Módulo 4 Aplicaciones de R en epidemiología

Selección de subconjuntos de regresión

En este apartado proponemos algunos paquetes para selección de subconjuntos de regresión. El objetivo de estos procedimientos es el de elegir entre un subconjunto de potenciales variables predictoras para proponer diferentes modelos entre los cuales elegir conforme a un criterio concreto, por ejemplo mayor R2 ajustado.

Paquete "leaps"

Este paque sólo sirve para el ajuste de modelos lineales. Para ilustrar su funcionamiento utilizaremos la base de datos "lowbwt.csv". Baja los datos del repositorio y analiza su contenido.

```
lbw <- read.csv("lowbwt.csv")
str(lbw)
```

```
## 'data.frame':    189 obs. of  11 variables:
##  $ id   : int  4 10 11 13 15 16 17 18 19 20 ...
##  $ low  : chr  "< 2500 g" "< 2500 g" "< 2500 g" "< 2500 g" ...
##  $ age  : int  28 29 34 25 25 27 23 24 24 21 ...
##  $ lwt  : int  120 130 187 105 85 150 97 128 132 165 ...
##  $ race : chr  "Other" "White" "Black" "Other" ...
##  $ smoke: chr  "Yes" "No" "Yes" "No" ...
##  $ ptl  : int  1 0 0 1 0 0 0 1 0 0 ...
##  $ ht   : chr  "No" "No" "Yes" "Yes" ...
##  $ ui   : chr  "Yes" "Yes" "No" "No" ...
##  $ ftv  : int  0 2 0 0 0 0 1 1 0 1 ...
##  $ bwt  : int  709 1021 1135 1330 1474 1588 1588 1701 1729 1790 ...
```

cargamos el paquete

```
library(leaps)
```

A continuación creo un objeto "Mod1" donde indico la variable que quiero explicar "bwt" y el conjunto de variables predictoras candidatas, tipificadas según el tipo de variable (discreta/factor o continua).

```
Mod1 <- regsubsets(bwt ~ age + lwt + race +
                     smoke + as.factor(ptl) + ht + ui + ftv,
                   data = lbw,
                   nbest = 1,       # muestra el mejor modelo en función del nº de predictores
                   nvmax = NULL,    # no limitamos el nº de variables
                   force.in = NULL, force.out = NULL,
                   method = "exhaustive")
```

"summary" nos facilita el mejor modelo de acuerdo con el número de predictores incorporados

```
res.sum <- summary(Mod1)
res.sum
```

```
## Subset selection object
## Call: regsubsets.formula(bwt ~ age + lwt + race + smoke + as.factor(ptl) +
##     ht + ui + ftv, data = lbw, nbest = 1, nvmax = NULL, force.in = NULL,
##     force.out = NULL, method = "exhaustive")
## 11 Variables  (and intercept)
##                  Forced in Forced out
## age                  FALSE      FALSE
## lwt                  FALSE      FALSE
## raceOther            FALSE      FALSE
## raceWhite            FALSE      FALSE
## smokeYes             FALSE      FALSE
## as.factor(ptl)1      FALSE      FALSE
## as.factor(ptl)2      FALSE      FALSE
## as.factor(ptl)3      FALSE      FALSE
## htYes                FALSE      FALSE
## uiYes                FALSE      FALSE
## ftv                  FALSE      FALSE
## 1 subsets of each size up to 11
## Selection Algorithm: exhaustive
##           age lwt raceOther raceWhite smokeYes as.factor(ptl)1 as.factor(ptl)2
## 1  ( 1 )  " " " " " "       " "       " "       " "              " "
## 2  ( 1 )  " " " " " "       "*"       " "       " "              " "
## 3  ( 1 )  " " " " " "       "*"       "*"       " "              " "
## 4  ( 1 )  " " " " " "       "*"       "*"       " "              " "
## 5  ( 1 )  " " "*" " "       "*"       "*"       " "              " "
## 6  ( 1 )  " " "*" " "       "*"       "*"       "*"              " "
## 7  ( 1 )  " " "*" " "       "*"       "*"       "*"              " "
## 8  ( 1 )  " " "*" "*"       "*"       "*"       "*"              " "
## 9  ( 1 )  "*" "*" "*"       "*"       "*"       "*"              " "
## 10  ( 1 ) "*" "*" "*"       "*"       "*"       "*"              " "
## 11  ( 1 ) "*" "*" "*"       "*"       "*"       "*"              "*"
##           as.factor(ptl)3 htYes uiYes ftv
## 1  ( 1 )  " "             " "   "*"   " "
## 2  ( 1 )  " "             " "   "*"   " "
## 3  ( 1 )  " "             " "   "*"   " "
## 4  ( 1 )  " "             "*"   "*"   " "
## 5  ( 1 )  " "             "*"   "*"   " "
## 6  ( 1 )  " "             "*"   "*"   " "
## 7  ( 1 )  "*"             "*"   "*"   " "
## 8  ( 1 )  "*"             "*"   "*"   " "
## 9  ( 1 )  "*"             "*"   "*"   " "
## 10  ( 1 ) "*"             "*"   "*"   "*"
## 11  ( 1 ) "*"             "*"   "*"   "*"
```

elijo entre los modelos anteriores en función de cuatro criterios (R2 ajustado, estadístico cp de Mallow, criterio de información bayesiano BIC y suma del cuadrado de los residuos).

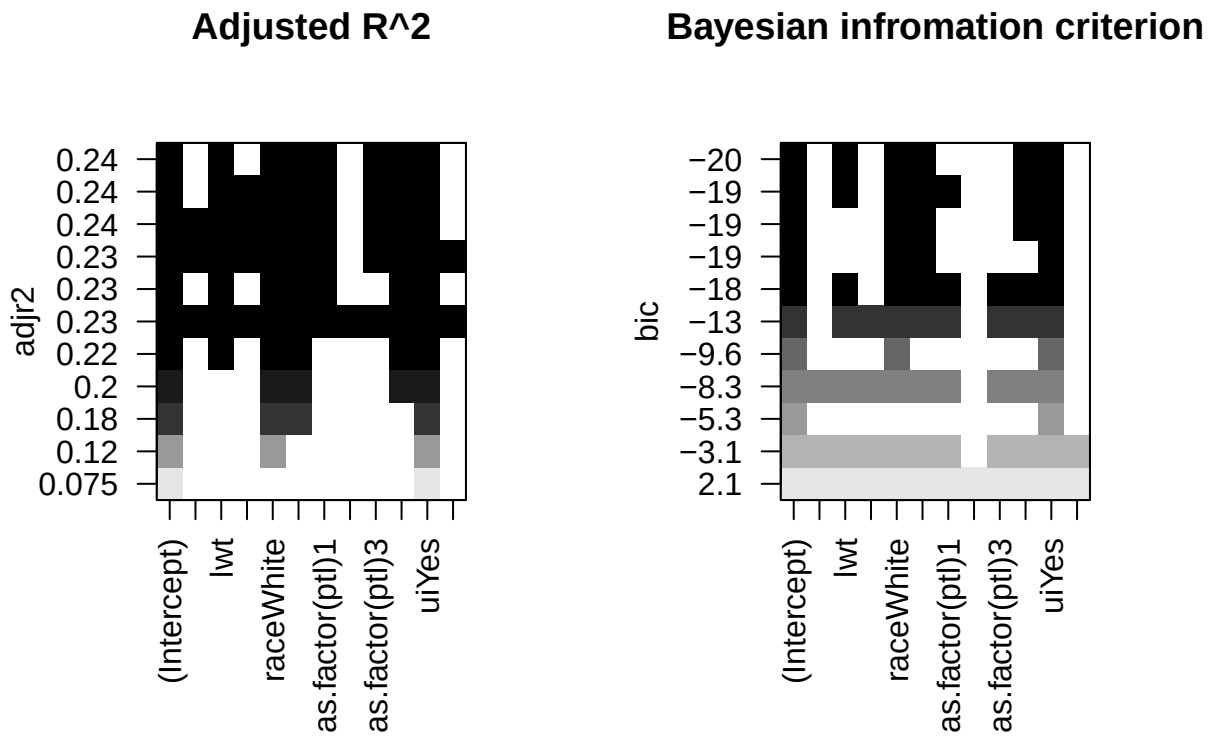
```
data.frame(
  Adj.R2 = which.max(res.sum$adjr2),
  cp.Mallow = which.min(res.sum$cp),
```

```
  BIC = which.min(res.sum$bic),
  rss = which.min(res.sum$rss)
)
```

```
##   Adj.R2 cp.Mallow BIC rss
## 1      7         7   5  11
```

El paquete permite presentar los resultados con gráficos.

```
par(mfrow=c(1,2))
plot(Mod1, scale = "adjr2", main = "Adjusted R^2")
plot(Mod1, scale = "bic", main = "Bayesian infromation criterion")
```




```
par(mfrow=c(1,1))
```

En este ejemplo voy a decidir en función del bic, por lo que mi mejor modelo tiene 6 variables. Sus coeficientes son:

```
coef(Mod1, 6)
```

```
##     (Intercept)             lwt       raceWhite        smokeYes as.factor(ptl)1
##     2554.410935        3.712452      367.199032     -331.193656     -309.414098
##           htYes           uiYes
##     -565.805402     -494.852077
```

Lo compruebo estimando dicho modelo. Para facilitar el código identifico que las variables race y ptl son factores.

```
lbw$race <- as.factor(lbw$race)
lbw$ptl <- as.factor(lbw$ptl)

best.model <- lm(bwt ~ lwt + I(race=="White") + smoke + I(ptl==1) + ht + ui, data = lbw)
summary(best.model)
```

```
##
## Call:
## lm(formula = bwt ~ lwt + I(race == "White") + smoke + I(ptl ==
##     1) + ht + ui, data = lbw)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1888.4  -427.4    37.0   485.4  1611.8
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)              2554.411    218.374  11.697  < 2e-16 ***
## lwt                         3.712      1.593   2.331 0.020874 *
## I(race == "White")TRUE    367.199     99.337   3.697 0.000289 ***
## smokeYes                 -331.194    102.338  -3.236 0.001439 **
## I(ptl == 1)TRUE          -309.414    143.123  -2.162 0.031933 *
## htYes                    -565.805    197.625  -2.863 0.004689 **
## uiYes                    -494.852    133.707  -3.701 0.000284 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 638.8 on 182 degrees of freedom
## Multiple R-squared:  0.2567, Adjusted R-squared:  0.2322
## F-statistic: 10.47 on 6 and 182 DF,  p-value: 5.68e-10
```

Paquete "bestglm"

```
library(bestglm)
```

Nuevo paquete para selección de subconjuntos de regresión que permite la selección en modelos no-lineales, logit, probit, etc. El paquete me obliga a seleccionar previmente las variables candidatas y a continuación clasificarlas como numéricas o factores.

```
lbw$age <- as.numeric(lbw$age)
lbw$lwt <- as.numeric(lbw$lwt)
lbw$ftv <- as.numeric(lbw$ftv)
lbw$bwt <- as.numeric(lbw$bwt)
lbw$race <- as.factor(lbw$race)
lbw$ptl <- as.factor(lbw$ptl)
lbw$smoke <- as.factor(lbw$smoke)
lbw$ht <- as.factor(lbw$ht)
lbw$ui <- as.factor(lbw$ui)
```

```
lbw.for.bestglm <- within(lbw, {
  id   <- NULL        # Delete
  low  <- NULL
  y    <- bwt         # bwt into y
  bwt  <- NULL        # Delete bwt
})

## Reordeno las variables
lbw.for.bestglm <-
  lbw.for.bestglm[, c("age","lwt","race","smoke","ptl","ht","ui","ftv","y")]
```

Busco el mejor modelo (criterio BIC), muestro los 5 mejores y presento el resultado final del modelo elegido.

```
res.bestglm <-
  bestglm(Xy = lbw.for.bestglm,
          family = gaussian,
          IC = "BIC",                 # Information criteria for
          method = "exhaustive")

## Show top 5 models
res.bestglm$BestModels
```

```
##     age   lwt race smoke   ptl    ht   ui   ftv Criterion
## 1 FALSE  TRUE TRUE  TRUE FALSE  TRUE TRUE FALSE  2470.181
## 2 FALSE FALSE TRUE  TRUE FALSE  TRUE TRUE FALSE  2471.476
## 3 FALSE FALSE TRUE  TRUE FALSE FALSE TRUE FALSE  2472.055
## 4 FALSE  TRUE TRUE  TRUE FALSE FALSE TRUE FALSE  2473.658
## 5  TRUE  TRUE TRUE  TRUE FALSE  TRUE TRUE FALSE  2475.151
```

```
summary(res.bestglm$BestModel)
```

```
##
## Call:
## lm(formula = y ~ ., data = data.frame(Xy[, c(bestset[-1], FALSE),
##     drop = FALSE], y = y))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1843.33  -432.71    66.99   460.95  1630.91
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 2361.830    281.573   8.388 1.34e-14 ***
## lwt            4.239      1.675   2.531 0.012225 *
## raceOther    125.810    157.567   0.798 0.425646
## raceWhite    475.808    145.578   3.268 0.001293 **
## smokeYes    -354.900    103.426  -3.431 0.000743 ***
## htYes       -585.112    199.610  -2.931 0.003809 **
## uiYes       -524.439    134.652  -3.895 0.000138 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 645.8 on 182 degrees of freedom
## Multiple R-squared:  0.2403, Adjusted R-squared:  0.2152
## F-statistic: 9.592 on 6 and 182 DF,  p-value: 3.659e-09
```

Veamos cómo funciona el paquete sobre modelos logit. Necesitamos reordenar de nuevo las variables candidatas y especificar la nuava variable a explicar "loW".

```
lbw$low <- as.factor(lbw$low)

## Prepare data
lbw.for.best.logistic <- within(lbw, {
  id   <- NULL
  bwt  <- NULL
  y    <- low
  low  <- NULL
})

## Reorder variables
lbw.for.best.logistic <-
  lbw.for.best.logistic[, c("age","lwt","race","smoke","ptl","ht","ui","ftv","y")]

## Perform
res.best.logistic <-
  bestglm(Xy = lbw.for.best.logistic,
          family = binomial,          # binomial family for logistic
          IC = "BIC",
          method = "exhaustive")

res.best.logistic$BestModels
```

```
##     age  lwt  race smoke   ptl   ht    ui   ftv Criterion
## 1 FALSE TRUE FALSE FALSE FALSE TRUE FALSE FALSE  231.6256
## 2 FALSE TRUE FALSE FALSE FALSE TRUE  TRUE FALSE  232.3381
## 3 FALSE TRUE FALSE FALSE  TRUE TRUE FALSE FALSE  232.4507
## 4 FALSE TRUE FALSE  TRUE FALSE TRUE FALSE FALSE  232.5830
## 5 FALSE TRUE FALSE  TRUE FALSE TRUE  TRUE FALSE  233.7927
```

```
summary(res.best.logistic$BestModel)
```

```
##
## Call:
## glm(formula = y ~ ., family = family, data = Xi, weights = weights)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1805  -1.2920   0.7383   0.8727   1.8598
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.450679   0.820965  -1.767  0.07722 .
## lwt          0.018653   0.006594   2.829  0.00467 **
## htYes       -1.855511   0.700976  -2.647  0.00812 **
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 234.67  on 188  degrees of freedom
## Residual deviance: 221.14  on 186  degrees of freedom
## AIC: 227.14
##
## Number of Fisher Scoring iterations: 4
```

Análisis de supervivencia

El análisis de supervivencia es un área estadística en la que la variable respuesta es el tiempo que transcurre entre un evento inicial (que determina la inclusión del individuo en el estudio) y un evento final (genéricamente llamado fallo o muerte) que ocurre cuando el individuo sale del estudio (muerte, alta de la enfermedad, etc.). Son estudios de cohorte donde puede ocurrir que algún individuo lo abandone antes de tiempo, registrándose sólo información parcial (censura) sobre la variable de interés (tiempo hasta el fallo). Uno de los objetivos del análisis de supervivencia es incorporar la información parcial que proporcionan los individuos censurados.

La importancia de esta técnica estadística se ve reflejada en los númerosos paquetes de R con funciones para el análisis de supervivencia. En este curso presentaresmo sólo unos pocos paquetes: survival, y ggfortify.

- El paquete survival

El paquete "survival" es la piedra angular del análisis de supervivencia de R. El paquete en sí no solo es rico en características, sino que el objeto creado por la función "Surv(...)", que contiene el tiempo hasta el fallo y la información de censura, es la estructura básica de datos de análisis de supervivencia en R.

A modo de complemento, el paquete "ggfortify" nos proporciona la salida gráfica del análisis de supervivencia bajo el marco gráfico de "ggplot2".

Como siempre ilustraremos el manejo de los paquetes anteriores con un ejemplo númerico.

A continuación cargamos los paquetes (previamente tendrás que instalarlos) y los daots del ejemplo, fichero "veteran.csv" del repositorio GitHub del curso.

```
library(survival)
library(ggplot2)
library(ggfortify)

veteran <- read.csv("veteran.csv", sep=";")
head(veteran)
```

```
##   trt celltype time status karno diagtime age prior
## 1   1 squamous   72      1    60        7  69     0
## 2   1 squamous  411      1    70        5  64    10
## 3   1 squamous  228      1    60        3  38     0
## 4   1 squamous  126      1    60        9  63    10
## 5   1 squamous  118      1    70       11  65    10
## 6   1 squamous   10      1    20        5  49     0
```

Análisis no paramétrico Kaplan-Meier

Creamos el objeto km donde alojamos l ainformación de la función de supervivencia:

```
km <- with(veteran, Surv(time, status))
```

A continuación, utilizamod la función survfit(...) para producir las estimaciones de Kaplan-Meier de la probabilidad de supervivencia en el tiempo. Podemos particularizar los cortes en la variable tiempo, por ejmeplo,estimamos los estadísticos para 1, 30, 60 y 90 días, y a partir de entonces cada 90 días.

```
km_fit <- survfit(km ~ 1, data=veteran)
km_fit
```

```
## Call: survfit(formula = km ~ 1, data = veteran)
##
##        n events median 0.95LCL 0.95UCL
## [1,] 137    128     80      52     105
```

```
summary(km_fit, times = 365.25) # supervivencia al año
```

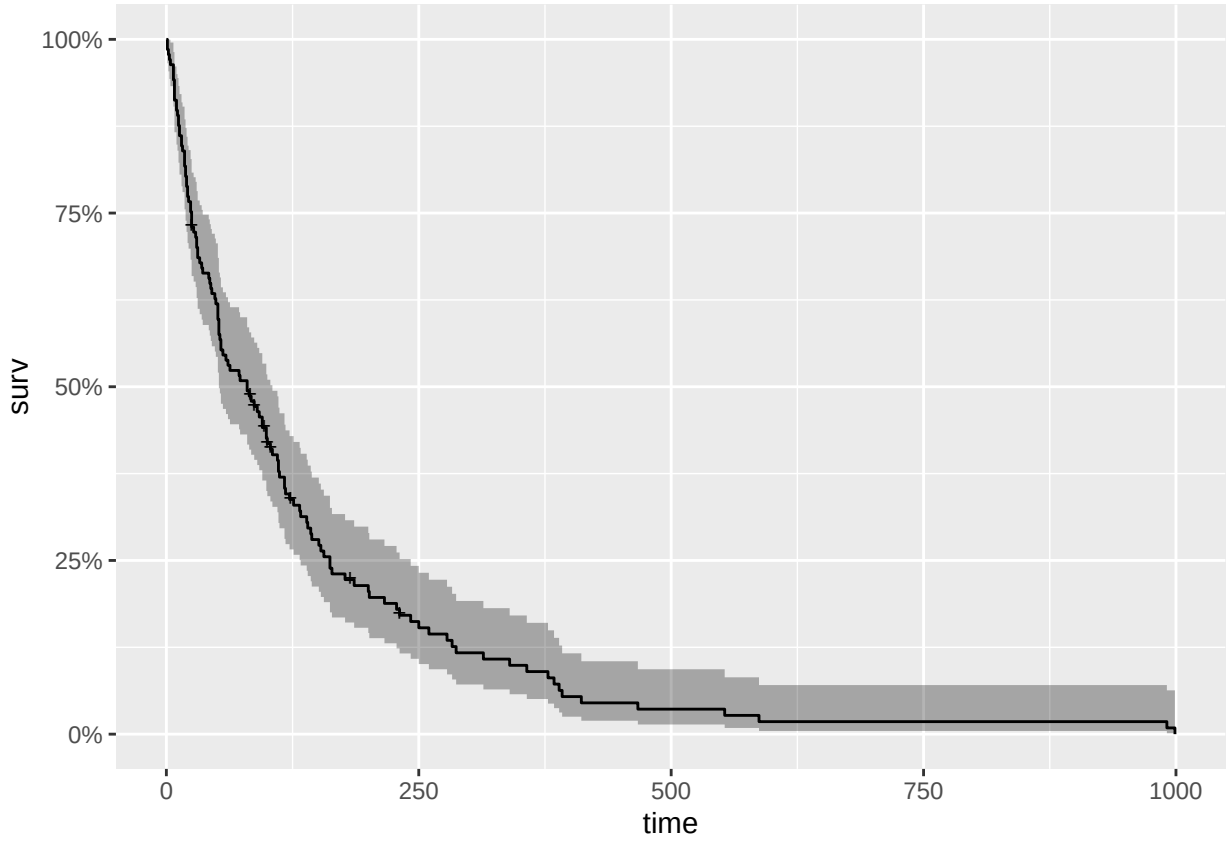
```
## Call: survfit(formula = km ~ 1, data = veteran)
##
##  time n.risk n.event survival std.err lower 95% CI upper 95% CI
##   365     10     118     0.09  0.0265       0.0506         0.16
```

```
summary(km_fit, times = c(1,30,60,90*(1:10)))
```

```
## Call: survfit(formula = km ~ 1, data = veteran)
##
##  time n.risk n.event survival std.err lower 95% CI upper 95% CI
##     1    137       2    0.985  0.0102      0.96552       1.0000
##    30     97      39    0.700  0.0392      0.62774       0.7816
##    60     73      22    0.538  0.0427      0.46070       0.6288
##    90     62      10    0.464  0.0428      0.38731       0.5560
##   180     27      30    0.222  0.0369      0.16066       0.3079
##   270     16       9    0.144  0.0319      0.09338       0.2223
##   360     10       6    0.090  0.0265      0.05061       0.1602
##   450      5       5    0.045  0.0194      0.01931       0.1049
##   540      4       1    0.036  0.0175      0.01389       0.0934
##   630      2       2    0.018  0.0126      0.00459       0.0707
##   720      2       0    0.018  0.0126      0.00459       0.0707
##   810      2       0    0.018  0.0126      0.00459       0.0707
##   900      2       0    0.018  0.0126      0.00459       0.0707
```

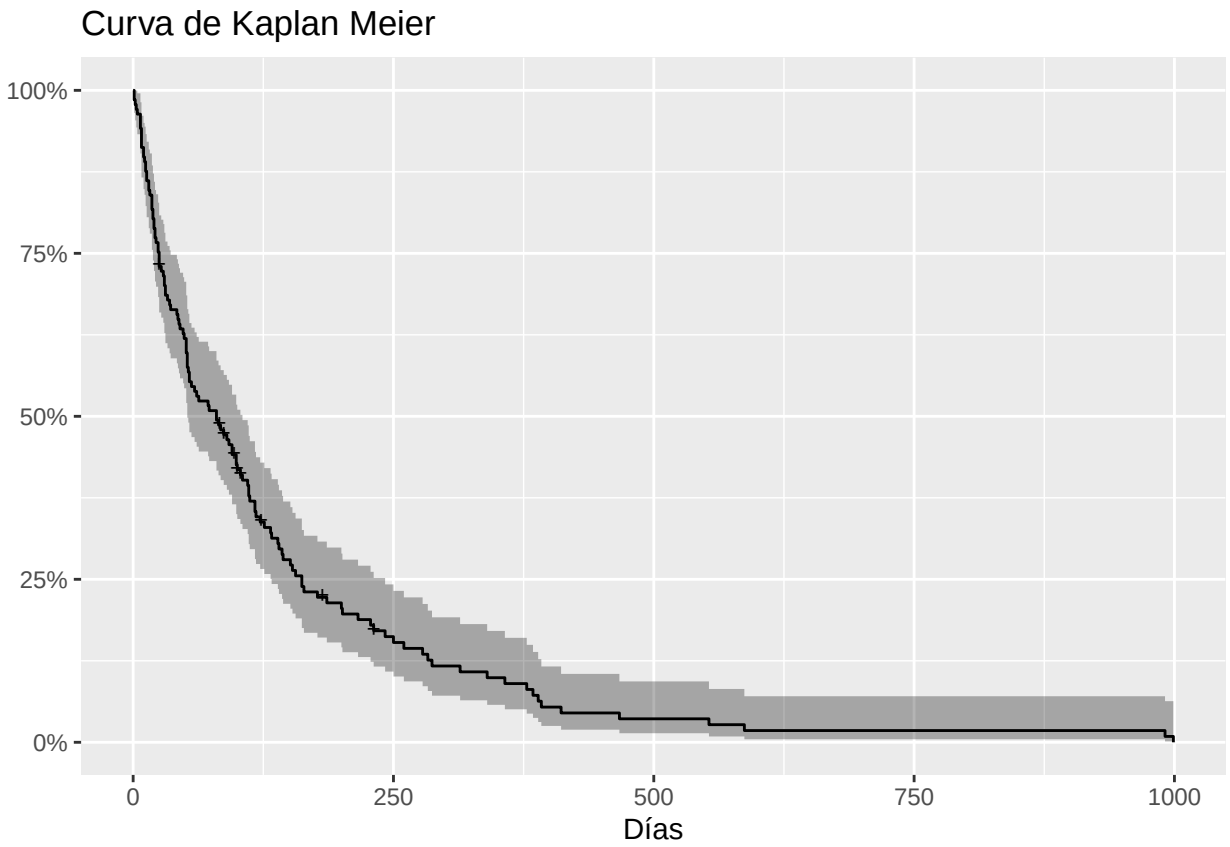
La representación gráfica de la curva de Kaplan-Meier la obtenemos a partir del objeto km_fit.

```
autoplot(km_fit)
```



Al construirse el gráfico bajo gramática "ggplot2" podemos modificarlo utilizando las funciones específica de ggplot2.

```
autoplot(km_fit) +
         ylab(NULL) +
         xlab("Días") +
         ggtitle("Curva de Kaplan Meier")
```



El análisis de superviviencia lo podemos realizar difenrenciando entre categorías de otra variable. Veamos primero las diferentes curvas por tipo de tratamiento, estándar y nuevo (antes he etiqueto la variable trt). Con survdiff(...) realizamos un test para comprobar si existe diferencia entre las curvas de supervivencia. Finalmente graficamos ambas curvas conjuntamente.

```
veteran$trt <- factor(veteran$trt,
                      levels = c(1, 2),
                      labels = c("Estándar", "Nuevo"))

km_trt_fit <- survfit(km ~ trt, data=veteran)
km_trt_fit
```

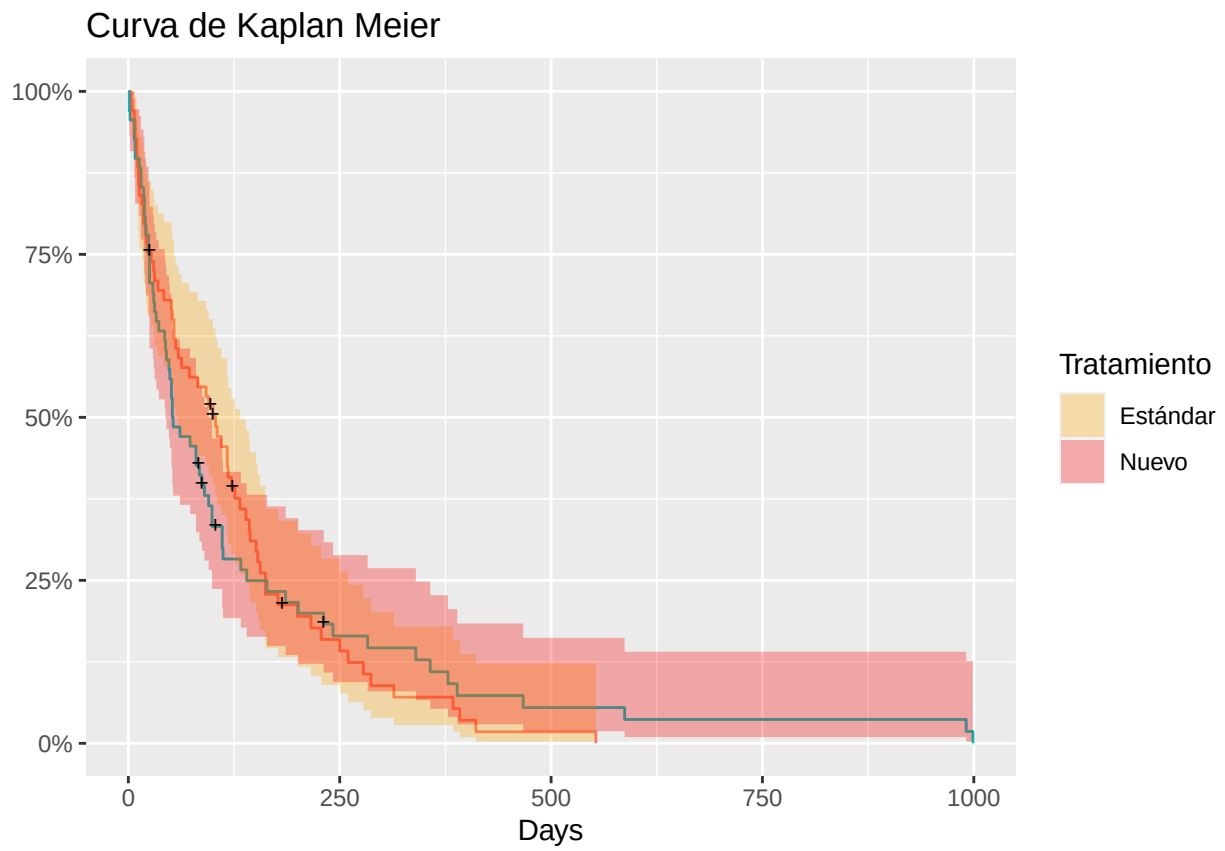
```
## Call: survfit(formula = km ~ trt, data = veteran)
##
##                n events median 0.95LCL 0.95UCL
## trt=Estándar 69     64  103.0      59     132
## trt=Nuevo    68     64   52.5      44      95
```

```
survdiff(km ~ trt, data=veteran)
```

```
## Call:
## survdiff(formula = km ~ trt, data = veteran)
##
##                N Observed Expected (O-E)^2/E (O-E)^2/V
## trt=Estándar 69       64     64.5   0.00388   0.00823
```

```
## trt=Nuevo    68       64     63.5   0.00394   0.00823
##
##  Chisq= 0  on 1 degrees of freedom, p= 0.9
```

```
legend_title <- c("Tratamiento")

autoplot(km_trt_fit) +
  ylab(NULL) +
  xlab("Days") +
  ggtitle("Curva de Kaplan Meier") +
  guides(col = FALSE)    +
  scale_fill_manual(legend_title,
                    values=c("orange","red"))
```



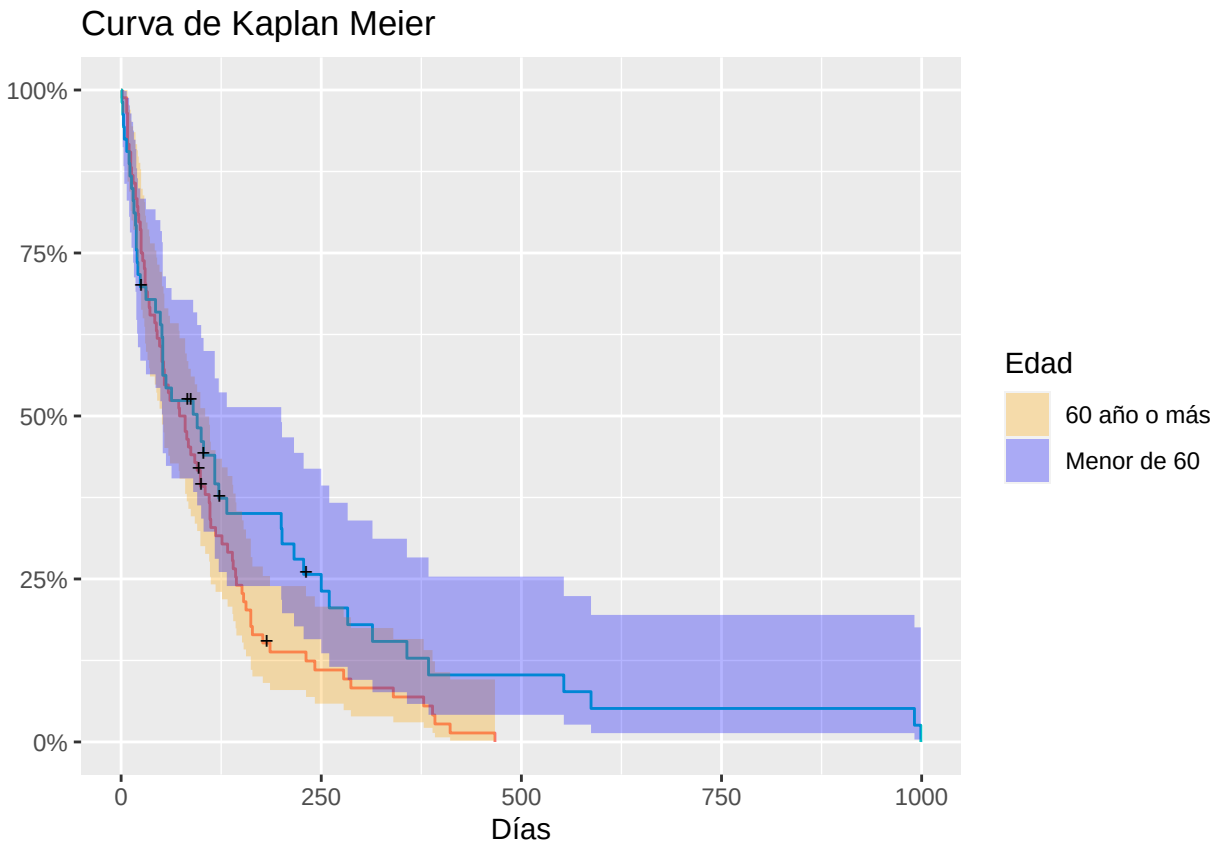
Prueba ahora a representar las curvas de indiferencia diferenciado a los individuos con menos de 60 años de los de 60 o más años.

```
veteran$Edadi <- ifelse((veteran$age < 60), "Menor de 60", "60 año o más")

km_trt_fit2 <- survfit(km ~ Edadi, data=veteran)

survdiff(km ~ Edadi, data=veteran)
```

```
## Call:
## survdiff(formula = km ~ Edadi, data = veteran)
```

```
##
##                      N Observed Expected (O-E)^2/E (O-E)^2/V
## Edadi=60 año o más 84       81     71.8      1.18      2.89
## Edadi=Menor de 60  53       47     56.2      1.51      2.89
##
##  Chisq= 2.9  on 1 degrees of freedom, p= 0.09
```

```
legend_title <- c("Edad")

autoplot(km_trt_fit2) +
  ylab(NULL) +
  xlab("Días") +
  ggtitle("Curva de Kaplan Meier") +
  guides(col = FALSE)    +
  scale_fill_manual(legend_title,
                    values=c("orange","blue"))
```



Ejercicio propuesto: Calcula y representa gráficamente las curvas de supervivencia respecto de la variable "celltype". Qué tipo de célula parece la más agresiva. Podrías contrastar estadísticamente dicha diferencia.

Análisis semiparamétrico: Modelo de Cox de riesgos proporcionales

La supervivencia entre dos tratamientos alternativos puede depender no sólo del tratamiento, sino también de otras variables como la edad, el sexo, o la gravedad de la afección de cada paciente. En las curvas de Kaplan-Meier se asume que los grupos son homogéneos, sin embargo no siempre es así. Los modelos de

regresión de Cox permiten controlar por los efectos de estas variables. La función coxph(...) nos estima la regresión de Cox.

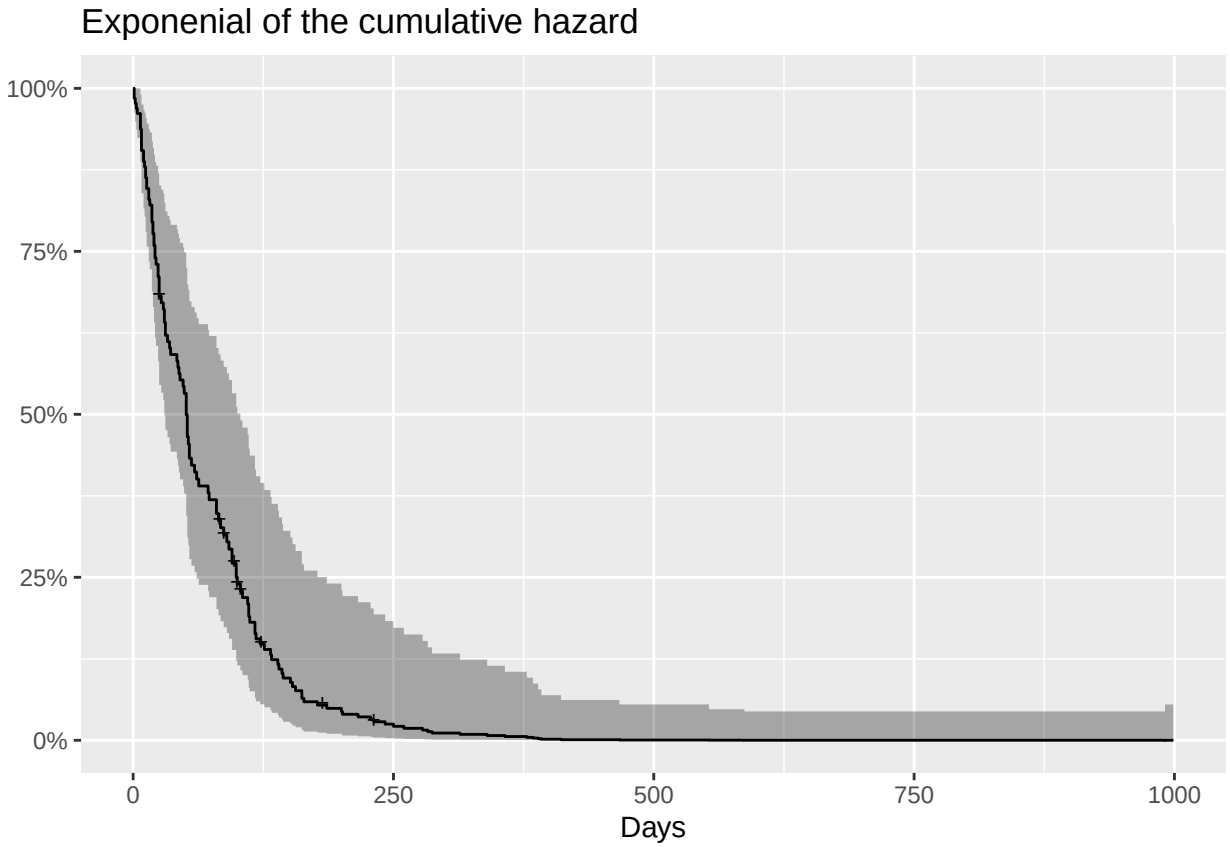
```
Mcox <- coxph(Surv(time, status) ~ trt + celltype + karno + diagtime + age + as.factor(prior) ,
              data = veteran)
summary(Mcox)
```

```
## Call:
## coxph(formula = Surv(time, status) ~ trt + celltype + karno +
##     diagtime + age + as.factor(prior), data = veteran)
##
##   n= 137, number of events= 128
##
##                           coef  exp(coef)   se(coef)      z Pr(>|z|)
## trtNuevo             2.946e-01  1.343e+00  2.075e-01  1.419  0.15577
## celltypelarge       -7.948e-01  4.517e-01  3.029e-01 -2.624  0.00869 **
## celltypesmallcell   -3.345e-01  7.157e-01  2.760e-01 -1.212  0.22548
## celltypesquamous    -1.196e+00  3.024e-01  3.009e-01 -3.975 7.05e-05 ***
## karno               -3.282e-02  9.677e-01  5.508e-03 -5.958 2.55e-09 ***
## diagtime             8.132e-05  1.000e+00  9.136e-03  0.009  0.99290
## age                 -8.706e-03  9.913e-01  9.300e-03 -0.936  0.34920
## as.factor(prior)10   7.159e-02  1.074e+00  2.323e-01  0.308  0.75794
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##                     exp(coef) exp(-coef) lower .95 upper .95
## trtNuevo               1.3426     0.7448    0.8939    2.0166
## celltypelarge          0.4517     2.2139    0.2495    0.8178
## celltypesmallcell      0.7157     1.3972    0.4167    1.2292
## celltypesquamous       0.3024     3.3071    0.1677    0.5454
## karno                  0.9677     1.0334    0.9573    0.9782
## diagtime               1.0001     0.9999    0.9823    1.0182
## age                    0.9913     1.0087    0.9734    1.0096
## as.factor(prior)10     1.0742     0.9309    0.6813    1.6937
##
## Concordance= 0.736  (se = 0.021 )
## Likelihood ratio test= 62.1  on 8 df,   p=2e-10
## Wald test            = 62.37  on 8 df,   p=2e-10
## Score (logrank) test = 66.74  on 8 df,   p=2e-11
```

```
Mcox_fit <- (survfit(Mcox, stype=2, ctype=2))
Mcox_fit
```

```
## Call: survfit(formula = Mcox, stype = 2, ctype = 2)
##
##        n events median 0.95LCL 0.95UCL
## [1,] 137    128     51      30     103
```

```
autoplot(Mcox_fit) +
  ylab(NULL) +
  xlab("Days") +
  ggtitle("Exponenial of the cumulative hazard")
```



El modelo marca el tipo de célula pequeña, el tipo de célula adeno y karno como significativos. Sin embargo, es necesario tener cierta precaución al interpretar estos resultados. Si bien se cree que el modelo de riesgo proporcional de Cox es "robusto", un análisis cuidadoso verificaría los supuestos subyacentes al modelo. Por ejemplo, el modelo de Cox asume que las covariables no varían con el tiempo.

El paquete "surv" permite calcular estad´sticos del efecto individual de las variables predictoras. Mediante la función aareg(...) ajustamos el modelo de regresión aditiva de Aalen para datos censurados

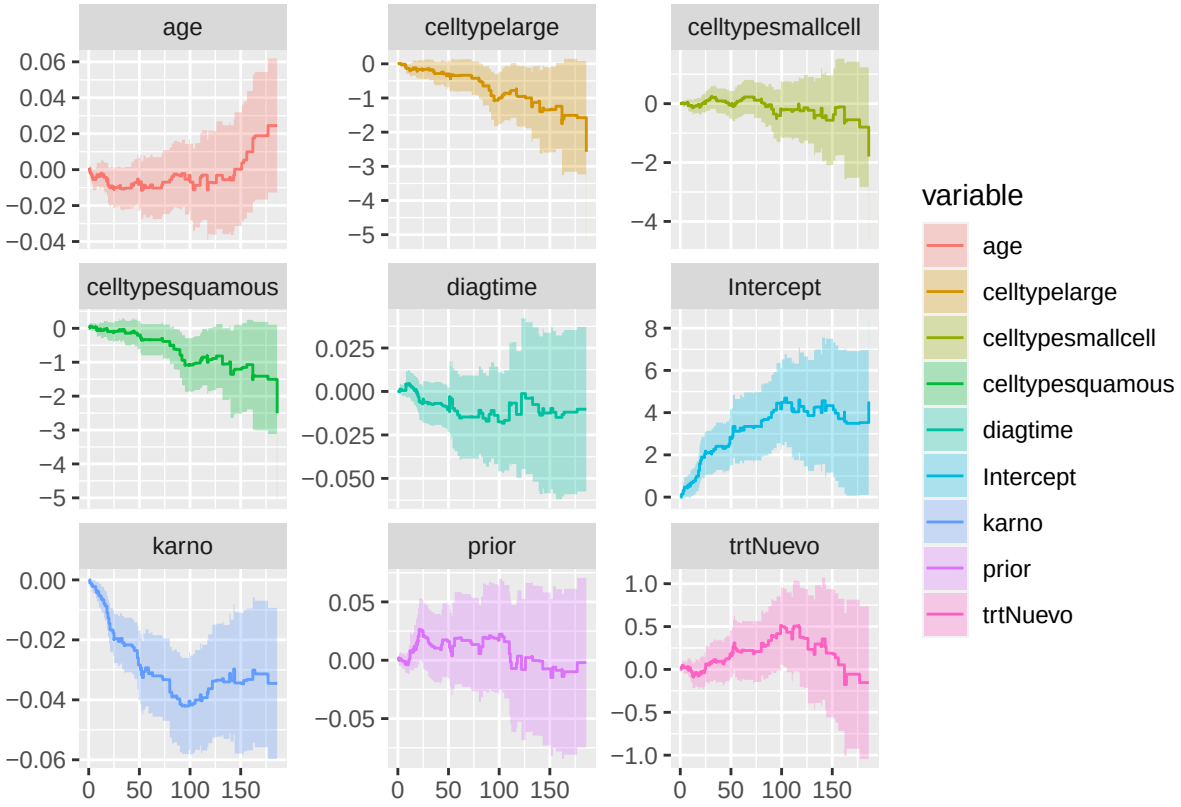
```
aa_fit <-aareg(Surv(time, status) ~ trt + celltype +
                 karno + diagtime + age + prior ,
                 data = veteran)
aa_fit
```

```
## Call:
## aareg(formula = Surv(time, status) ~ trt + celltype + karno +
##     diagtime + age + prior, data = veteran)
##
##   n= 137
##     73 out of 97 unique event times used
##
##                       slope      coef se(coef)      z        p
## Intercept          0.122000  5.12e-02 1.16e-02  4.400 1.10e-05
## trtNuevo           0.006640  2.50e-03 2.59e-03  0.968 3.33e-01
## celltypelarge     -0.018700 -1.19e-02 4.20e-03 -2.830 4.68e-03
## celltypesmallcell -0.002880 -2.70e-03 4.51e-03 -0.599 5.49e-01
## celltypesquamous  -0.016700 -1.03e-02 4.20e-03 -2.450 1.42e-02
```

```
## karno             -0.001170 -4.44e-04 8.78e-05 -5.060 4.20e-07
## diagtime          -0.000245 -7.16e-05 1.63e-04 -0.440 6.60e-01
## age               -0.000250 -4.81e-05 1.28e-04 -0.377 7.06e-01
## prior              0.000334  1.27e-04 2.85e-04  0.445 6.57e-01
##
## Chisq=43.53 on 8 df, p=6.99e-07; test weights=aalen
```

Los gráficos muestran cómo cambian en el tiempo los efectos de las covariables (riesgo relativo).

```
autoplot(aa_fit)
```



Ejercio propuesto

El dataset "breast.dta" contiene información sobre ciertos factores pronósticos de la mortalidad por cáncer de mama: edad, grado del tumor (en tres niveles), tamaño del tumor (en cm) y estado de los ganglios linfáticos.

Disponemos también de información sobre el tiempo de supervivencia (en días) después del tratamiento además de si el paciente murió o está censurado.

Considera primero un modelo solo con las covariables edad y tamaño del tumor. Utiliza un modelo de supervivencia de Cox.

Trata de responder las siguientes preguntas:

1. ¿Cuánto aumenta el riesgo de morir con un aumento del tamaño del tumor de 1 cm?
2. ¿Cuánto aumenta el riesgo de morir con un aumento de la edad de 10 años?
3. ¿Cuál es la razón de la probabilidad de morir entre un paciente de 75 años con un tamaño de tumor de 3 cm y un paciente de 55 años con un tamaño de tumor de 4 cm?

4. ¿Podemos concluir a partir de estos datos que un aumento del tamaño del tumor de 2 cm se asocia con al menos un aumento del 60% del riesgo de morir?
5. Intente ahora tener en cuenta las cuatro covariables pronósticas. ¿Es razonable incoroporar las dos covariables categóricas del estado y la clasificación de los ganglios linfáticos como continuas?

Análisis de intervenciones sobre datos de series temporales

Cerramos este capítulo con una introducción a algunas técnicas para la búsqueda del efecto, sobre datos de series temporales, de una intervención temporal en materia de, por ejemplo, políticas públicas. Como ilustración trabajaremos con el dataset "cigsales.dta". Esta base de datos muestra el efecto que en tuvo un cambio en la fiscalidad del tabaco sobre las ventas de cigarillos en el estado de California. Esta intervención entró en vigor en 1989 y diferención claramente los precios del tabaco en California en comparación con el resto de estados.

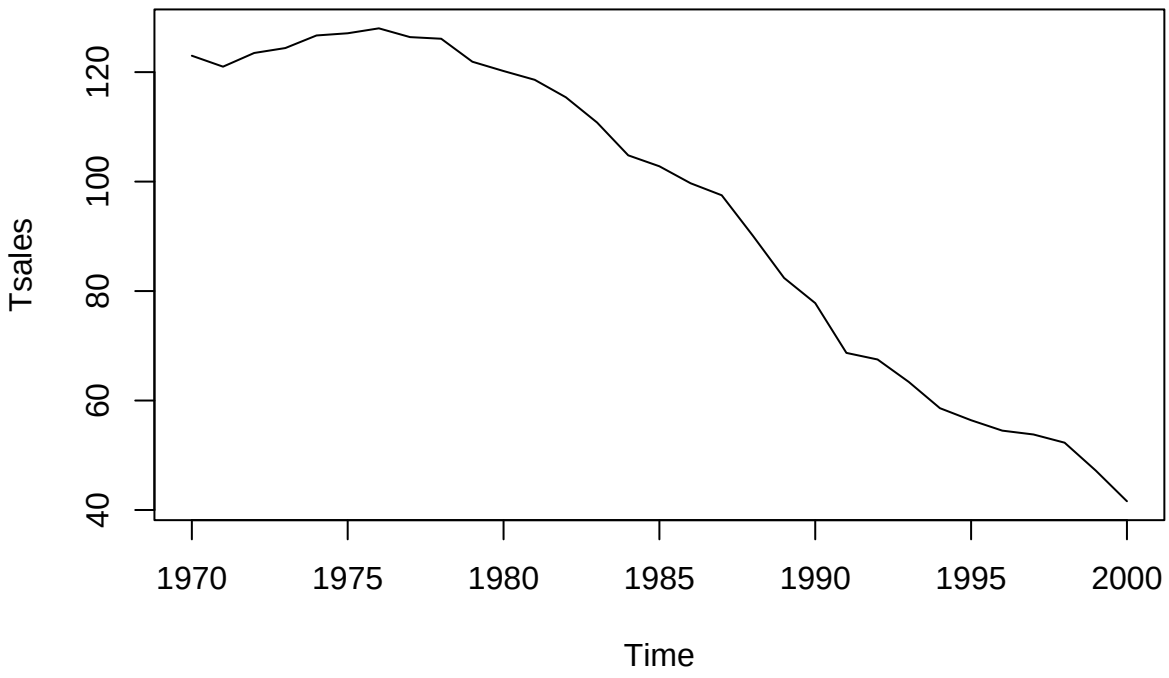
Paquete changepoint

El objetivo de este paquete es la búsqueda automática de cambios (en la media o en la varianza) en datos de una serie temporal. A continuación, cargo los datos y los pquetes necesarios:

```
library(changepoint)
library(ggplot2)
library(haven)
cigsales <- read_dta("cigsales.dta")

California <-subset(cigsales, state==3)
```

La función ts(...) de R base se utiliza para definir un objeto como una serie temporal

```
Tsales <- ts(California$cigsale, frequency=1, start=c(1970))
plot(Tsales)
```



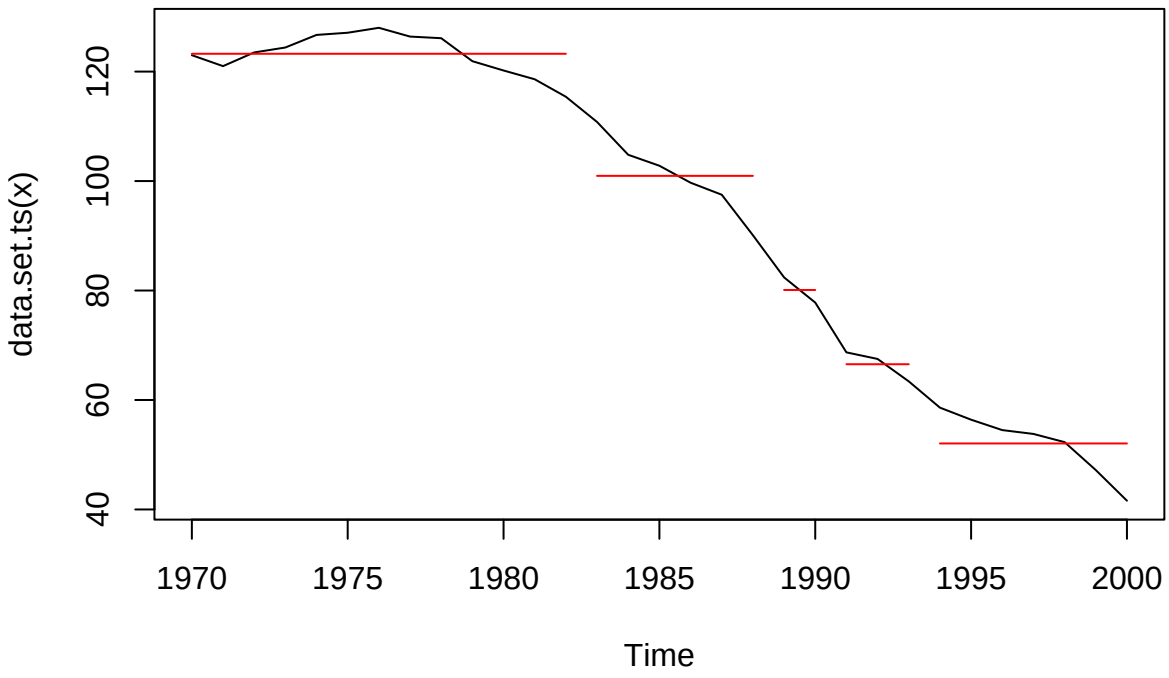
En primer lugar buscamos un máximo de 5 change points para luego acotar en 1 solo, esperando que identifique la intervención en 1989.

```
m1 <- cpt.mean(Tsales, penalty = "BIC", method = "BinSeg", Q = 4)
m1
```

```
## Class 'cpt' : Changepoint Object
##         ~~   : S4 class containing 14 slots with names
##                cpts.full pen.value.full data.set cpttype method test.stat pen.type pen.value minsegle
##
## Created on  : Sun Nov  6 01:51:48 2022
##
## summary(.)  :
## ----------
## Created Using changepoint version 2.2.4
## Changepoint type      : Change in mean
## Method of analysis    : BinSeg
## Test Statistic  : Normal
## Type of penalty       : BIC with value, 6.867974
## Minimum Segment Length : 1
## Maximum no. of cpts   : 4
## Changepoint Locations : 13 19 21 24
## Range of segmentations:
##      [,1] [,2] [,3] [,4]
## [1,]   19   NA   NA   NA
## [2,]   19   13   NA   NA
## [3,]   19   13   24   NA
```

```
## [4,]   19   13   24   21
##
##  For penalty values: 22950.03 2042.21 1155.361 220.8654
```

```
plot(m1)
```

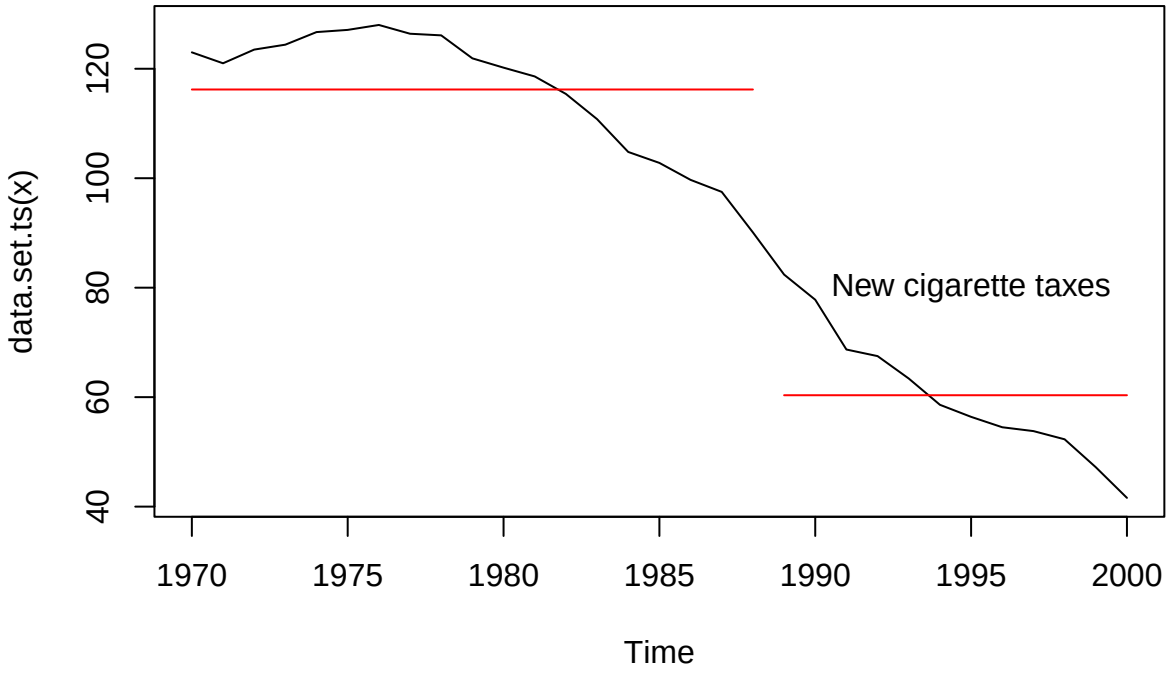



```
m2 <- cpt.mean(Tsales, penalty = "BIC", method = "BinSeg", Q = 1)
m2
```

```
## Class 'cpt' : Changepoint Object
##         ~~   : S4 class containing 14 slots with names
##                cpts.full pen.value.full data.set cpttype method test.stat pen.type pen.value minsegle
##
## Created on  : Sun Nov  6 01:51:48 2022
##
## summary(.)  :
## ----------
## Created Using changepoint version 2.2.4
## Changepoint type      : Change in mean
## Method of analysis    : BinSeg
## Test Statistic  : Normal
## Type of penalty       : BIC with value, 6.867974
## Minimum Segment Length : 1
## Maximum no. of cpts   : 1
## Changepoint Locations : 19
```

```
## Range of segmentations:
##      [,1]
## [1,]   19
##
##  For penalty values: 22950.03
```

```
plot(m2)
text(1995, 80, "New cigarette taxes")
```



Análisis econométric mediante cambios de tendencia en estimación OLS

Conocido como series temporales interrumpidas, estimamos un modelo lm(...) incorporando una dummy para medir el efecto puntual de la intervención y posteriormente su interacción con la tendencia para medir el cambio en tendencia.

```
library(sandwich)
library(lmtest)
library(car)

California$trend <-c(California$year-rep(1969,31))
California$tax_dummy <- ifelse(California$year>=1989, 1, 0)
California$t_taxdummy <-ifelse(California$year>1989,
                               California$year-1989, 0)
```

```
ITSA1 <- lm(cigsale ~ trend + tax_dummy + t_taxdummy, data=California)
summary(ITSA1)
```

```
##
## Call:
## lm(formula = cigsale ~ trend + tax_dummy + t_taxdummy, data = California)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.0953  -2.9216  -0.7423   3.9069   8.1100
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 134.0053     2.4746  54.152  < 2e-16 ***
## trend        -1.7795     0.2170  -8.199 8.36e-09 ***
## tax_dummy   -20.0581     3.7471  -5.353 1.18e-05 ***
## t_taxdummy   -1.4947     0.4846  -3.084  0.00467 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.182 on 27 degrees of freedom
## Multiple R-squared:  0.9732, Adjusted R-squared:  0.9702
## F-statistic: 326.4 on 3 and 27 DF,  p-value: < 2.2e-16
```

A continuación contrasto el efecto sobre la tendencia

```
hypothesis <- "trend = -t_taxdummy"
df <- ITSA1$df.residual
test <- linearHypothesis (ITSA1, hypothesis)
Fstat <- test$F[2]
pval <- 1-pf(Fstat, 1, df)
Coeff <- ITSA1$coefficients[2] + ITSA1$coefficients[4]

Tendencia.Test <- c(Coeff, Fstat, pval)
names(Tendencia.Test) <- c("Coeff", "Fstat", "pval")
Tendencia.Test
```

```
##         Coeff         Fstat          pval
## -3.274126e+00  5.709358e+01  3.968305e-08
```

Al ser datos de series temporales combiene controlar por la autocorrelación. Para ello reestimamos los errores estándar robustos de Newey West

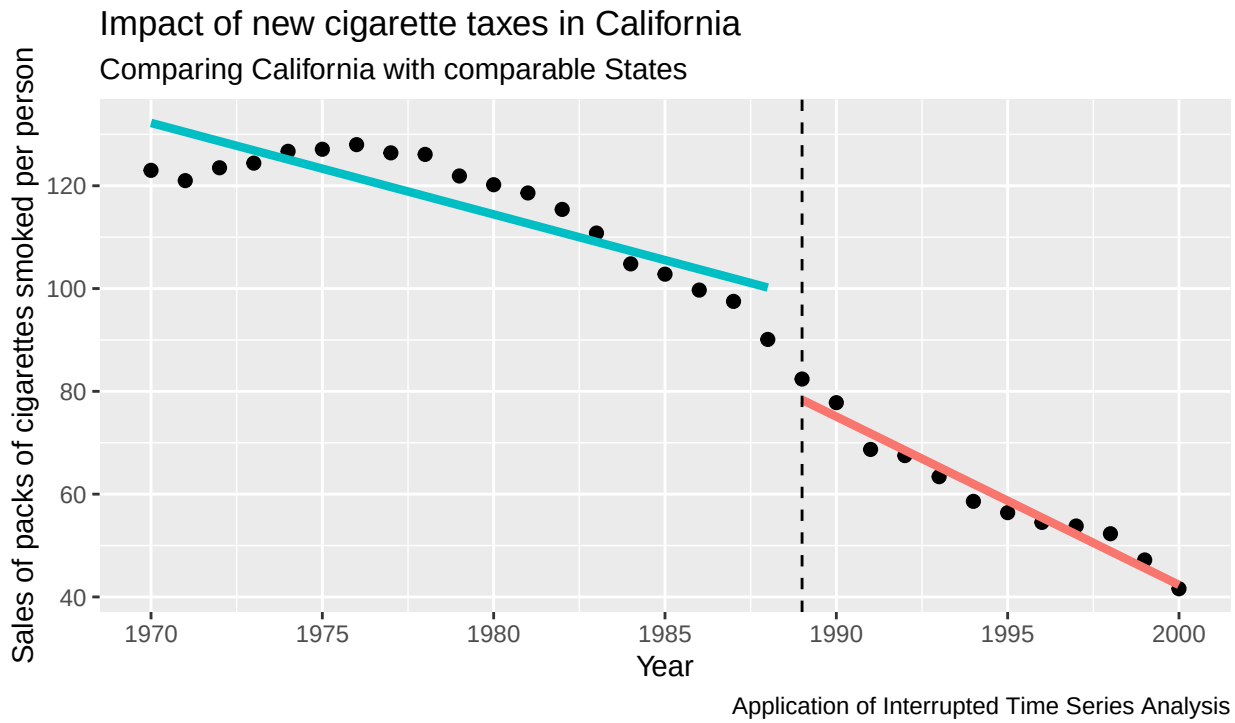
```
coeftest(ITSA1, vcov=NeweyWest(ITSA1))
```

```
##
## t test of coefficients:
##
##              Estimate Std. Error t value  Pr(>|t|)
## (Intercept) 134.00526    5.26310 25.4613 < 2.2e-16 ***
## trend        -1.77947    0.41006 -4.3395 0.0001793 ***
```

```
## tax_dummy   -20.05810    6.86746 -2.9207 0.0069714 **
## t_taxdummy   -1.49465    0.43478 -3.4378 0.0019163 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Finalmente mostramos el ajuste en un gráfico ggplot2 (que seguro sabes hacer)

```
California$preditsa = predict(ITSA1)

California$pre = c(predict(ITSA1)[1:19], rep(NA,12))
California$post = c(rep(NA,19), predict(ITSA1)[20:31])


ggplot(California, aes(x = year, y = cigsale)) +
  geom_point(size=2) +
  geom_line(aes(y = pre, color="red"), size = 1.5) +
  geom_line(aes(y = post, color="blue"), size = 1.5) +
  scale_x_continuous(limits = c(1970, 2000),
                     breaks = seq(1970,2000,5)) +
  geom_vline(xintercept = 1989, linetype="dashed") +
  labs(title = "Impact of new cigarette taxes in California",
       subtitle = "Comparing California with comparable States" ,
       caption = " Application of Interrupted Time Series Analysis",
       y = "Sales of packs of cigarettes smoked per person", x= "Year") +
  coord_fixed(ratio = .15) +
  theme(plot.caption = element_text(lineheight=.5), legend.position = "none")
```



Estimación del efecto causal creando un control sintético

Con los anteriores paquetes hemos trabajado la serie para las ventas en California por separado. Hemos encontrado un efecto de la política de impuestos. Pero no podemos hablar de causalidad pues no hemos tenido contrafactual. Este es el objetivo del siguiente paquete.

El paquete "CausalImpact" contruye un cotrafactual a patir de la información de ventas de cigarrillos en otros estados no afectados por la intervención. Estados con características muy similares a California. Estos son Alabama 1, Arkansas 2 y West Virginia 37.

Para información sobre le paquete CausalImpact

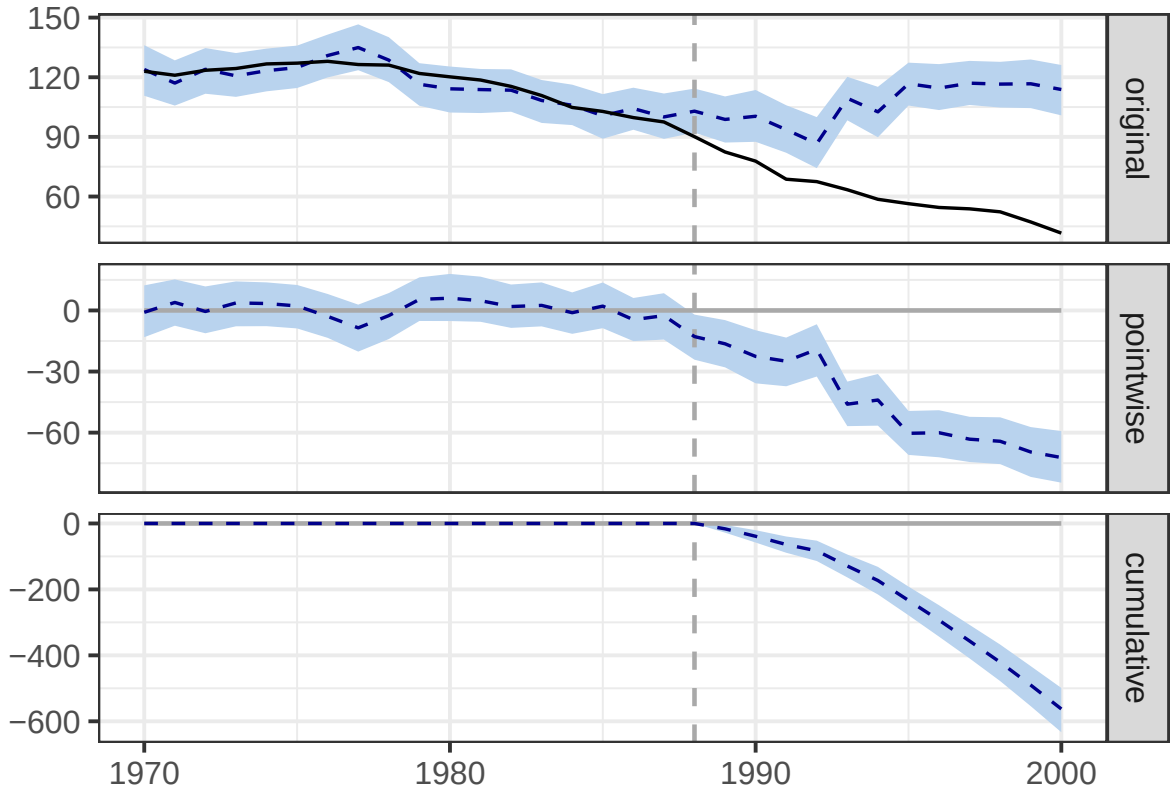
http://google.github.io/CausalImpact/CausalImpact.html

```
library(CausalImpact)
```

Definimos los periodos pre y post-intervención, y estimamos un modelo de series temporales bayesianas

```
time.points <-seq(1970,2000)

y<-cigsales$cigsale[cigsales$state==3]
  x1<-cigsales$cigsale[cigsales$state==1]
  x2<-cigsales$cigsale[cigsales$state==2]
  x37<-cigsales$cigsale[cigsales$state==37]


data3 <- zoo(cbind(y,x1,x2,x37), time.points)

  plot(data3)
```


```
pre.period <- c(1970, 1988)
post.period <- c(1989, 2000)

impact <- CausalImpact(data3, pre.period, post.period)
plot(impact)
```




```
summary(impact)
```

```
## Posterior inference {CausalImpact}
##
##                          Average        Cumulative
## Actual                   60             724
## Prediction (s.d.)        107 (2.8)      1287 (33.8)
## 95% CI                   [102, 113]     [1223, 1357]
##
## Absolute effect (s.d.)   -47 (2.8)      -563 (33.8)
## 95% CI                   [-53, -42]     [-633, -499]
##
## Relative effect (s.d.)   -44% (2.6%)    -44% (2.6%)
## 95% CI                   [-49%, -39%]   [-49%, -39%]
##
## Posterior tail-area probability p:   0.00101
## Posterior prob. of a causal effect:  99.89889%
##
## For more details, type: summary(impact, "report")
```