i(t) = \begin{cases} \alpha_0^1(t)e^{\beta \text{trt}_i}, & \text{laser=argon} \\ \alpha_0^2(t)e^{\beta \text{trt}_i}, & \text{laser=xenon} \end{cases}$$

That is, we believe that the baseline intensity differs between the type of laser used in treatment. We may still estimate each baseline intensity non-parametrically by specifying `strata(laser)`:

```
coxph(Surv(time, status) ~ trt + strata(laser), data= diabetic)
```

```
## Call:
## coxph(formula = Surv(time, status) ~ trt + strata(laser), data = diabetic)
##
##          coef exp(coef) se(coef)      z      p
## trt -0.7831    0.4570   0.1692 -4.628 3.68e-06
##
## Likelihood ratio test=22.64 on 1 df, p=1.953e-06
## n= 394, number of events= 155
```

Comparing with Table 1, we see that the treatment effect is very similar, suggesting that estimating baseline intensity by laser type did not have a big effect.

3.5.2 Parameter Tying

Consider a simplified version of the competing risks model from Section 3.2 including only covariates for sex and year:

```
coxph(Surv(futime, event) ~ sex + year,
      data = transplant,
      id = rownames(transplant))
```

```
## Call:
```

```

## coxph(formula = Surv(futime, event) ~ sex + year, data = transplant,
##       id = rownames(transplant))
##
##
## 1:2      coef exp(coef) se(coef) robust se      z      p
##   sexf -0.44043   0.64376 0.25825   0.25722 -1.712 0.08685
##   year -0.14797   0.86245 0.05085   0.04730 -3.128 0.00176
##
##
## 1:3      coef exp(coef) se(coef) robust se      z      p
##   sexf -0.12331   0.88399 0.08049   0.08998  -1.37 0.171
##   year -0.27717   0.75792 0.01567   0.02451 -11.31 <2e-16
##
##
## 1:4      coef exp(coef) se(coef) robust se      z      p
##   sexf  0.28057   1.32388 0.33321   0.32661  0.859 0.39032
##   year -0.16323   0.84939 0.06871   0.05451 -2.995 0.00275
##
## States:  1= (s0), 2= death, 3= ltx, 4= withdraw
##
## Likelihood ratio test=305.4  on 6 df, p=< 2.2e-16
## n= 815, number of events= 739

```

We see that the coefficients for year are quite similar within all three categories. It may be desirable to consider a more parsimonious model where all three categories share the same coefficient for year. I.e.:

$$\alpha_{0j}^i(t) = \alpha_{0j,0}(t)e^{\beta_j \text{female}_i + \gamma \text{year}_i}; \quad j = 1, 2, 3.$$

with only 4 parameters $\beta_1, \beta_2, \beta_4, \gamma$

To estimate such a model we must first convert the data into “long” format. I do so here using the `mstate` package:

```

transplant.expand = transplant %>%
  mutate(event_death=as.numeric(as.numeric(event)==2)) %>%
  mutate(event_ltx=as.numeric(as.numeric(event)==3)) %>%
  mutate(event_withdraw=as.numeric(as.numeric(event)==4))
transplant.expand = mstate::msprep(time=c(NA, "futime", "futime", "futime"),
  status=c(NA, "event_death", "event_ltx", "event_withdraw"),
  data=transplant.expand,
  trans=trans.comprisk(3),
  keep=c("sex", "year"))
transplant.expand = transplant.expand %>%
  mutate(e_death = as.numeric(to==2)) %>%
  mutate(e_ltx = as.numeric(to==3)) %>%

```

```
mutate(e_withdraw = as.numeric(to==4)) %>%
mutate(female=as.numeric(as.numeric(sex)==2))

head(transplant.expand)
```

```
## An object of class 'msdata'
##
## Data:
##   id from to trans Tstart Tstop time status sex year e_death e_ltx e_withdraw
## 1  1  1  2    1      0  1197 1197     1  m 1994     1    0         0
## 2  1  1  3    2      0  1197 1197     0  m 1994     0    1         0
## 3  1  1  4    3      0  1197 1197     0  m 1994     0    0         1
## 4  2  1  2    1      0    28   28     0  m 1991     1    0         0
## 5  2  1  3    2      0    28   28     1  m 1991     0    1         0
## 6  2  1  4    3      0    28   28     0  m 1991     0    0         1
##   female
## 1      0
## 2      0
## 3      0
## 4      0
## 5      0
## 6      0
```

We see that the data now contains 3 rows for each individual, one for each possible transition. We can replicate the original results using interaction terms and stratification:

```
coxph(Surv(time, status) ~ female:e_death + year:e_death +
      female:e_ltx + year:e_ltx +
      female:e_withdraw + year:e_withdraw +
      strata(to) ,
      data = transplant.expand,
      id = id)
```

```
## Call:
## coxph(formula = Surv(time, status) ~ female:e_death + year:e_death +
##       female:e_ltx + year:e_ltx + female:e_withdraw + year:e_withdraw +
##       strata(to), data = transplant.expand, id = id)
##
##               coef exp(coef) se(coef)      z      p
## female:e_death -0.44043   0.64376  0.25825  -1.705 0.08812
## e_death:year   -0.14797   0.86245  0.05085  -2.910 0.00361
## female:e_ltx   -0.12331   0.88399  0.08049  -1.532 0.12551
## year:e_ltx     -0.27717   0.75792  0.01567 -17.691 < 2e-16
## female:e_withdraw 0.28057   1.32388  0.33321   0.842 0.39979
## year:e_withdraw -0.16323   0.84939  0.06871  -2.376 0.01751
```

```
##
## Likelihood ratio test=305.4 on 6 df, p=< 2.2e-16
## n= 2445, number of events= 739
```

The variable `to`, automatically generated by `msprep`, represents the three competing risk outcomes. By “stratifying” on it, we have gotten back to our original model where each outcome is associated with its own baseline intensity function. Similarly, while `coxph` originally estimated separate parameters for each outcome, we now must specify this manually by interacting each covariate with event indicators `e_death`, `e_ltx`, and `e_withdraw`.

To conduct parameter tying, we simply include `year` as a main effect:

```
coxph(Surv(time, status) ~ female:e_death + female:e_ltx + female:e_withdraw +
      year +
      strata(to) ,
      data = transplant.expand,
      id = id)
```

```
## Call:
## coxph(formula = Surv(time, status) ~ female:e_death + female:e_ltx +
##       female:e_withdraw + year + strata(to), data = transplant.expand,
##       id = id)
##
##               coef exp(coef) se(coef)      z      p
## year            -0.26056   0.77062  0.01461 -17.832 <2e-16
## female:e_death  -0.52005   0.59449  0.25636  -2.029 0.0425
## female:e_ltx    -0.11106   0.89488  0.08033  -1.383 0.1668
## female:e_withdraw 0.21190   1.23603  0.33070   0.641 0.5217
##
## Likelihood ratio test=297.2 on 4 df, p=< 2.2e-16
## n= 2445, number of events= 739
```

Similarly, we can assume that all outcomes share the same baseline intensity function, up to a proportionality constant, by moving ‘`to`’ from a stratum variable to a normal variable:

```
coxph(Surv(time, status) ~ female:e_death + female:e_ltx + female:e_withdraw +
      year +
      factor(to) ,
      data = transplant.expand,
      id = id)
```

```
## Call:
## coxph(formula = Surv(time, status) ~ female:e_death + female:e_ltx +
##       female:e_withdraw + year + factor(to), data = transplant.expand,
##       id = id)
```

```

##
##           coef exp(coef) se(coef)      z      p
## year      -0.26058  0.77060  0.01462 -17.826 < 2e-16
## factor(to)3  2.09732  8.14429  0.16353  12.825 < 2e-16
## factor(to)4 -0.90457  0.40472  0.28746  -3.147 0.00165
## female:e_death -0.52443  0.59189  0.25614  -2.047 0.04061
## female:e_ltx  -0.10982  0.89600  0.08033  -1.367 0.17161
## female:e_withdraw 0.19786  1.21879  0.33008   0.599 0.54890
##
## Likelihood ratio test=1191 on 6 df, p=< 2.2e-16
## n= 2445, number of events= 739

```

References

- Andersen, Per Kragh and Niels Keiding. 2002. "Multi-state models for event history analysis." *Statistical methods in medical research* 11(2):91–115.
- Byar, DP. 1980. "The veterans administration study of chemoprophylaxis for recurrent stage i bladder tumours: comparisons of placebo, pyridoxine and topical thiotepa." *Bladder tumors and other topics in urological oncology* pp. 363–370.
- Huster, William J, Ron Brookmeyer and Steven G Self. 1989. "Modelling paired survival data with covariates." *Biometrics* pp. 145–156.
- Kim, W Ray, Terry M Therneau, Joanne T Benson, Walter K Kremers, Charles B Rosen, Gregory J Gores and E Rolland Dickson. 2006. "Deaths on the liver transplant waiting list: an analysis of competing risks." *Hepatology* 43(2):345–351.
- Laurie, John A, Charles G Moertel, Thomas R Fleming, Harry S Wieand, John E Leigh, Jebal Rubin, Greg W McCormack, James B Gerstner, James E Krook and James Malliard. 1989. "Surgical adjuvant therapy of large-bowel carcinoma: an evaluation of levamisole and the combination of levamisole and fluorouracil. The North Central Cancer Treatment Group and the Mayo Clinic." *Journal of Clinical Oncology* 7(10):1447–1456.
- Moertel, Charles G, Thomas R Fleming, John S Macdonald, Daniel G Haller, John A Laurie, Phyllis J Goodman, James S Ungerleider, William A Emerson, Douglas C Tormey, John H Glick et al. 1990. "Levamisole and fluorouracil for adjuvant therapy of resected colon carcinoma." *New England Journal of Medicine* 322(6):352–358.