

Riesgos_Competitivos

Introducción y preeliminares: Descripción de la base elegida

En este trabajo, abordamos el análisis de supervivencia de riesgos competitivos, por lo que el esquema de trabajo cambia un poco respecto al visto en clase, que es realizar el modelo de Cox.

Trabajaremos con la base `sir.adm`; disponible en el paquete “`mvna`”, llamado así por las iniciales de “Multivariate Nelson-Aalen estimator”, que permite estimar de manera no paramétrica los riesgos acumulados de transición de modelos de Markov multi-estados arbitrarios, usando dicho estimador.

```
library(mvna)
```

```
## Warning: package 'mvna' was built under R version 4.1.2
```

Cargamos la base de datos

```
data(sir.adm)
```

Veamos las primeras 5 observaciones

```
head(sir.adm)
```

```
##      id pneu status time      age sex
## 1   41     0      1    4 75.34153  F
## 2  395     0      1   24 19.17380  M
## 3  710     1      1   37 61.56568  M
## 4 3138     0      1    8 57.88038  F
## 5 3154     0      1    3 39.00639  M
## 6 3178     0      1   24 70.27762  M
```

El dataset contiene una submuestra aleatoria de 747 pacientes y 6 variables:

`id`: Es un id generado aleatoriamente para cada paciente

`pneu`: Es un indicador de si el paciente presentaba neumonía (1) o no la presentaba (0) al momento de ser admitido en el estudio

`status`: Un indicador del estatus de la observación: 0 es una observación censurada, 1 es que el paciente fue dado de alta, 2 es que el paciente murió

`time`: Es el tiempo en días

`age`: La edad del paciente cuando entró al estudio

`sex`: F para mujer, M para hombre.

```
sum(is.na.data.frame(sir.adm))
```

```
## [1] 0
```

Vemos que no hay ningún dato faltante (NA) en el dataset.

Estas 747 pacientes son de SIR 3 (Spread of nosocomial Infections and Resistant pathogens, es decir, Esparcimiento de infecciones nosocomiales y patógenos resistentes), un estudio conjunto en el hospital Charité de la Universidad de Berlin, Alemania, con una valoración prospectiva de datos para examinar el efecto de infecciones adquiridas en el hospital, en terapia-intensiva.

Notemos que se presenta una censura por la derecha.

El dataset contiene información en el estado de admisión de neumonía, tiempo de estadía en la unidad de terapia intensiva y “desenlace o resultado del tratamiento en unidad intensiva”, es decir, si se le dio de alta al paciente o este murió.

La neumonía es una infección severa, de la cual se tiene la sospecha, causa el requerimiento de cuidados adicionales (Es decir, una prolongada estadía en unidades de cuidado intensivo) y de incrementar la mortalidad.

```
sum(sir.adm$status==0)
```

```
## [1] 14
```

14 Observaciones censuradas, es decir, que seguían en unidad intensiva al final del estudio.

```
sum(sir.adm$status==1)
```

```
## [1] 657
```

657 Pacientes que fueron dados de alta

```
sum(sir.adm$status==2)
```

```
## [1] 76
```

76 pacientes fallecieron

```
sum(sir.adm$status==2 & sir.adm$pneu==1)
```

```
## [1] 21
```

21 de los pacientes que murieron tenían neumonía al momento de ser admitidos.

```
sum(sir.adm$status==2 & sir.adm$pneu==0)
```

```
## [1] 55
```

55 de los pacientes que murieron, no tenían neumonía al momento de ser admitidos

Esta base la elegimos porque nos parece un buen ejemplo de riesgos competitivos por lo siguiente:

Se investiga el tiempo hasta el final de la estadía y el estatus de el estado final: Si se dio de alta o murió en el hospital. Un desafío en el análisis de este dataset es que se ha encontrado que la neumonía incrementa la probabilidad de morir en el hospital, pero parece no tener efecto en el *riesgo* de muerte, es decir, no tiene efecto en la probabilidad diaria de morir en el hospital, dado que uno sigue vivo y en la unidad de cuidados intensivos al inicio del día.

Los puntos finales competitivos son el ser dado de alta y morir en la unidad de terapia intensiva

Análisis descriptivo

Función de incidencia acumulada

Necesitamos modificar primero el dataset en un dataset de tipo multiestado. Además, renombramos los datos para que, el evento de interés que es la muerte, corresponda al estado 1, y el estado 2 el evento competitivo:

```
#Transformamos sir.adm a un dataset de tipo multiestado
to <- ifelse(sir.adm$status == 0, "cens", ifelse(sir.adm$status == 1, 2, 1))
my.sir.data <- data.frame(id = sir.adm$id, from = 0, to, time = sir.adm$time, pneu = sir.adm$pneu)
```

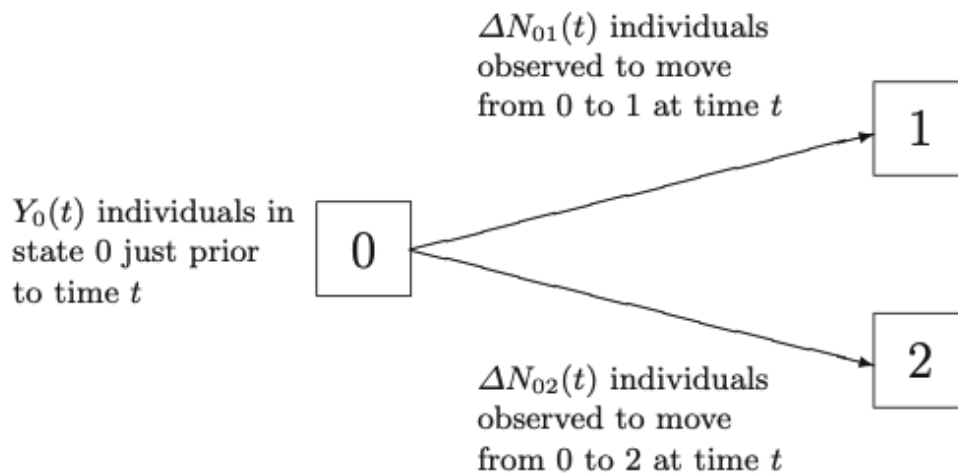
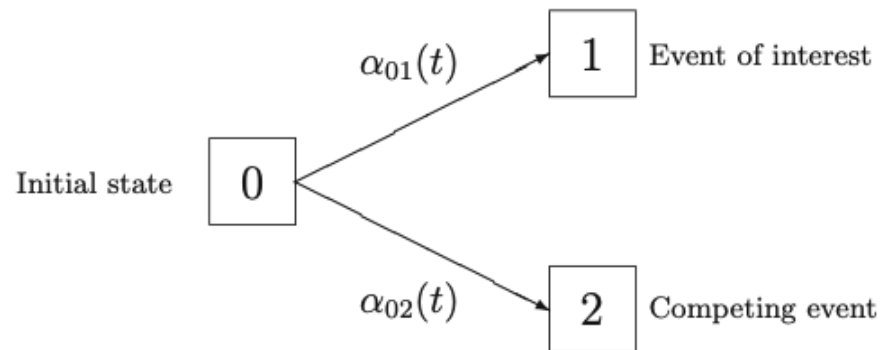
Notemos que my.sir.data tiene un componente pneu con el estatus de neumonía al momento de admisión. Revisamos que hicimos bien el renombramiento:

```
table(my.sir.data$to)
```

```
##
##      1      2 cens
##    76  657   14
```

Lo hicimos de manera correcta.

Antes de continuar, necesitamos describir el modelo de riesgos competitivos multiestado siguiente:



Lo hacemos definiendo una matriz de valores lógicos indicando los posibles tipos de transición en nuestro modelo multiestado:

```
tra <- matrix(FALSE, ncol = 3, nrow = 3)
dimnames(tra) <- list(c("0", "1", "2"), c("0", "1", "2"))
tra[1, 2:3] <- TRUE
tra
```

```
##      0      1      2
## 0 FALSE  TRUE  TRUE
## 1 FALSE FALSE FALSE
## 2 FALSE FALSE FALSE
```

Esta matriz nos dice que un individuo se puede mover del estado 0 al estado 1, y del estado 0 al estado 2, pero las transiciones en el sentido contrario no son posibles; además los valores en diagonal están como falsos: Las transiciones de un estado a sí mismo no están modeladas. No es necesario un modelo para dicha “transición”. Los individuos que no hacen una transición a uno de los dos pares competitivos al tiempo t , permanece en el estado inicial 0 a tiempo t .

Ahora, calculamos la función de acumulación de riesgos específicos para muerte y dados de alta, respectivamente, y estratificados por su estatus de neumonía al momento de admisión:

```
## sin neumonía
my.nelaal.nop <- mvna(my.sir.data[my.sir.data$pneu == 0, ], c("0", "1", "2"), tra, "cens")
## con neumonía
my.nelaal.p <- mvna(my.sir.data[my.sir.data$pneu == 1, ], c("0", "1", "2"), tra, "cens")
```

Graficamos:

```
library(lattice)
ltheme <- canonical.theme(color = FALSE)
## in-built B&W theme
ltheme$strip.background$col <- "white"
## change strip bg
lattice.options(default.theme = ltheme)
## set as default

#dessin.nop <- xyplot(my.nelaal.nop, tr.choice = c("0 2", "0 1"), lwd = 2, layout = c(1, 2), strip = st
```