

# Exercici d'events recurrents en R (part II)

Jose Calatayud Mateu

```
## Llibreria
library(devtools)
library(tidyverse)
devtools::install_github("isglobal-brge/survrec",
                          build = FALSE)

library(survrec)
library(kableExtra)
```

Procedim a descarregar manualment l'arxiu associats als paquet "gcmrec" que està dedicat a funcions de modelatge de models generals per events recurrents i el data sobre el qual treballarem "lymphoma":

```
source(file="lymphoma.R")
source(file="survrec.R")
source(file="gcmrec.R")
```

El data "lymphoma" contenen els temps de recaiguda del càncer després del primer tractament en pacients diagnosticats amb lymphoma de grau baix.

```
head(lymphoma)
```

##	id	time	event	enum	delay	age	sex	distrib	effage
## 1	6	3.900826	0	1	17	79	1	1	CR
## 2	7	63.173554	0	1	33	25	1	1	CR
## 3	8	41.289256	0	1	26	37	1	2	CR
## 4	11	29.421488	1	1	31	43	2	2	CR
## 5	11	20.826446	1	2	31	43	2	2	CR
## 6	11	17.950413	0	3	31	43	2	2	CR

*NOTA:* la variable *time* conté els temps entre esdeveniments, *event* és la variable de censura que val 1 per a recaigudes de càncer i 0 per al darrer moment de seguiment (indicant que l'esdeveniment no s'ha observat), i la variable *id* identifica cada pacient.

## Exercici 1:

- *Estime los modelos AG, PWP-Gap time, PWP-Total time y WLW para determinar si existen diferencias en el tiempo de recaída en función del número de lesiones al momento del diagnóstico (variable distrib). NOTA: Deberá crear los marcos de datos adecuados para los modelos PWT-Gap time y WLW.*

Utilizando los datos de lymphoma (package **gcmrec**), determinar si hay diferencias en el temps de recaiguda segons el número de lesions al diagnòstic (variable **distrib**). Per dur-ho a terme, es tenen que estimar els models AG, PWP-Gap time, PWP-Total time y WLW.

El conjunt de dades contenen les següents variables:

- **id**: identificador del pacient
- **time**: temps entre recaiguda
- **event**: variable de censura (1 = recaiguda, 0 = últim seguiment sense observar res).
- **distrib**: nombre de lesions al diagnostic (0 = única, 1 = localitzada, 2 = més d'un lloc nodal, 3 = generalitzada).
- **sex**: sexe del pacient.
- **treat**: resposta al tractament.
- **age**: edat del pacient.

## Preparar los datos para los diferentes modelos

**1. Model AG (Andersen-Gill)** El model AG és una extensió del model de Cox proporcional de risc per a dades d'events recurrents. El seu objectiu és modelar el risc d'ocurrència d'un event recurrent, considerant que un subjecte pot experimentar múltiples events durant el període de seguiment. A continuació, es detalla la seua estructura i suposits

El model AG assumeix que els events recurrents d'un subjecte segueixen un procés de recompte amb increments independents. El risc per al k-èssim event del subjecte i es modela com:

$$\lambda_k(t; Z_{ik}) = \lambda_0(t)e^{\beta' Z_{it}(t)}$$

d'on:

- $\lambda_0(t)$  és la funció de risc basal comú a tots els individus
- $Z_{ik}(t)$  és un vector de covariables per al individu i en el k-èssim event.
- $\beta$  és el vector de paràmetres a estimar

El model AG presenta els següent suposits en la seua construcció:

- *Independència dels increments*: El model AG assumeix que els temps entre esdeveniments consecutius son independents. Açò significa que, el risc d'un nou esdeveniment no depen d'esdeveniments previs, almenys que s'incloguen covariables que capturen eixa dependència (per exemple, el número d'esdeveniments previs com covariable del tipus temps)
- *Risc basal comú*: Es usposa que tots els individus comparteixen la mateixa funció de risc basal  $\lambda_0(t)$ . Açò pot ser un limitació si el risc basal varia entre esdeveniments
- *Efecte proporcional de les covariables*: Les covariable tenen un efecte multiplicatiu constant sobre el risc, independentment del número d'esdeveniments previs

Per poder dur a terme el model AG, les dades tenen que estar en format de recompte, és a dir, que cada fila representa un interval de temps entre esdeveniments consecutius i esto ho podem fer amb la funció `getCountingProcess` de la següent manera:

```
# getCountingProcess: convertir les dades en format de recompte
library(tidyverse)
getCountingProcess <- function(x, colID, colTime) {
  id <- x[, colID]
  ids <- unique(id)
  tt <- x[,colTime]
```

```

out <- NULL
for (i in 1:length(ids)){
  tt.i <- tt[id%in%ids[i]]
  start <- c(0, tt.i[-length(tt.i)])
  out.i <- cbind(start, tt.i)
  out <- rbind(out, out.i)
}
ans <- data.frame(x[,colID,drop=TRUE],
                  x[,colTime],start=out[,1],stop=out[,2],
                  x %>% dplyr::select(!c(colID, colTime)))

names(ans)[1:2] <- names(x[,c(colID, colTime)])
return(ans)
}

```

Abans de res, les dades deuen ordenar-se per id i time per garantir un anàlisi adequat d'events recurrents:

```
lymphoma <- lymphoma[order(lymphoma$id, lymphoma$time), ]
```

Apliquem la funció `getCountingProcess` per obtenir el data en el format correcte per les meues dades:

```

# Aplicar a les nostres dades
lymphoma_ag <- getCountingProcess(lymphoma, colID = "id", colTime = "time")

```

```

## Warning: Using an external vector in selections was deprecated in tidysselect 1.1.0.
## i Please use 'all_of()' or 'any_of()' instead.
##   # Was:
##   data %>% select(colID)
##
##   # Now:
##   data %>% select(all_of(colID))
##
## See <https://tidysselect.r-lib.org/reference/faq-external-vector.html>.
## This warning is displayed once every 8 hours.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was
## generated.

```

```

## Warning: Using an external vector in selections was deprecated in tidysselect 1.1.0.
## i Please use 'all_of()' or 'any_of()' instead.
##   # Was:
##   data %>% select(colTime)
##
##   # Now:
##   data %>% select(all_of(colTime))
##
## See <https://tidysselect.r-lib.org/reference/faq-external-vector.html>.
## This warning is displayed once every 8 hours.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was
## generated.

```

```
head(lymphoma_ag)
```

```
##      id      time      start      stop event enum delay age sex distrib effage
## 1  6  3.900826  0.00000  3.900826    0    1    17  79  1      1    CR
## 2  7 63.173554  0.00000 63.173554    0    1    33  25  1      1    CR
## 3  8 41.289256  0.00000 41.289256    0    1    26  37  1      2    CR
## 6 11 17.950413  0.00000 17.950413    0    3    31  43  2      2    CR
## 5 11 20.826446 17.95041 20.826446    1    2    31  43  2      2    CR
## 4 11 29.421488 20.82645 29.421488    1    1    31  43  2      2    CR
```

El model AG per avaluar si el número de lesions al diagnòstic s'associa amb els temps de recaiguda del càncer després del primer tractament en pacients diagnosticats amb lymphoma de grau baix s'ajusta ampliant el model Cox implementat al **survival** package:

```
library(survival)
```

```
##
## Attaching package: 'survival'
```

```
## The following object is masked from 'package:boot':
##
##      aml
```

```
# Ajustar el modelo AG
```

```
ag_model <- coxph(Surv(start, stop, event) ~ as.factor(distrib) + cluster(id), data = lymphoma_ag)
summary(ag_model)
```

```
## Call:
## coxph(formula = Surv(start, stop, event) ~ as.factor(distrib),
##       data = lymphoma_ag, cluster = id)
##
##      n= 110, number of events= 49
##      (2 observations deleted due to missingness)
##
##              coef exp(coef) se(coef) robust se      z Pr(>|z|)
## as.factor(distrib)1 1.0304    2.8021   0.4987    0.4801 2.146  0.03187 *
## as.factor(distrib)2 1.4776    4.3826   0.5036    0.4775 3.094  0.00197 **
## as.factor(distrib)3 1.2377    3.4478   0.7400    0.6367 1.944  0.05190 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##              exp(coef) exp(-coef) lower .95 upper .95
## as.factor(distrib)1    2.802    0.3569    1.0935    7.18
## as.factor(distrib)2    4.383    0.2282    1.7189   11.17
## as.factor(distrib)3    3.448    0.2900    0.9899   12.01
##
## Concordance= 0.632 (se = 0.042 )
## Likelihood ratio test= 10.98 on 3 df,  p=0.01
## Wald test              = 9.76 on 3 df,  p=0.02
## Score (logrank) test = 10 on 3 df,  p=0.02,  Robust = 7.87 p=0.05
##
##      (Note: the likelihood ratio and score tests assume independence of
##      observations within a cluster, the Wald and robust score tests do not).
```

Anem a interpretar els resultats del model AG:

- Coeficients: els coeficients estimats  $\beta$  indiquen que l'efecte de cada categoria de **distrib** sobre el risc de recaiguda.
- Exp(coef): És la interpretació en termes de risc relatiu respecte la categoria basal. En el nostre cas, la categoria basal és quan la lesió durant el diagnòstic és única.

Amb els resultat del model, es pot veure que totes les categories de lesions durant el diagnostic son covariables significatives excepte la generalitzada.

A l'hora d'escollir un model per al temps de recaiguda del cancer, és important tenir en compte el procés de la malaltia. Si després de patir la primera recaiguda, el riscde la següent pot augmentar, això suggereix un model que incorpori una covariable dependent del temps per al nombre de recaigudes o utilitzar un model que contingui estrats separats per a cada esdeveniment com el model PWP. Tanmateix, si el risc de recaure es manté constant independentment del nombre de recaigudes anteriors, el model AG seria adequat. Per investigar la dependència dels esdeveniments, hem de crear una variable dependent del temps (enum) que codifiqui el nombre de recaigudes. Aquesta variable es pot crear mitjançant aquesta funció:

```
# getEnum: indica el número de recaigudes
getEnum <- function(x, colID) {
  id <- x[, colID]
  ids <- unique(id)
  out <- NULL
  for (i in 1:length(ids)) {
    tt.i <- sum(id %in% ids[i])
    out.i <- 1:tt.i
    out <- c(out, out.i)
  }
  out
}
```

Aleshores, incorporem aquesta variable dependent del temps al model de recaigudes per analitzar la dependència entre els temps de recurrència.

```
lymphoma_ag$enum <- getEnum(lymphoma_ag, colID=1)
head(lymphoma_ag)
```

##	id	time	start	stop	event	enum	delay	age	sex	distrib	effage
## 1	6	3.900826	0.00000	3.900826	0	1	17	79	1	1	CR
## 2	7	63.173554	0.00000	63.173554	0	1	33	25	1	1	CR
## 3	8	41.289256	0.00000	41.289256	0	1	26	37	1	2	CR
## 6	11	17.950413	0.00000	17.950413	0	1	31	43	2	2	CR
## 5	11	20.826446	17.95041	20.826446	1	2	31	43	2	2	CR
## 4	11	29.421488	20.82645	29.421488	1	3	31	43	2	2	CR

Fent el nou model incorporant la covariable **enum**:

```
ag.fit <- coxph(Surv(start, stop, event) ~ as.factor(distrib)+
               cluster(id), data=subset(lymphoma_ag, enum<4))
```

```
## Warning in Surv(start, stop, event): Stop time must be > start time, NA created
```

```
ag.fit
```

```
## Call:
## coxph(formula = Surv(start, stop, event) ~ as.factor(distrib),
##       data = subset(lymphoma_ag, enum < 4), cluster = id)
##
##               coef exp(coef) se(coef) robust se      z      p
## as.factor(distrib)1 0.9472    2.5784   0.5099   0.5002 1.894 0.05827
## as.factor(distrib)2 1.3789    3.9704   0.5105   0.4759 2.897 0.00376
## as.factor(distrib)3 1.4745    4.3689   0.7322   0.7090 2.080 0.03755
##
## Likelihood ratio test=9.68  on 3 df, p=0.02152
## n= 101, number of events= 43
## (2 observations deleted due to missingness)
```

```
ag.fit.dep <- coxph(Surv(start, stop, event) ~ as.factor(distrib) + cluster(id) + enum, data=subset(lymphoma_ag, enum < 4), cluster = id)
```

```
## Warning in Surv(start, stop, event): Stop time must be > start time, NA created
```

```
ag.fit.dep
```

```
## Call:
## coxph(formula = Surv(start, stop, event) ~ as.factor(distrib) +
##       enum, data = subset(lymphoma_ag, enum < 4), cluster = id)
##
##               coef exp(coef) se(coef) robust se      z      p
## as.factor(distrib)1 0.5944    1.8120   0.5260   0.4383 1.356 0.175
## as.factor(distrib)2 0.7600    2.1383   0.5450   0.4668 1.628 0.103
## as.factor(distrib)3 0.7721    2.1644   0.7597   0.5516 1.400 0.162
## enum                0.8676    2.3812   0.2053   0.2209 3.927 8.6e-05
##
## Likelihood ratio test=27.26  on 4 df, p=1.76e-05
## n= 101, number of events= 43
## (2 observations deleted due to missingness)
```

Incloure la variable dependent del temps que té en compte el nombre de recaigudes previs (enum) reduceix el valor de l'estadística  $-2\log L$  (per exemple, la desviació) que mesura la bondat d'ajust d'un model.

```
deviance.1 <- -2*summary(ag.fit)$loglik[2]
deviance.2 <- -2*summary(ag.fit.dep)$loglik[2]
pchisq(deviance.1 - deviance.2, df=1, lower=FALSE)
```

```
## [1] 2.747985e-05
```

```
qchisq(p=0.95, df=1, lower=FALSE)
```

```
## [1] 0.00393214
```

que es estadísticament significatiu (95%).

Per tant, el risc de recaiguda de càncer augmenta un 70%  $\exp(0.5337) = 1.70$  a mesura que augmenta el nombre de recaigudes previes. El model AG assumeix una funció de risc de referència comuna per a totes les recaigudes. Per tant, podem pensar que utilitzar un model estratificat (model PWP) per nombre de recaigudes podria ser més apropiat. Això estima el model de temps total PWP, ja que les variables d'aturada estan anotades en el temps de calendari.

**2. Model PWP-Total time** En l'apartat anterior amb el model AG assumim una funció de risc de referència per a totes les recaigudes. Si volem pensar en utilitzar un model estratificant pel nombre de recaigudes aleshores el PWP model és més apropiat, és a dir, estimant el model de PWP total time fins que l'stop variables.

D'aquesta forma, el nou model consistirà en la incorporació de la variable `enum` com a estrat dins del model AG per obtenir el model PWP-Total time:

```
pwp.fit.total <- coxph(Surv(start, stop, event) ~ as.factor(distrib) + cluster(id) + strata(enum), data =
```

```
## Warning in Surv(start, stop, event): Stop time must be > start time, NA created
```

```
pwp.fit.total
```

```
## Call:
## coxph(formula = Surv(start, stop, event) ~ as.factor(distrib) +
##       strata(enum), data = subset(lymphoma_ag, enum < 4), cluster = id)
##
##               coef exp(coef) se(coef) robust se      z      p
## as.factor(distrib)1 0.5890   1.8023   0.5367   0.3874 1.521 0.1284
## as.factor(distrib)2 0.7086   2.0312   0.5505   0.4286 1.653 0.0982
## as.factor(distrib)3 0.5210   1.6838   0.7845   0.4616 1.129 0.2590
##
## Likelihood ratio test=1.87 on 3 df, p=0.6009
## n= 101, number of events= 43
## (2 observations deleted due to missingness)
```

Podem observar del resultat, que cap dels coeficients del model és significatiu i que la incorporació per estrats disminueix el risc en totes tres categories respecte el model AG.

En canvi, encara que reduisca el risc, el que sí que es pot veure, és que es manté la relació de que la lesió en més d'un lloc nodal durant el diagnòstic continua matint un risc superior que en els altres casos.

**3. Model PWP-Gap time** El model PWP-Gap time és una extensió del model de Cox per a esdeveniments recurrents que es centra en el temps entre events consecutius (gap time). A diferència del model AG, que considera el temps total desde el inici de seguiment, el model de PWP-Gap time modela el temps entre la ocurrència d'un esdeveniment i el següent. Açò és útil quan el risc d'un nou esdeveniment depèn del temps transcurrit des de l'últim esdeveniment.

El model PWP-Gap time assumeix que els temps entre esdeveniments consecutius formen un procés de renovació. El risc per al  $k$ -èssim esdeveniment del individu  $i$  es modela com:

$$\lambda_k(t; Z_{ik}) = \lambda_{0k}(t)e^{\beta' Z_{it}(t)}$$

d'on:

- $\lambda_{0k}(t)$  és la funció de risc basal per al k-èssim esdeveniment.
- $Z_{ik}(t)$  és un vector de covariables per al individu i en el k-èssim esdeveniment.
- $\beta$  és el vector de paràmetres a estimar.

En aquest model, cada esdeveniment pot tindre la seua propia funció de risc basal ( $\lambda_{0k}(t)$ ), lo que permeteix capturar diferències en el risc entre esdeveniments successius. Endemés, dintre dels seus suposits trobem:

- *Procés de renovació*: els temps entre esdeveniments consecutius es suposen independents y la distribució del pròxim temps entre events depen només del últim esdeveniment.
- *Risc basal específic del esdeveniment*: cada esdeveniment pot tindre una funció de risc basal diferent.
- *Efecte proporcional de les covariables*: les covariables tenen un efecte multiplicatiu constant sobre el risc, independent del número d'esdeveniments previs.

En el model PWP-Gap time, les dades tenen que estructurar-se en el format de temps entre esdeveniments (gap time), és a dir, cada fila té que representar el temps transcurrit des de l'últim esdeveniment fins al següent. La següent funció ens ajudarà a fer-ho:

```
convert_gap_time <- function(df, colID, colStop, colStart ) {
  df_gap <- df
  df_gap <- df[order(df[[colID]], df[[colStop]]), ] # ordenar per individu i temps

  # Inicialitzar nova columna 'gap_time'
  df_gap$gap_stop <- NA

  ids <- unique(df[[colID]])
  for (i in ids) {
    idx <- which(df[[colID]] == i)
    n <- length(idx)

    # Calcular temps entre esdeveniments consecutius
    times <- df[[colStop]][idx]
    gaps <- c(times[1], diff(times))
    df_gap$gap_stop[idx] <- gaps
  }

  df_gap$start <- 0
  df_gap$stop <- df_gap$gap_stop
  df_gap$gap_stop <- NULL # opcional: eliminar columna auxiliar

  return(df_gap)
}
```

Aplicant la funció a les nostres dades per obtindre el format correcte:

```
lymphoma_pwp <- convert_gap_time(lymphoma_ag, colID = "id", colStop = "stop", colStart = "start")
head(lymphoma_pwp)
```

##	id	time	start	stop	event	enum	delay	age	sex	distrib	effage
## 1	6	3.900826	0	3.900826	0	1	17	79	1	1	CR
## 2	7	63.173554	0	63.173554	0	1	33	25	1	1	CR
## 3	8	41.289256	0	41.289256	0	1	26	37	1	2	CR



```
## 6 11 17.950413      0 17.950413      0 1 31 43 2 2 CR
## 5 11 20.826446      0 2.876033      1 2 31 43 2 2 CR
## 4 11 29.421488      0 8.595041      1 3 31 43 2 2 CR
```

Ara, realitzem el model PWP-Gap time amb les dades en el format adequat:

```
pwp.fit <- coxph(Surv(start, stop, event) ~ as.factor(distrib)+
                 cluster(id) + strata(enum), data=lymphoma_pwp)
pwp.fit
```

```
## Call:
## coxph(formula = Surv(start, stop, event) ~ as.factor(distrib) +
##       strata(enum), data = lymphoma_pwp, cluster = id)
##
##               coef exp(coef) se(coef) robust se      z      p
## as.factor(distrib)1 0.7856    2.1937   0.5161    0.3363 2.336 0.01949
## as.factor(distrib)2 1.0536    2.8680   0.5235    0.3416 3.085 0.00204
## as.factor(distrib)3 0.5410    1.7177   0.7741    0.4099 1.320 0.18690
##
## Likelihood ratio test=4.9 on 3 df, p=0.1789
## n= 110, number of events= 49
## (2 observations deleted due to missingness)
```

I amb el model PWP-Gap time també podem obtindre coeficients de regressió per estrat, de la següent manera:

```
pwp.fit.strata <- coxph(Surv(start, stop, event) ~ as.factor(distrib)* strata(enum) +
                       cluster(id), data=lymphoma_pwp)
pwp.fit.strata
```

```
## Call:
## coxph(formula = Surv(start, stop, event) ~ as.factor(distrib) +
##       strata(enum) + as.factor(distrib):strata(enum), data = lymphoma_pwp,
##       cluster = id)
##
##               coef exp(coef) se(coef)
## as.factor(distrib)1      1.074e+00 2.927e+00 8.168e-01
## as.factor(distrib)2      1.365e+00 3.916e+00 8.666e-01
## as.factor(distrib)3      1.616e+00 5.034e+00 1.233e+00
## as.factor(distrib)1:strata(enum)enum=2 -5.036e-01 6.043e-01 1.070e+00
## as.factor(distrib)2:strata(enum)enum=2 -7.492e-01 4.728e-01 1.111e+00
## as.factor(distrib)3:strata(enum)enum=2 -5.350e-01 5.857e-01 1.706e+00
## as.factor(distrib)1:strata(enum)enum=3 8.891e-01 2.433e+00 1.577e+00
## as.factor(distrib)2:strata(enum)enum=3 9.003e-01 2.460e+00 1.579e+00
## as.factor(distrib)3:strata(enum)enum=3      NA      NA 0.000e+00
## as.factor(distrib)1:strata(enum)enum=4 1.784e+01 5.605e+07 5.003e+03
## as.factor(distrib)2:strata(enum)enum=4 1.945e+01 2.809e+08 5.003e+03
## as.factor(distrib)3:strata(enum)enum=4      NA      NA 0.000e+00
##
##               robust se      z      p
## as.factor(distrib)1      7.872e-01 1.364 0.1725
## as.factor(distrib)2      8.264e-01 1.652 0.0986
## as.factor(distrib)3      1.283e+00 1.260 0.2078
```

```
## as.factor(distrib)1:strata(enum)enum=2 1.189e+00 -0.424 0.6718
## as.factor(distrib)2:strata(enum)enum=2 1.273e+00 -0.588 0.5562
## as.factor(distrib)3:strata(enum)enum=2 1.539e+00 -0.348 0.7282
## as.factor(distrib)1:strata(enum)enum=3 1.408e+00 0.631 0.5278
## as.factor(distrib)2:strata(enum)enum=3 1.390e+00 0.648 0.5172
## as.factor(distrib)3:strata(enum)enum=3 0.000e+00 NA NA
## as.factor(distrib)1:strata(enum)enum=4 2.080e+00 8.579 <2e-16
## as.factor(distrib)2:strata(enum)enum=4 2.211e+00 8.798 <2e-16
## as.factor(distrib)3:strata(enum)enum=4 0.000e+00 NA NA
##
## Likelihood ratio test=10.5 on 10 df, p=0.3981
## n= 110, number of events= 49
## (2 observations deleted due to missingness)
```

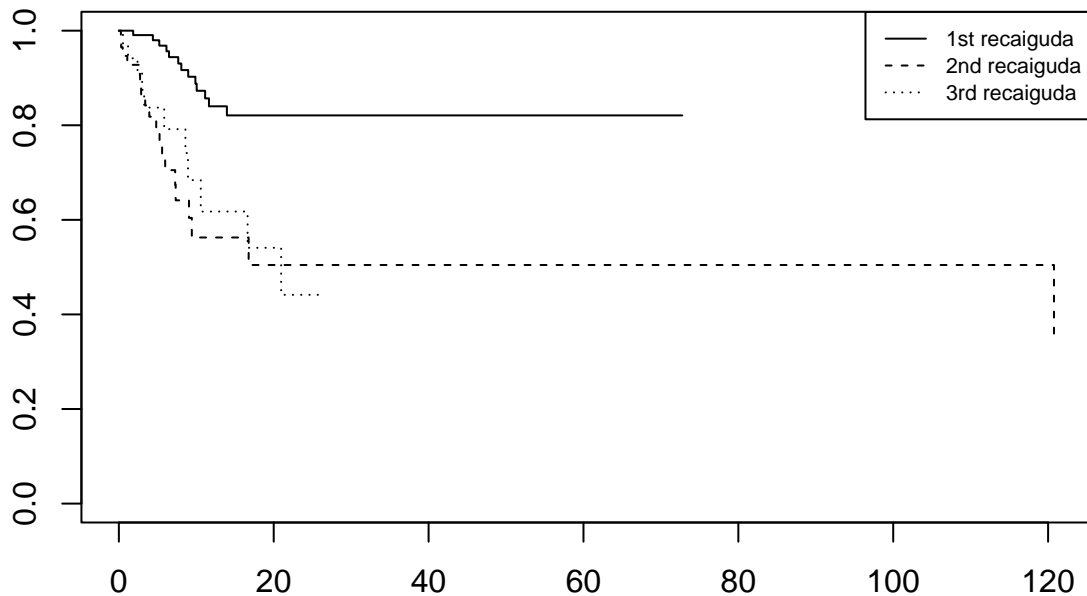
Cal tenir en compte que els coeficients d'algunes covariables no es poden estimar. Si el nombre de subjectes disminueix a mesura que augmenta  $k$  (esdeveniment), no es poden obtenir estimacions de coeficients estables per a aquests rangs més alts,  $k$ , de recaigudes. Veïem que, els únics valors on el model no té la capacitat d'obtenir els NA son en la categoria de lesió generalitzada quan el nombre de recaigudes és superior a 2. Per solventar aquesta problema podem limitar el nombre de recaigudes a 3 i que el model sí tinga la capacitat de ajustar correctament els coeficients, com es mostra:

```
pwp.fit.strata.3 <- coxph(Surv(start, stop, event) ~ as.factor(distrib) + strata(enum) +
  cluster(id) , data=subset(lymphoma_pwp, enum<3))
pwp.fit.strata.3
```

```
## Call:
## coxph(formula = Surv(start, stop, event) ~ as.factor(distrib) +
## strata(enum), data = subset(lymphoma_pwp, enum < 3), cluster = id)
##
##               coef exp(coef) se(coef) robust se      z      p
## as.factor(distrib)1 0.8033    2.2330   0.5273    0.3754 2.140 0.03235
## as.factor(distrib)2 0.9333    2.5430   0.5477    0.3492 2.673 0.00752
## as.factor(distrib)3 1.3328    3.7916   0.8482    0.6001 2.221 0.02636
##
## Likelihood ratio test=4.36 on 3 df, p=0.2255
## n= 87, number of events= 32
## (2 observations deleted due to missingness)
```

Les funcions de supervivència de referència es podem estimar i representar gràficament:

```
pwp.fit.strata.3 <- coxph(Surv(start, stop, event) ~ as.factor(distrib) + strata(enum) + cluster(id),da
plot(survfit(pwp.fit.strata.3),lty=c(1,2,3))
legend("topright", c("1st recaiguda", "2nd recaiguda", "3rd recaiguda"), lty=c(1,2,3), cex = 0.7)
```

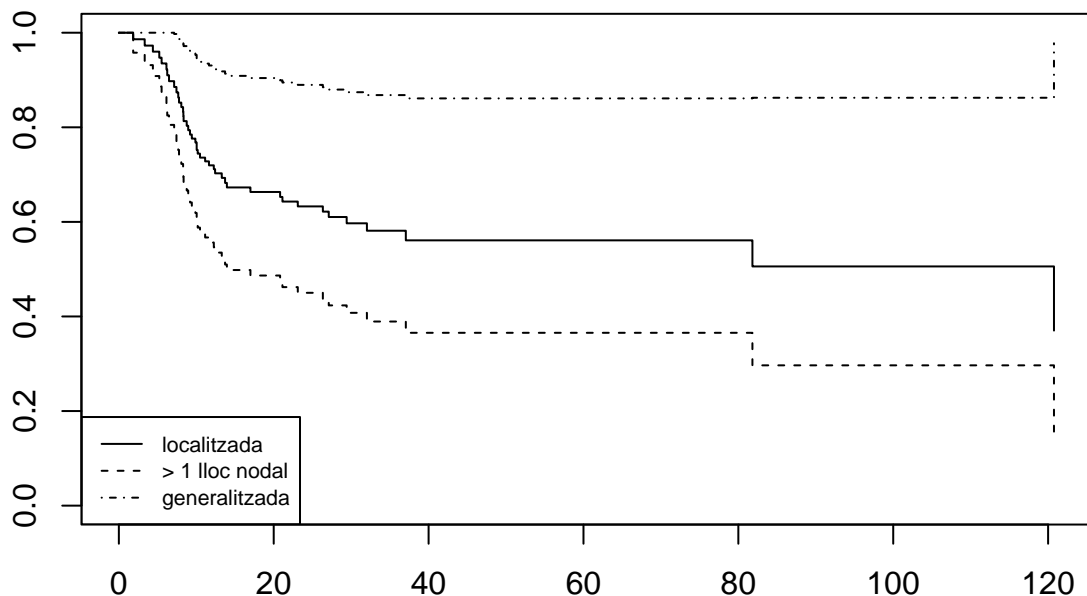


gràficament, es pot observar les mateixes conclusions que hem deduït en el model AG i la necessitat d'estratificar pel nombre de recaigudes previes. Ací, s'observa que les estimacions de les funcions de supervivència disminueixen significativament després de la primera recaiguda al llarg del temps. Veiem que aquells individus que presenten més d'una recaiguda la seua supervivència es prou inferior a aquells que només experimenten una.

Podem veure com afecta al nombre de llessions del model AG:

```
ag.fit <- coxph(Surv(start, stop, event) ~ as.factor(distrib) +
               cluster(id), data=subset(lymphoma_ag, enum<4))

plot(survfit(ag.fit.dep), lty=c(1,2,4))
legend("bottomleft", c("localitzada", "> 1 lloc nodal", "generalitzada"), lty=c(1,2,4), cex = 0.7)
```



Veiem que, quan els individus presenten llessions en més d'un lloc nodal, la seua funció de supervivència decau més rapidament que en els altres condicions. Endemés, si presenta una llessió generalitzada, la seua

funció de supervivència és la més elevada, inclús al llarg del temps experimenta un petit increment. Encara que en es altre casos açò no passa.

**3. Model WLW (Wei, Lin y Weisfeld)** El model WLW és una extensió del model de COx per a esdeveniments recurrents que permeteix utilitzar dades amb múltiples temps de fallada incomplets. A diferència dels models AG i PWP, que es centren en el procés de recompte o el temps entre esdeveniments, el model WLW modela la distribució marginal de cada esdeveniment recurrent, permetint que cada esdeveniment tinga la seua propia funció de risc basal.

El model assumeix que la funció de risc per a k-èssim esdeveniment del individu i és:

$$\lambda_k(t; Z_{ik}) = \lambda_{0k}(t)e^{\beta' Z_{it}(t)}$$

d'on:

- $\lambda_{0k}(t)$  és la funció de risc basal per al k-èssim esdeveniment.
- $Z_{ik}(t)$  és un vector de covariables per al individu i en el k-èssim esdeveniment.
- $\beta$  és el vector de paràmetres a estimar.

En aquest model, cada esdeveniment pot tindre la seua propia funció de risc basal ( $\lambda_{0k}(t)$ ), lo que permeteix capturar diferències en el risc entre esdeveniments successius. Endemés, dintre dels seus suposits trobem:

- *Independència condicional*: els temps entre esdeveniments recurrents son independents condicionalment a les covariables
- *Risc basal específic del esdeveniment*: cada esdeveniment pot tindre una funció de risc basal diferent.
- *Efecte proporcional de les covariables*: les covariables tenen un efecte multiplicatiu constant sobre el risc, independent del número d'esdeveniments previs.

En el model WLW, les dades tenen que estructurar-se en el format de temps total des de l'inici, és a dir, cada fila té que representar el temps transcurrit des de l'últim esdeveniment fins al següent i que cada pacient tinga el mateix nombre d'entrades. Per exemple podem fixar el nombre de recaigudes a 3, però els casos que presenten menys, tenim que duplicar-les. La següent funció ens dona el data ven estructurat

```
conver_total_time <- function(df, colID, lim=3){
  df.total <- NULL
  ids <- unique(df[[colID]])
  for (i in ids){
    aux <- sum(df[colID]==i)
    mat <- df[df[,colID]==i, ]

    if(aux < lim){
      df.total <- rbind(df.total, mat)
      ultima <- mat[aux,]
      while(aux < lim){
        df.total <- rbind(df.total, ultima)
        aux <- aux +1
      }
    } else{
      df.total <- rbind(df.total, mat)
    }
  }
}
```

```

    return(df.total)
}

```

Aplicant aquesta funció a la nostra data set anterior:

```

lymphoma_wlw <- conver_total_time(lymphoma_ag, colID = "id")
head(lymphoma_wlw)

```

```

##      id      time start      stop event enum delay age sex distrib effage
## 1    6  3.900826      0  3.900826      0      1    17  79   1        1    CR
## 11   6  3.900826      0  3.900826      0      1    17  79   1        1    CR
## 12   6  3.900826      0  3.900826      0      1    17  79   1        1    CR
## 2    7 63.173554      0 63.173554      0      1    33  25   1        1    CR
## 21   7 63.173554      0 63.173554      0      1    33  25   1        1    CR
## 22   7 63.173554      0 63.173554      0      1    33  25   1        1    CR

```

Un cop tenim les dades en el format adequat, li passem les dades al model:

```

wlw.fit.total <- coxph(Surv(time, event) ~ as.factor(distrib) + strata(enum) +
                      cluster(id) , data=subset(lymphoma_wlw, enum<4))
wlw.fit.total

```

```

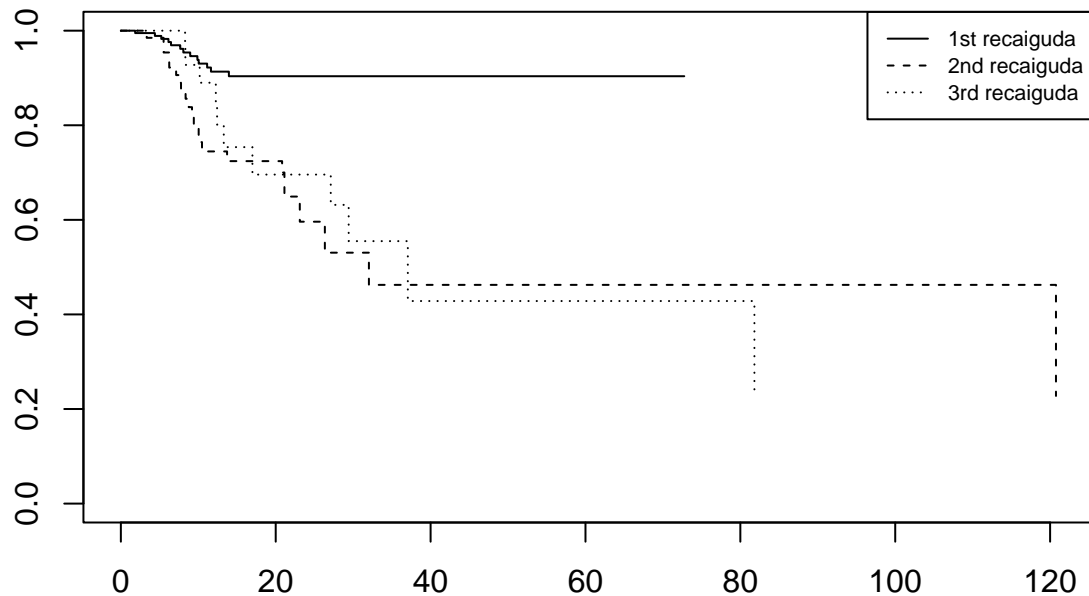
## Call:
## coxph(formula = Surv(time, event) ~ as.factor(distrib) + strata(enum),
##       data = subset(lymphoma_wlw, enum < 4), cluster = id)
##
##              coef exp(coef) se(coef) robust se      z      p
## as.factor(distrib)1 0.5564    1.7443   0.4378    0.4332 1.284 0.1991
## as.factor(distrib)2 0.7496    2.1162   0.4366    0.4616 1.624 0.1043
## as.factor(distrib)3 0.9701    2.6382   0.7023    0.5445 1.782 0.0748
##
## Likelihood ratio test=3.68 on 3 df, p=0.2976
## n= 189, number of events= 51

```

```

plot(survfit(wlw.fit.total), , lty=c(1,2,3))
legend("topright", c("1st recaiguda", "2nd recaiguda", "3rd recaiguda"), lty=c(1,2,3), cex = 0.7)

```



- ¿Obtenemos la misma conclusión utilizando los tres modelos? (NOTA: utilice algunas de las funciones que hemos visto en las clases para preparar los datos necesarios).

Per veure si els tres models presenten resultat semblant, que com podem esperar, el model AG serà diferent dels altres dos ja que aquests últims consideren que els increments no son independents. Aleshora, tenim que observar els coeficients dels models:

```
##      distrib1 distrib2 distrib3
## AG  0.9471567 1.3788592 1.4745074
## PWP 0.7855832 1.0536113 0.5409816
## WLW 0.5563699 0.7496388 0.9700970
```

- Repita los mismos análisis ajustando por sexo y respuesta al tratamiento (variable tt effage). ¿Obtenemos la misma conclusión que en los modelos sin dicho ajuste?

Ajustem el model AG per sexe i resposta al tractament

```
ag.fit.adj <- coxph(Surv(start, stop, event) ~ as.factor(distrib) +
                    cluster(id) + as.factor(sex) + as.factor(effage), data=lymphoma_ag)
```

```
## Warning in Surv(start, stop, event): Stop time must be > start time, NA created
```

```
ag.fit.adj
```

```
## Call:
## coxph(formula = Surv(start, stop, event) ~ as.factor(distrib) +
##       as.factor(sex) + as.factor(effage), data = lymphoma_ag, cluster = id)
##
##              coef exp(coef) se(coef) robust se      z      p
## as.factor(distrib)1 1.5476   4.7002   0.5615   0.5577 2.775 0.005521
## as.factor(distrib)2 1.9322   6.9049   0.5684   0.5412 3.570 0.000356
## as.factor(distrib)3 1.7824   5.9443   0.8121   0.7884 2.261 0.023768
```

```
## as.factor(sex)2      0.6861    1.9859    0.3391    0.3489 1.966 0.049253
## as.factor(effage)PR 0.1912    1.2107    0.5122    0.5765 0.332 0.740147
## as.factor(effage)SD 1.6751    5.3395    1.0906    1.2208 1.372 0.170029
##
## Likelihood ratio test=17.03 on 6 df, p=0.009187
## n= 110, number of events= 49
## (2 observations deleted due to missingness)
```

```
pwp.fit.total.adj <- coxph(Surv(start, stop, event) ~ as.factor(distrib) + cluster(id) + strata(enum) +
```

```
## Warning in Surv(start, stop, event): Stop time must be > start time, NA created
```

```
pwp.fit.total.adj
```

```
## Call:
## coxph(formula = Surv(start, stop, event) ~ as.factor(distrib) +
##       strata(enum) + as.factor(sex) + as.factor(effage), data = subset(lymphoma_ag,
##       enum < 4), cluster = id)
##
##               coef exp(coef) se(coef) robust se      z      p
## as.factor(distrib)1 1.0962060 2.9927897 0.6146930 0.4921437 2.227 0.0259
## as.factor(distrib)2 1.2301217 3.4216458 0.6299074 0.4870274 2.526 0.0115
## as.factor(distrib)3 1.1911575 3.2908881 0.8730001 0.5998920 1.986 0.0471
## as.factor(sex)2      0.6219273 1.8625142 0.3895888 0.2814146 2.210 0.0271
## as.factor(effage)PR -0.0001036 0.9998964 0.5949579 0.5701175 0.000 0.9999
## as.factor(effage)SD 1.9308135 6.8951170 1.1228109 1.1719176 1.648 0.0994
##
## Likelihood ratio test=6.37 on 6 df, p=0.3832
## n= 101, number of events= 43
## (2 observations deleted due to missingness)
```

Ara fem el mateix per al model PWP-Gap time:

```
pwp.fit.adj <- coxph(Surv(start, stop, event) ~ as.factor(distrib)+
                    cluster(id) + strata(enum) + as.factor(sex) + as.factor(effage), data=subset(lymphoma,
```

```
## Warning in Surv(start, stop, event): Stop time must be > start time, NA created
```

```
pwp.fit.adj
```

```
## Call:
## coxph(formula = Surv(start, stop, event) ~ as.factor(distrib) +
##       strata(enum) + as.factor(sex) + as.factor(effage), data = subset(lymphoma_pwp,
##       enum < 4), cluster = id)
##
##               coef exp(coef) se(coef) robust se      z      p
## as.factor(distrib)1 1.2229    3.3972    0.5886    0.4033 3.032 0.002426
## as.factor(distrib)2 1.3379    3.8110    0.5991    0.3988 3.355 0.000794
## as.factor(distrib)3 1.5677    4.7958    0.8543    0.5732 2.735 0.006233
## as.factor(sex)2      0.4673    1.5957    0.3699    0.2520 1.854 0.063695
## as.factor(effage)PR 0.4835    1.6217    0.5552    0.5891 0.821 0.411857
```

```
## as.factor(effage)SD 2.1222    8.3491    1.1235    1.0903 1.946 0.051598
##
## Likelihood ratio test=8.54 on 6 df, p=0.2012
## n= 101, number of events= 43
## (2 observations deleted due to missingness)
```

i pel model WLW:

```
wlw.fit.total.adj <- coxph(Surv(time, event) ~ as.factor(distrib) + strata(enum) +
  cluster(id) + as.factor(sex) + as.factor(effage), data=subset(lymphoma_wlw, enum<4))
wlw.fit.total.adj
```

```
## Call:
## coxph(formula = Surv(time, event) ~ as.factor(distrib) + strata(enum) +
##       as.factor(sex) + as.factor(effage), data = subset(lymphoma_wlw,
##       enum < 4), cluster = id)
##
##               coef exp(coef) se(coef) robust se      z      p
## as.factor(distrib)1  1.1442   3.1398   0.5330   0.6565  1.743 0.0814
## as.factor(distrib)2  1.2130   3.3636   0.5072   0.5915  2.051 0.0403
## as.factor(distrib)3  1.7193   5.5807   0.7849   0.8253  2.083 0.0372
## as.factor(sex)2      0.6673   1.9489   0.3723   0.4667  1.430 0.1528
## as.factor(effage)PR  0.6060   1.8331   0.5437   0.5242  1.156 0.2477
## as.factor(effage)SD  2.7671  15.9119   1.1039   1.1897  2.326 0.0200
##
## Likelihood ratio test=11.12 on 6 df, p=0.08479
## n= 189, number of events= 51
```

Ara de la mateixa forma que abans recollim tota la informació en la següent taula:

	distrib1	distrib2	distrib3	sex2	effagePR	effageSD
## AG	1.547615	1.932237	1.782433	0.6860770	0.1911997164	1.675123
## PWP-Total	1.096206	1.230122	1.191157	0.6219273	-0.0001036279	1.930813
## PWP-Gap	1.222941	1.337894	1.567747	0.4673196	0.4834619590	2.122156
## WLW	1.144171	1.213006	1.719320	0.6672706	0.6060317095	2.767069

Si ajustem per sexe i per el diagnostic, el resultats no son igual que abans, però les relacions sí, en el model AG ajustat es continua mantenint la relacions entre les categories de llessions durant el diagnostic ja que la categoria que presenta més risc respecte les altres és la associada a lesió en més d'un lloc nodal respecte les altres. En canvi, en els altres model, el comportament de que la lesió generalitzada és la que presenta major risc. Per que respecta a les variables ajustades, el sex 2 té un major risc respecte el sexe 1 i el mateix passa pels diagnostics PR i SD respecte CD.

Endemés, en el model PWP-Total time el risc de tindre un diagnostic "PR" disminueix el risc de patir recaiguda respecte la categoria basal.

### Conclusió *Sense ajust per covariables*

Els resultats dels models AG, PWP-Gap time, PWP-Total time i WLW mostren diferències en els coeficients de la variable `distrib`. Això indica que el nombre de llessions en el diagnòstic té un efecte significatiu en el temps fins a la recaiguda, però la magnitud d'aquest efecte varia segons el model utilitzat.

*Amb ajust per covariables (sexe i resposta al tractament)*



Després d'ajustar per sexe i resposta al tractament, els coeficients de la variable **distrib** continuen sent significatius, però els seus valors canvien lleugerament. Això suggereix que les covariables addicionals expliquen part de la variabilitat en el temps fins a la recaiguda, però el nombre de lesions continua sent un factor important.

#### *Comparació de models*

- Els models AG i PWP-Total time produeixen resultats similars, cosa que indica que l'efecte del nombre de lesions és consistent quan es considera el temps total des de l'inici del seguiment.
- El model PWP-Gap time mostra diferències més marcades, la qual cosa podria reflectir que el temps entre recaigudes consecutives està influït per altres factors no capturats en el model.
- El model WLW també mostra resultats similars als models AG i PWP-Total time, cosa que suggereix que l'efecte del nombre de lesions és robust a diferents formulacions temporals.

En conclusió, tots els models coincideixen que el nombre de lesions en el diagnòstic és un predictor significatiu del temps fins a la recaiguda en pacients amb limfoma de grau baix. Tanmateix, la magnitud exacta d'aquest efecte pot variar segons el model i les covariables incloses.