MEMORIA PRÁCTICA – 5:
Correlación y regresión.

Escuela Politécnica Superior - Universidad Autónoma de Madrid

GRADO EN INGENIERÍA BIOMÉDICA

BIOESTADÍSITICA

Versión del documento número 1
Práctica realizada por: *Laura Sánchez Garzón*
Fecha: *5 de mayo de 2023*
Profesorado de la práctica: *Mercedes Sotos Prieto y María Téllez Plaza*

1. ÍNDICE

Contenido

2. INTRODUCCIÓN

En la práctica 3, vimos algunas técnicas sencillas para evaluar la asociación entre 2 variables categóricas mediante tablas de contingencia, pero es posible que también se desee modelar las relaciones dosis-respuesta (donde el predictor o variable independiente es una variable continua) o modelar múltiples variables simultáneamente. Así, se ve la necesidad de utilizar técnicas parecidas de modelado como las que vimos en la Práctica 4 para las variables continuas con la regresión lineal.

En esta práctica, veremos cómo realizar un análisis de regresión logística en R. Hay una gran similitud con el material relacionado con los modelos lineales, ya que la descripción de los modelos es bastante similar, pero también hay algunos aspectos especiales relacionados con las tablas de “deviance”, cuyo concepto se explicará más adelante.

3. EJERCICIOS GUIADOS

3.1. Regresión logística

A veces se desea modelar variables dicotómicas que pueden tener sólo dos valores posibles, por ejemplo: enfermo/no enfermo o verdadero/falso. Ahora, partiendo de los conocimientos obtenido en la práctica 4, se puede aplicar lo estudiado sobre los modelos lineares, que no son apropiados para modelar probabilidades ya que las probabilidades van de 0 a 1 **y los modelos de regresión logística podrían predecir valores fuera de escala por debajo de cero o por encima de 1**. Para esto se modelan las probabilidades en una escala transformada, quedando un modelo lineal para probabilidades transformadas se puede configurar como:

$$\text{logit } p = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ...\beta_k x_k$$

en la que **logit p = log[p/(1 – p)]**. La elección de la función logit no es la única posible, pero tiene algunas propiedades matemáticamente convenientes. Una cosa a tener en cuenta sobre el modelo logístico es que **no hay ningún término de error como en los modelos lineales**, estamos modelando **la probabilidad de un evento directamente, y eso ya es un valor esperado**. Además, los parámetros del modelo se pueden estimar por el método de máxima verosimilitud, técnica parecida al método de mínimos cuadrados que veíamos en la regresión lineal, que optimiza un criterio de bondad de ajuste.

Durante los ejercicios guiados se mostrará también conceptos como tablas de “deviance” (una medida que compara cómo de lejos está nuestro modelo de un modelo “saturado”, que incluiría todas nuestras variables, así como sus interacciones) y la especificación de modelos para datos pre-tabulados (agregados).

3.1.1. Interpretación de la salida del modelo logístico

El análisis de regresión logística pertenece a la clase de modelos lineales generalizados, caracterizados por una **distribución binomial de respuesta** y una **función de enlace (logit p)**, **que transforma el valor esperado en una escala en la que la relación con las variables de ajuste se describe como lineal y aditiva**.

En R, los modelos lineales generalizados son manejados por la función *glm()* (“Generalized Linear Models ”), función muy similar a lm() (usada para modelos lineales). Las dos funciones usan esencialmente el mismo formato de fórmula dentro del modelo, además de funciones extractoras (como *summary()*), pero *glm* también **debe especificar qué tipo de modelo lineal generalizado se desea, mediante el argumento family**, así como al utilizar la función de enlace logit, es decir ***family=binomial("logit")***.

3.1.1.1. Ejemplo sobre el uso de la regresión logística

Se utilizarán para el experimento datos tabulares, en concreto los correspondientes a un experimento (Hemmingsen y Krogh, 1926) en el que a los ratones se les inyectó una dosis de insulina. Se registró el número de ratones que mostraron síntomas adversos.

Dosis	Número de ratones (n)	Número con síntomas (s)	$p_i=s/n$
3.4	33	0	0
5.2	32	5	0.156
7.0	38	11	0.289
8.5	37	14	0.378
10.5	40	18	0.45
13.0	37	21	0.567
18.0	31	23	0.742
21.0	37	30	0.81
28.0	30	27	0.9

Figura 0: Datos recogidos de ratones a los que se les inyectó una dosis de insulina

```
> #Primero cargaremos los datos:
> dat <- as.data.frame(cbind(c(3.4, 5.2, 7.0, 8.5, 10.5, 13.0, 18.0, 21.0, 28.0),c(33, 32, 38, 37, 40, 37, 3
1, 37, 30), c(0, 5, 11, 14, 18, 21, 23, 30, 27)) )
> names(dat) <- c("dosis", "n", "s")
> attach(dat)
The following objects are masked _by_ .GlobalEnv:

    n, s

The following objects are masked from dat (pos = 3):

    dosis, n, s

> #attach(dat)
> n=dat$n
> s=dat$s
> n
[1] 33 32 38 37 40 37 31 37 30
> s
[1]  0  5 11 14 18 21 23 30 27
```

Figura 1: n y s recogen respectivamente los datos de las columnas Número de ratones y Número con síntomas.

Observar que la función attach() dio error, se tuvo que recurrir a la opción manual.

```
> #R puede ajustar análisis de regresión logística para datos tabulares de dos maneras diferentes.
> #En la primera opción, hay que especificar la respuesta como una matriz, donde una columna es el
> #número de "enfermos" y el otro es el número de "sanos" (o "éxito" y "fracaso", según el contexto).
> #Es decir:
> mice.tbl <- cbind(s,(n-s))
> mice.tbl
      s
 [1,]  0 33
 [2,]  5 27
 [3,] 11 27
 [4,] 14 23
 [5,] 18 22
 [6,] 21 16
 [7,] 23  8
 [8,] 30  7
 [9,] 27  3
> #La función cbind ("c" para "columna") se utiliza para vincular variables, por columnas, para formar
> #una matriz.
> #Hay que tener cuidado en no cometer el error de usar el conteo total para la columna 2 en lugar del
> #número de fallos.
```

Figura 2: mice.tbl recoge una matriz en la que la primera columna indica el número de enfermos (s, Número con síntomas), y la segunda n-s (número de sanos que van quedando en el tiempo).

```
> #Luego, se puede especificar el modelo de regresión logística como:
> glm(mice.tbl~dosis,family=binomial("logit"))

Call:  glm(formula = mice.tbl ~ dosis, family = binomial("logit"))

Coefficients:
(Intercept)        dosis
     -2.444        0.193

Degrees of Freedom: 8 Total (i.e. Null);  7 Residual
Null Deviance:      116.8
Residual Deviance: 13.84        AIC: 47.41
```

Figura 3: Se aplica la función glm() ("Generalized Linear Models ") y se obtiene el intercepto (β_0 = 0.193). Null Deviance representa cúanto de bien es predicha la respuesta del modelo utilizando únicamente el intercepto. Residual Deviance muestra cómo de bien es predicha la respuesta al incluir los predictores (variable independiente).

```
> prop.mice <- s/n
> prop.mice
[1] 0.0000000 0.1562500 0.2894737 0.3783784 0.4500000 0.5675676 0.7419355 0.8108108 0.9000000
> glm(prop.mice~dosis, family=binomial("logit"),weights=n)

Call:  glm(formula = prop.mice ~ dosis, family = binomial("logit"),
    weights = n)

Coefficients:
(Intercept)        dosis
     -2.444        0.193

Degrees of Freedom: 8 Total (i.e. Null);  7 Residual
Null Deviance:      116.8
Residual Deviance: 13.84        AIC: 47.41
```

Figura 4: En prop.mice se guarda el ratio. Antes de aplicar la función glm(), es necesario dar pesos (weights=n) porque R no puede ver en cuántas observaciones se basa una proporción (si no se dieran pesos, se entendería que todas las dosis se han administrado al mismo número de ratones). También en el resultado de glm hay información no visible, que se puede obtener con funciones Extractoras.

```
> summary(glm(mice.tbl~dosis,family=binomial("logit")))

Call:
glm(formula = mice.tbl ~ dosis, family = binomial("logit"))

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.1954  -0.5193   0.0655   0.6275   0.8940

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -2.44388    0.30018  -8.141 3.91e-16 ***
dosis        0.19295    0.02351   8.208 2.25e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 116.750  on 8  degrees of freedom
Residual deviance:  13.841  on 7  degrees of freedom
AIC: 47.408

Number of Fisher Scoring iterations: 4
```

Figura 5: Salida de la función summary.

A continuación, repasaremos los componentes de la salida de summary(glm(mice.tbl dosis,family=binomial("logit"))).

Deviance Residuals es la distribución de la contribución de cada celda de la tabla a la desviación del modelo (la desviación conceptualmente se correspondería con la suma de cuadrados en los modelos lineares).

Coefficients es la tabla de interés principal. Aquí, obtenemos estimaciones de los coeficientes de regresión, errores estándar de los mismos y pruebas para determinar la hipótesis nula de que cada coeficiente de regresión es cero. La salida es casi idéntica a la salida que obteníamos con lm().

Residual Deviance es análogo a la regresión lineal donde se estimaba la desviación estándar de los errores alrededor de la línea de regresión. En modelos GLM con family=binomial, se puede utilizar la desviación para evaluar la especificación del modelo (es decir, la importancia de cada variable predictora), cómo veremos a continuación en la sección de selección de variables.

AIC (criterio de información de Akaike) es una medida de bondad global de ajuste que toma el número de parámetros del modelo en cuenta. La distribución asintótica de la desviación residual es una distribución χ^2 con los grados de libertad indicados.

```
> qchisq(p=0.05, df=7, lower.tail = FALSE)
[1] 14.06714
> pchisq(q=102.909, df=(8-7), lower.tail = FALSE)
[1] 3.509019e-24
```

Figura 6: pchisq devuelve el valor del test de la chi-cuadrado de la densidad acumulatia; mientras que qchisq devuelve el resultado de aplicar la chi-cuadrado a los cuantiles de la función.

En la función *summary()* veíamos *Residual Deviance* = 13.841, resultado muy cercano al límite de significación estadística del 5%, así que esto es evidencia de que las variables del modelo explican una parte relevante de la desviación. La desviación nula es la desviación de un modelo que contiene solo el intercepto (es decir, describe una probabilidad fija para la ocurrencia de síntomas, en todas las celdas irrespectivamente de la dosis).

Lo que también podría interesar es la diferencia con la desviación residual, aquí 116.750−13.841 = 102.909, que se puede usar para un test conjunto de todas las variables si hay algún efecto global presente en el modelo. En este caso, la variable “dosis” claramente explica una gran cantidad de la deviance.

Finalmente, Number of Fisher Scoring iterations: 4. Se refiere al procedimiento de estimación y es un elemento puramente técnico. No hay información estadística en él, pero se debe vigilar si el número de iteraciones se vuelve demasiado grande porque eso podría ser una señal de que el modelo es demasiado complejo para ajustarse según los datos disponibles. Normalmente, glm() detiene el procedimiento de ajuste si el número de iteraciones excede 25, pero es posible reconfigurar el límite.

Finalmente, se ha vuelto tradicional **presentar análisis de regresión logística en términos de razones de odds**, que se interpreta como **incremento relativo en las odds de tener el evento/pertenecer a una categoría definida en la variable respuesta por un incremento en una unidad de la covariable**. Esto se consigue con el antilogaritmo (exponenciación cuando el logaritmo es en base e) de los coeficientes de regresión debido a que por las reglas de los logaritmos log(A) - log(B) = log(A/B). Dado que los errores estándar se vuelven multiplicativos y tienen poco sentido después de la transformación, también se acostumbra a **dar intervalos de confianza** en su lugar. Esto se puede obtener muy fácilmente de la siguiente manera:

```
> exp(cbind( coef(glm(mice.tbl~dosis,family=binomial("logit"))),
+            confint(glm(mice.tbl~dosis,family=binomial("logit")) )))
Waiting for profiling to be done...
                               2.5 %    97.5 %
(Intercept) 0.08682294 0.04707246 0.1531231
dosis       1.21282410 1.16082555 1.2732223
```

Figura 7: Intercept es las log-odds de que los ratones presenten síntomas. Si exponenciamos el intercepto obtendremos exp(0.08682294)= p/(1−p), y sólo habría que despejar para tener la probabilidad de síntomas en nuestros datos cuando todas las demás covariables son 0.

3.2. Predicción y bondad de ajuste del modelo logístico

Para dibujar un gráfico de la curva ajustada con predicción podemos usar *predict()*. Para evitar problemas causados por la falta de ordenación de los datos por dosis, usamos la opción newdata, que permite la predicción de valores para un conjunto elegido de predictores.

```
> fit <- glm(mice.tbl~dosis,family=binomial("logit"))
> fit

Call:  glm(formula = mice.tbl ~ dosis, family = binomial("logit"))

Coefficients:
(Intercept)        dosis
     -2.444        0.193

Degrees of Freedom: 8 Total (i.e. Null);  7 Residual
Null Deviance:      116.8
Residual Deviance: 13.84        AIC: 47.41
> pred.frame <- data.frame(dose=c(3.4, 5.2, 7.0, 8.5, 10.5, 13.0, 18.0, 21.0, 28.0))
> pred.frame
  dose
1  3.4
2  5.2
3  7.0
4  8.5
5 10.5
6 13.0
7 18.0
8 21.0
9 28.0
```

Figura 8: fit guarda la función glm aplicada a la tabla de la Figura 0. En la varibale pred.frame se añade un nuevo vector de datos: la dosis administrada.

```
> predict(fit, newdata=pred.frame)
          1           2           3           4           5           6           7           8
-1.78784889 -1.44053600 -1.09322311 -0.80379570 -0.41789249  0.06448652  1.02924455  1.60809937
          9
 2.95876061
> predict(fit, newdata=pred.frame, type="response")
        1         2         3         4         5         6         7         8         9
0.1433367 0.1914624 0.2510118 0.3092142 0.3970212 0.5161160 0.7367694 0.8331473 0.9506759
```

Figura 9: Estos números están en la escala logit (o log-odds). Para obtener los valores predichos en la escala respuesta (ósea en términos de probabilidades), se usa el argumento type="response":

Los valores predichos en la escala proporción también se podrían obtener usando la función extractora fitted(), pero está no permite obtener valores predichos para otros valores nuevos que no sea los observados.

El Código 1, muestra ejemplos de cómo utilizar los valores predichos para visualizar el comportamiento del modelos de ajuste.

3.2.1. Código 1

Para datos tabulares es obvio tratar de comparar las proporciones observados y las predichas por el modelo logístico, para los valores, correspondientes a los observados.

```
> plot(dosis, s/n, type="b", pch=16, ylim=c(0,1))
> title("Sample proportions of sypmtoms vs. dose")
> lines(x=dosis, y=predict(fit, newdata=pred.frame, type="response"), type="l", lty="dashed")
> lines(x=dosis, y=predict(fit2, newdata=log(pred.frame), type="response"), type="l", pch=3)
> legend ("bottomright", legend= c("empirical probabilities", "Model 1: dose original scale", "Model 2: do
se log scale"),
+         lty=c(0, 2, 1), cex=0.7, pch=c(16,NA, NA))
```

Figura 10: Entrada del código para ver la relación entre dosis y ratio.

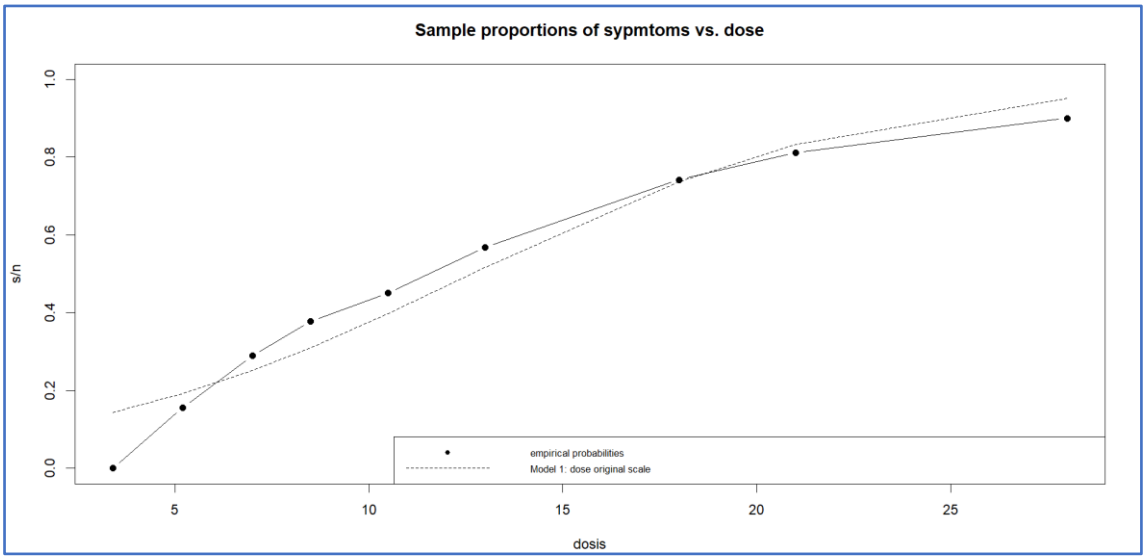

Figura 11: Gráfica que muestra la relación entre la dosis inyectada a los ratones y el ratio (ratones con síntomas entre ratones).

```
> # Sin embargo observamos que el modelo con la dosis en la escala original no se
> # ajusta del todo bien. ¿Que pasaría si log-transformamos la dosis?
> fit2 <- glm(mice.tbl~log(dosis),family=binomial("logit"))
> fit2

Call:  glm(formula = mice.tbl ~ log(dosis), family = binomial("logit"))

Coefficients:
(Intercept)   log(dosis)
     -5.791        2.396

Degrees of Freedom: 8 Total (i.e. Null);  7 Residual
Null Deviance:      116.8
Residual Deviance: 4.547        AIC: 38.11
```

```
> summary(fit2)

Call:
glm(formula = mice.tbl ~ log(dosis), family = binomial("logit"))

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.9190  -0.1930  -0.1183   0.3097   0.6311

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -5.7907     0.6839  -8.467   <2e-16 ***
log(dosis)    2.3964     0.2799   8.561   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 116.7501  on 8  degrees of freedom
Residual deviance:   4.5475  on 7  degrees of freedom
AIC: 38.115

Number of Fisher Scoring iterations: 4
```

Figura 12: se aplica la función logit para transformar logarítimicamente el eje de la dosis.

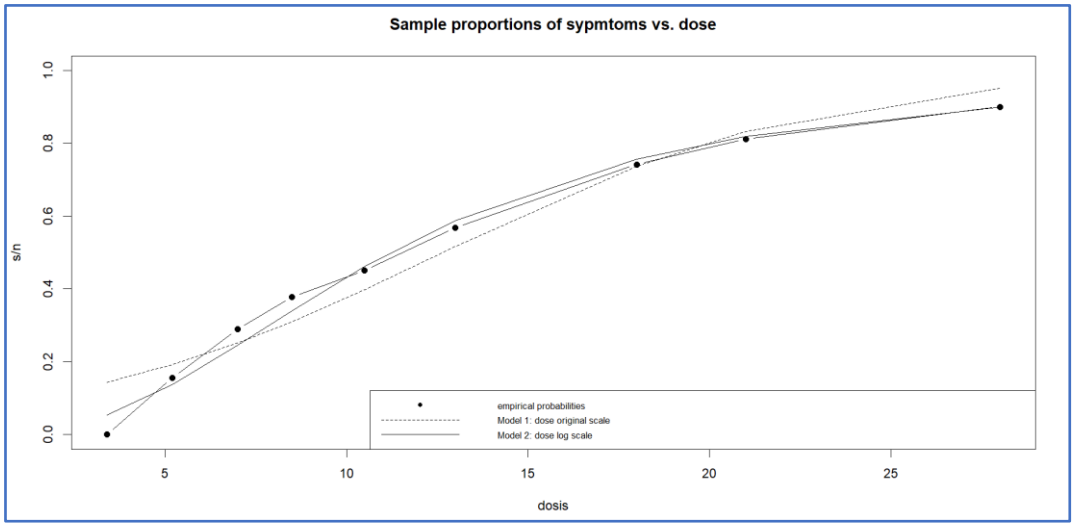

Figura 13: Gráfica que muestra la relación entre la dosis inyectada a los ratones y el ratio (ratones con síntomas entre ratones). Observar la línea continua (función logit), y compararla respecto a la discontinua. Es clara una mayor similitud respecto a la línea unida por los puntos.

```
> predict(fit, newdata=pred.frame, type="response")*n
        1         2         3         4         5         6         7         8         9
 4.730110  6.126795  9.538450 11.440924 15.880847 19.096294 22.839852 30.826452 28.520277
> predict(fit2, newdata=log(pred.frame), type="response")*n
        1         2         3         4         5         6         7         8         9
 1.790777  4.386153  9.295489 12.588708 18.444548 21.758240 23.464153 30.278973 26.992960
> #y para obtener una buena impresión para la comparación, puede usar
> data.frame(fit1=predict(fit, newdata=pred.frame, type="response")*n,fit2=predict(fit2, newdata=log(pred.
frame), type="response")*n,
+            s,n)
       fit1      fit2  s  n
1  4.730110  1.790777  0 33
2  6.126795  4.386153  5 32
3  9.538450  9.295489 11 38
4 11.440924 12.588708 14 37
5 15.880847 18.444548 18 40
6 19.096294 21.758240 21 37
7 22.839852 23.464153 23 31
8 30.826452 30.278973 30 37
9 28.520277 26.992960 27 30
```

Figura 14: Queda claro que el modelo 2 (fit2) se ajusta mejor a los datos

3.3. Selección de variables

Los modelos logísticos también se pueden utilizar con datos individuales (no agregados).

Para esto volveremos a utilizar la muestra adultos, no fumadores actuales, que participaron en la Encuesta Nacional de EEUU, que ya tuvimos ocasión de analizar en las Prácticas 3 y 4. En particular, estudiaremos la variable "Antecedentes de Enfermedad Cardiovascular" como la variable de respuesta. Esta variable indica para cada participante si ha tenido un evento cardiovascular (infarto de miocardio, accidente cerebrovascular o insuficiencia cardiaca) en el pasado, lo que se indica como 0= No, y 1=Sí. Cargaremos los datos como hicimos en Prácticas anteriores, y analizaremos la probabilidad de tener antecedentes cardiovasculares en función del estatus de ex-fumador como sigue:

```
> data <- read.csv("Mortality_NHANES8894_NonSmokers-1.csv")
> data
      X seqn race riagendr ridageyr smoking bmxbmi hbp highchol diab ckd    gfr.epi sedent prev.cvd
1     3    9    1        2       48       2   27.6   0        1    0   0 109.09726      1        0
2     4   11    3        1       48       2   25.0   0        1    1   0 101.95873      0        0
3    10   48    1        2       56       2   37.0   0        0    0   0  85.78298      0        0
4    11   49    1        2       82       1   19.1   1        0    1   1  54.70173      0        0
5    13   63    3        2       66       1   23.6   0        1    1   0  79.96383      0        0
6    14   78    1        2       80       1   26.8   0        0    1   0  82.87894      0        0
7    15   82    3        2       80       1   25.5   1        1    0   0  87.13443      1        0
8    17   86    1        2       83       1   26.7   1        1    0   1  48.45083      0        0
9    20   96    1        1       90       2   24.2   1        1    0   1  50.07860      1        1
10   21   97    3        2       86       1   26.4   1        0    0   1  53.18609      1        0
11   23  106    1        1       72       2   25.5   1        1    0   0  86.14030      0        0
12   25  116    1        1       84       2   25.5   1        0    0   1  57.50498      0        0
```

...

```
   prev.cancer peryr.exm.8yr peryr.age.8yr mortstat.8yr cancer.8yr heart.8yr
1            0      7.666667       55.66667           0          0         0
2            0      7.666667       55.66667           0          0         0
3            0      7.666667       63.66667           0          0         0
4            0      3.250000       85.25000           1          0         0
5            0      7.666667       73.66667           0          0         0
6            0      7.666667       87.66667           0          0         0
7            0      7.666667       87.66667           0          0         0
8            0      4.333333       87.33333           1          0         1
9            0      3.166667       93.16667           1          0         0
10           0      7.666667       93.66667           0          0         0
11           0      7.666667       79.66667           0          0         0
12           0      2.666667       86.66667           1          1         0
13           0      2.416667       71.41667           1          0         0
14           0      7.666667       70.66667           0          0         0
```

Figura 15: Se garda en data los datos guardados en el .csv.

```
> dim(data)
[1] 6195   20
> data <- data[complete.cases(data),]
```

```
> dim(data)
[1] 5929   20
> attach(data)
```

Figura 16: Nos aseguramos de quitarnos las filas con valores nulos.

```
> fit <- glm(prev.cvd~as.factor(smoking), family=binomial("logit"))
> fit

Call:  glm(formula = prev.cvd ~ as.factor(smoking), family = binomial("logit"))

Coefficients:
        (Intercept)  as.factor(smoking)2
            -2.0730                0.5099

Degrees of Freedom: 5928 Total (i.e. Null);  5927 Residual
Null Deviance:      4740
Residual Deviance: 4695         AIC: 4699
```

```
> summary(fit)

Call:
glm(formula = prev.cvd ~ as.factor(smoking), family = binomial("logit"))

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.6168  -0.6168  -0.4868  -0.4868   2.0936

Coefficients:
                    Estimate Std. Error z value Pr(>|z|)
(Intercept)         -2.07299    0.05380 -38.535  < 2e-16 ***
as.factor(smoking)2  0.50990    0.07581   6.726 1.74e-11 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 4739.7  on 5928  degrees of freedom
Residual deviance: 4694.5  on 5927  degrees of freedom
AIC: 4698.5

Number of Fisher Scoring iterations: 4
```

Figura 17: Aplicamos la función logit sobre la variable no agregada “antecedentes cardiovasculares”

La variable respuesta **“antecedentes cardiovasculares”** es **un factor con dos niveles**, donde el último nivel se considera el evento. **as.factor()**.

En este caso, **se sabe que la enfermedad cardiovascular** es una enfermedad multifactorial con **múltiples factores de riesgo reconocidos**. No podemos descartar que las personas que han fumado y lo ha dejado tenían muchos otros factores de riesgo simultáneamente que podrían explicar las asociaciones observadas para el status de ex-fumador.

En este contexto se ve la necesidad de tener en cuenta en los modelos posibles variables confusores que por definición implican: 1) son factores causales conocidos de la variable respuesta; 2) pueden estar asociados con la variable exposición, y 3) no son agentes intermedios en la asociación causal entre la exposición y la variable respuesta.

El Código 2, muestra ejemplos de cómo utilizar comandos que apoyan al proceso de selección de variables en el contexto de la regresión logística.

3.3.1. Código 2

```
> # Un experto cardiólogo nos indica que además del tabaco, hay otros factores de
> # riesgo cardiovascular establecidos: el sexo, la edad, el colesterol alto, la diabetes,
> # la enfermedad renal, el sedentarismo, y la hipertensión arterial.
> # Muchos de estos factores están relacionados con estilos de vida y por tanto
> # posiblemente correlacionados con el hábito tabáquico.
> # Para descartar que la asociación cruda entre tabaco y historia de enfermedad
> # cardiovascular no se explica por la confusión introducida por los otros,
> # factores de riesgo, procedemos ajustar un modelo de regresión logística que
> # incluye todos esos factores:
> fit <- glm(prev.cvd~as.factor(smoking)+as.factor(riagendr)+ridageyr+diab+ckd+sedent+hbp+highchol, family
=binomial("logit"))
> summary(fit)

Call:
glm(formula = prev.cvd ~ as.factor(smoking) + as.factor(riagendr) +
    ridageyr + diab + ckd + sedent + hbp + highchol, family = binomial("logit"))

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.6940  -0.5638  -0.3661  -0.2257   2.7520
```

Figura 18: Se incluyen todos los factores de riesgo en la variable fit.

```
Coefficients:
                      Estimate Std. Error z value Pr(>|z|)
(Intercept)          -6.014673   0.267870 -22.454  < 2e-16 ***
as.factor(smoking)2   0.372053   0.087684   4.243 2.20e-05 ***
as.factor(riagendr)2 -0.430174   0.090330  -4.762 1.91e-06 ***
ridageyr              0.050117   0.003571  14.033  < 2e-16 ***
diab                  0.704380   0.089310   7.887 3.10e-15 ***
ckd                   0.540409   0.097222   5.559 2.72e-08 ***
sedent                0.423809   0.083662   5.066 4.07e-07 ***
hbp                   0.582190   0.093242   6.244 4.27e-10 ***
highchol              0.094208   0.089770   1.049    0.294
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 4739.7  on 5928  degrees of freedom
Residual deviance: 4052.8  on 5920  degrees of freedom
AIC: 4070.8

Number of Fisher Scoring iterations: 5
```

Figura 19: Para cada factor de riesgo, se estiman desviación típica, error, valor z y p-valor.

```
> fit

Call:  glm(formula = prev.cvd ~ as.factor(smoking) + as.factor(riagendr) +
    ridageyr + diab + ckd + sedent + hbp + highchol, family = binomial("logit"))

Coefficients:
         (Intercept)   as.factor(smoking)2  as.factor(riagendr)2              ridageyr
            -6.01467               0.37205              -0.43017               0.05012
                diab                   ckd                sedent                   hbp
             0.70438               0.54041               0.42381               0.58219
            highchol
             0.09421

Degrees of Freedom: 5928 Total (i.e. Null);  5920 Residual
Null Deviance:      4740
Residual Deviance: 4053         AIC: 4071
```

```
> #Las tablas de desviación corresponden a las tablas ANOVA para regresión múltiple
> #análisis y se generan así con la función anova():
> anova(fit, test="Chisq")
Analysis of Deviance Table

Model: binomial, link: logit

Response: prev.cvd

Terms added sequentially (first to last)


                    Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
NULL                                 5928     4739.7
as.factor(smoking)   1    45.21       5927     4694.5 1.772e-11 ***
as.factor(riagendr)  1    12.80       5926     4681.7 0.0003462 ***
ridageyr             1   430.27       5925     4251.4 < 2.2e-16 ***
diab                 1    90.42       5924     4161.0 < 2.2e-16 ***
ckd                  1    38.81       5923     4122.2 4.672e-10 ***
sedent               1    26.69       5922     4095.5 2.391e-07 ***
hbp                  1    41.66       5921     4053.9 1.088e-10 ***
highchol             1     1.11       5920     4052.8 0.2922912
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
> # Note que la columna "Deviance" da diferencias entre modelos cuando las variables
> # se agregan al modelo secuencialmente.
> # Dichas desviaciones se distribuyen aproximadamente según una Ji cuadrado con los
> # grados de libertad indicados.
> #Es necesario agregar el argumento test="chisq" para obtener las pruebas de Ji cuadrado.
> #Dado que la variable de "high cholesterol" es la última en introducirse secuencialmente
> # en el modelos, es posible que no sea significativa porque no queda mucho margen
> # en la deviance para poder explicar, o hemos metido con anterioridad variables
> # relaciondas con high chol, que podrían, "robar" su efecto.
> # Sin embargo, si las variables se reorganizan de modo que high cholesterol viene
> # la primera, obtenemos una prueba basada en la desviación para ver si, esa variable
> # explica una parte de la desviación en ausencia de las otras variables:
> fit <- glm(prev.cvd~highchol+as.factor(smoking)+as.factor(riagendr)+ridageyr+bmxbmi+diab+ckd+sedent+hbp,
family=binomial("logit"))
```

```
> fit
```

```
Call:  glm(formula = prev.cvd ~ highchol + as.factor(smoking) + as.factor(riagendr) +
    ridageyr + bmxbmi + diab + ckd + sedent + hbp, family = binomial("logit"))

Coefficients:
         (Intercept)              highchol   as.factor(smoking)2  as.factor(riagendr)2
            -6.68768               0.09143               0.36578              -0.44786
            ridageyr                bmxbmi                  diab                   ckd
             0.05289               0.01904               0.66346               0.53639
              sedent                   hbp
             0.41399               0.55053

Degrees of Freedom: 5928 Total (i.e. Null);  5919 Residual
Null Deviance:      4740
Residual Deviance: 4047         AIC: 4067
```

```
> summary(fit)

Call:
glm(formula = prev.cvd ~ highchol + as.factor(smoking) + as.factor(riagendr) +
    ridageyr + bmxbmi + diab + ckd + sedent + hbp, family = binomial("logit"))

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.6469  -0.5678  -0.3656  -0.2214   2.7916

Coefficients:
                     Estimate Std. Error z value Pr(>|z|)
(Intercept)          -6.687680   0.391644 -17.076  < 2e-16 ***
highchol              0.091434   0.089807   1.018   0.3086
as.factor(smoking)2   0.365784   0.087758   4.168 3.07e-05 ***
as.factor(riagendr)2 -0.447861   0.090819  -4.931 8.16e-07 ***
ridageyr              0.052893   0.003779  13.997  < 2e-16 ***
bmxbmi                0.019040   0.007913   2.406   0.0161 *
diab                  0.663457   0.090871   7.301 2.85e-13 ***
ckd                   0.536389   0.097231   5.517 3.46e-08 ***
sedent                0.413986   0.083829   4.938 7.88e-07 ***
hbp                   0.550533   0.094153   5.847 5.00e-09 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 4739.7  on 5928  degrees of freedom
Residual deviance: 4047.1  on 5919  degrees of freedom
AIC: 4067.1
```

Figura 20: Las variables se han reorganizado, quedando high cholesterol la primera: la variable explica una parte de la desviación en ausencia de las otras variables.

```
> anova(fit, test="Chisq")
Analysis of Deviance Table

Model: binomial, link: logit

Response: prev.cvd

Terms added sequentially (first to last)


                    Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
NULL                                 5928     4739.7
highchol             1     2.13       5927     4737.6 0.1441646
as.factor(smoking)   1    44.99       5926     4692.6 1.984e-11 ***
as.factor(riagendr)  1    13.84       5925     4678.8 0.0001991 ***
ridageyr             1   429.30       5924     4249.4 < 2.2e-16 ***
bmxbmi               1    31.72       5923     4217.7 1.783e-08 ***
diab                 1    73.28       5922     4144.5 < 2.2e-16 ***
ckd                  1    36.98       5921     4107.5 1.193e-09 ***
sedent               1    24.86       5920     4082.6 6.170e-07 ***
hbp                  1    35.56       5919     4047.1 2.475e-09 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Figura 21: ANOVA es una fórmula estadística que se utiliza para comparar las varianzas entre las medias (o el promedio) de diferentes grupos

```
> # De esta salida se puede desprender que high chol es eliminable, mientras que las otras no.
> # También se podría re-configurar el orden de las otras variables explicativas para
> # ver su relevancia en ausencia de las otras.
> # Un método alternativo es usar drop1() para intentar eliminar un término a la vez:
> drop1(fit, test="Chisq")
Single term deletions

Model:
prev.cvd ~ highchol + as.factor(smoking) + as.factor(riagendr) +
    ridageyr + bmxbmi + diab + ckd + sedent + hbp
                    Df Deviance    AIC     LRT  Pr(>Chi)
<none>                   4047.1 4067.1
highchol             1   4048.1 4066.1   1.044     0.307
as.factor(smoking)   1   4064.4 4082.4  17.372 3.074e-05 ***
as.factor(riagendr)  1   4071.4 4089.4  24.390 7.867e-07 ***
ridageyr             1   4264.1 4282.1 217.052 < 2.2e-16 ***
bmxbmi               1   4052.8 4070.8   5.697     0.017 *
diab                 1   4098.5 4116.5  51.401 7.529e-13 ***
ckd                  1   4076.7 4094.7  29.599 5.314e-08 ***
sedent               1   4071.2 4089.2  24.182 8.763e-07 ***
hbp                  1   4082.6 4100.6  35.559 2.475e-09 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
> #Aquí LRT es la prueba de razón de verosimilitud, otro test para medir el cambio
> # en la desviación. La información en las tablas de desviación es fundamentalmente
> # la misma que la dada por las pruebas z en la tabla de coeficientes individuales de,
> # regresión porque en la práctica la diferencia entre ambos tests suele ser pequeña
> # debido a que en muestras grandes Ji cuadrado es igual a la distribución normal al
> # cuadrado en pruebas con un solo grado de libertad.
> # Sin embargo, para probar los factores con más de dos categorías, no hay más remedio
> # que usar la tabla de desviación ya que las pruebas z solo se relacionan con
> # coeficientes individuales y no con grupos de coeficientes, por ejemplo en el
> # contexto de variables, indicadores ("dummies") relacionadas con una variable
> # categórica de más de una categoría.
```

3.2. Análisis de supervivencia

El análisis de tiempos de supervivencia (es decir tiempo que transcurre hasta que un suceso ocurre) es un tema importante dentro de la biología y medicina en particular. Tales datos a menudo tienen una distribución muy sesgada (es decir, no siguen una distribución normal), de modo que la el uso de modelos lineales estándar es problemático. Los datos de supervivencia a menudo se censuran: no sabe el tiempo exacto de seguimiento hasta la ocurrencia del evento, solo que es más largo que un valor dado hasta el que se ha podido seguir al los individuos. Por ejemplo, en un ensayo de cáncer, algunas las personas se pierden durante el seguimiento o simplemente viven más allá del período de estudio. Es un error ignorar la censura en el análisis estadístico, a veces con consecuencias extremas, especialmente cuando la censura ocurre de una manera que es diferencial con respecto al evento de interés.

A continuación se definen conceptos esenciales para entender el análisis de supervivencia. Sea X el tiempo de supervivencia real y T el tiempo que transcurre hasta la censura. Lo que se observa es el mínimo de X y T junto con una indicación de si es uno o el otro. T puede ser una variable aleatoria o un tiempo fijo dependiendo de contexto, pero si es aleatorio, entonces generalmente no debería ser informativo para los métodos que describimos puedan ser aplicables. A veces “muerto de otras causas” se considera un evento de censura para la mortalidad de una determinada enfermedad, y en esos casos es particularmente importante asegurarse de que estas otras causas no están asociadas con el estado de la enfermedad. La función de supervivencia S(t) mide la probabilidad de estar vivo en un determinado tiempo. En realidad es solo 1 menos la función de distribución acumulada para X, es decir 1−F(t). La función de riesgo (”hazard”) h(t) mide la probabilidad (infinitesimal) de morir en un corto intervalo de tiempo t (instante), dado que el sujeto está vivo en el tiempo t. Si la distribución del tiempo de supervivencia tiene densidad f , entonces h(t) = f(t)/S(t). Esto a menudo se considera una cantidad más fundamental que la media o la mediana de la distribución de supervivencia y se utiliza como base para el modelado.

Para realizar el análisis de supervivencia en R cargaremos el paquete survival con library(survival). Las rutinas en supervivencia funcionan con objetos de clase ”Surv”, que es una estructura de datos que combina tiempos e información de censura. Tales objetos se construyen usando la función Surv(), que toma dos argumentos: un tiempo de seguimiento observado y un indicador de evento. Este último se puede codificar como un variable lógica, una variable 0/1 o una variable 1/2. La última codificación no es recomendado ya que Surv() asumirá la codificación 0/1 si todos los valores son 1. En realidad, Surv() también se puede usar con tres argumentos para tratar con datos que tienen un tiempo de inicio y un tiempo de finalización (”entrada escalonada”) y también datos censurados por intervalos (donde sabe que ocurrió un evento entre dos fechas, como sucede, por ejemplo, en la prueba repetida de una enfermedad).

A continuación seguiremos utilizando los datos de la Encuesta de Salud de EEUU que se utilizó con anterioridad. En este caso asumiremos que todos los tiempos de seguimiento empiezan a la vez, es decir el tiempo de origen 0, que correspondencia con el momento del examen físico que se realizó a todos los participantes, y que la escala temporal es ”años desde el examen físico”. En el análisis de regresión logística concluíamos que los ex-fumadores tenía una mayor probabilidad de tener antecedentes de enfermedad cardiovascular comparados con los que nunca fumaron, incluso teniendo en cuenta (ajustando) por otros factores de riesgo cardiovascular bien conocidos. Imagínese que un compañero epidemiológo le comenta que como el diseño del estudio es transversal (es decir, el estatus de fumador y la presencia/ausencia de antecedentes cardiovasculares, se determinaron en el mismo momento del tiempo), no se puede descartar que los ex-fumadores hayan dejado de fumar a consecuencia de tener enfermedad cardiovascular (y no al revés). En este momento se plantea realizar un estudio

prospectivo, donde el estatus de ex-fumador y variables de ajuste se midieron en las encuestas y examen físicos que se realizaron en 1999-2004, y la variable dependiente de interés es el tiempo que pasó hasta la ocurrencia de muerte por causa cardiovascular. El tiempo de seguimiento es de 8 años máximo (todas las personas que no murieron durante los 8 años después del examen físico son censurados). El primer paso en un análisis de supervivencia es definir el tiempo de supervivencia (variable respuesta) mediante Surv():

```
> Surv(peryr.exm.8yr, heart.8yr==1)[1:10]
 [1] 7.666667+ 7.666667+ 7.666667+ 3.250000+ 7.666667+ 7.666667+ 7.666667+ 4.333333  3.166667+ 7.666667+
```

Figura 22: Aplicación de la función surv

La variable heart.8yr es una variable indicadora del estatus de cada individuo al final del seguimiento: 1 significa “muerte por causa cardiovascular”, y 0 significa “muerte por causa no cardiovascular o censura (es decir vivo)”. La variable peryr.exm.8yr es el tiempo de seguimiento en años. Los nombres de las otras variables y su definición se pueden encontrar en el fichero ”Readme.txt”. Esta pieza de código muestra los 10 primeros valores creados por Surv(). Al pedir a R que nos muestre el objeto Surv se muestra se indica en pantalla, con un ‘+’ cuales son las observaciones censuradas. Por ejemplo, 7.666667+ significa que ese individuo no murió de causa cardiovascular dentro de los 8 años de seguimiento y luego se le dejo de observar, mientras que 4.333333 significa que el paciente murió a causa de la enfermedad cardiovascular poco más de 4 años después de que le hicieran el examen físico.

3.2.1. Descripción de datos de supervivencia

El estimador de Kaplan-Meier permite el cálculo de la curva de supervivencia, que es una función escalonada donde la supervivencia estimada se reduce por un factor (1−1/Rt) si hay una muerte en el tiempo t y una población de Rt que sigue viva y sin censura en ese momento. Se realiza el cálculo del estimador de Kaplan-Meier para la función de supervivencia con una función llamada survfit. En su forma más simple, solo se necesita definir un objeto Surv() para detallar los tiempos de seguimiento y el estatus al final del seguimiento. Alternativamente, el Log-Rank test se usa para evaluar si dos o más curvas de supervivencia son idénticas. Se basa en observar la población en cada tiempo t en el que ha ocurrido una muerte y calcular el número esperado de muertes en proporción al número de individuos a riesgo de morir en cada grupo. Esto luego se suma sobre todos los tiempos y se compara con el número observado de muertes por un procedimiento similar (pero no idéntico) a la prueba de χ 2 . El cálculo de la prueba Log-Rank se realiza mediante la función survdiff().

El código de la siguiente Figura ilustra como representar gráficamente la función de supervivencia, así como el uso de Surv() y survdiff().

```
# APARTADO A: Descripción de datos de supervivencia
# CÓDIGO 3: Descripción de datos de supervivencia
# Seguiremos con el ejemplo del estatus de ex-fumador y mortalidad cardiovascular
# en los participantes de la Encuesta Nacional de EEUU.
# Si queremos sólo una curva de supervivencia global para todos los datos:
km<- survfit(Surv(peryr.exm.8yr, heart.8yr==1)~1)
plot(km, ylim=c(0.90,1))
```

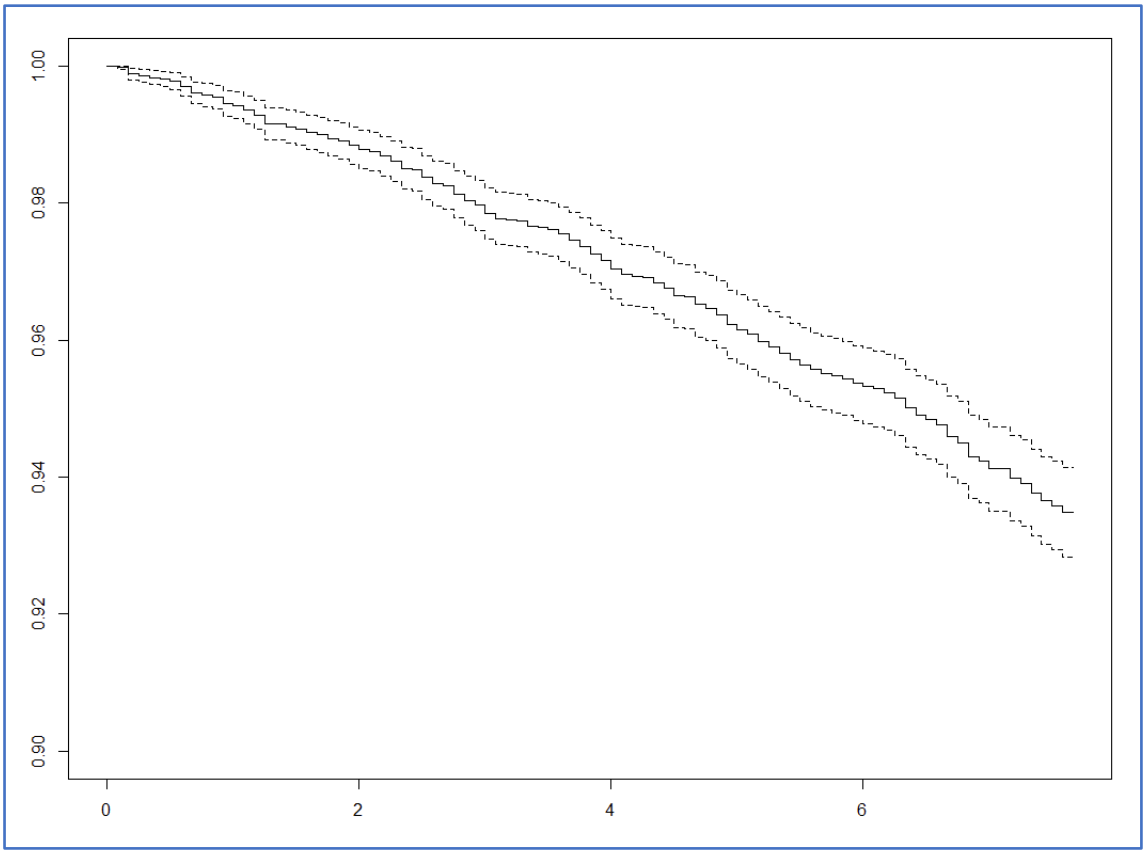

Figura 23: En gráfico, se muestra probabilidad de supervivencia respecto al tiempo.

Observar cómo la probabilidad de supervivencia disminuye a lo largo del tiempo.

Ahora, queremos representar por separado los individuos nunca fumadores y los exfumadores, utilizando “as.factor(smoking)” que permite la estimación por cada nivel del factor “smoking”.

```
# Si queremos una curva de supervivencia en nunca fumadores y ex-fumadores:
km<- survfit(Surv(peryr.exm.8yr, heart.8yr==1)~as.factor(smoking))
plot(km, ylim=c(0.90,1), lty=2:3)
legend("bottomright", c("Nunca Fum.", "Ex-Fum."), lty = 2:3)
```

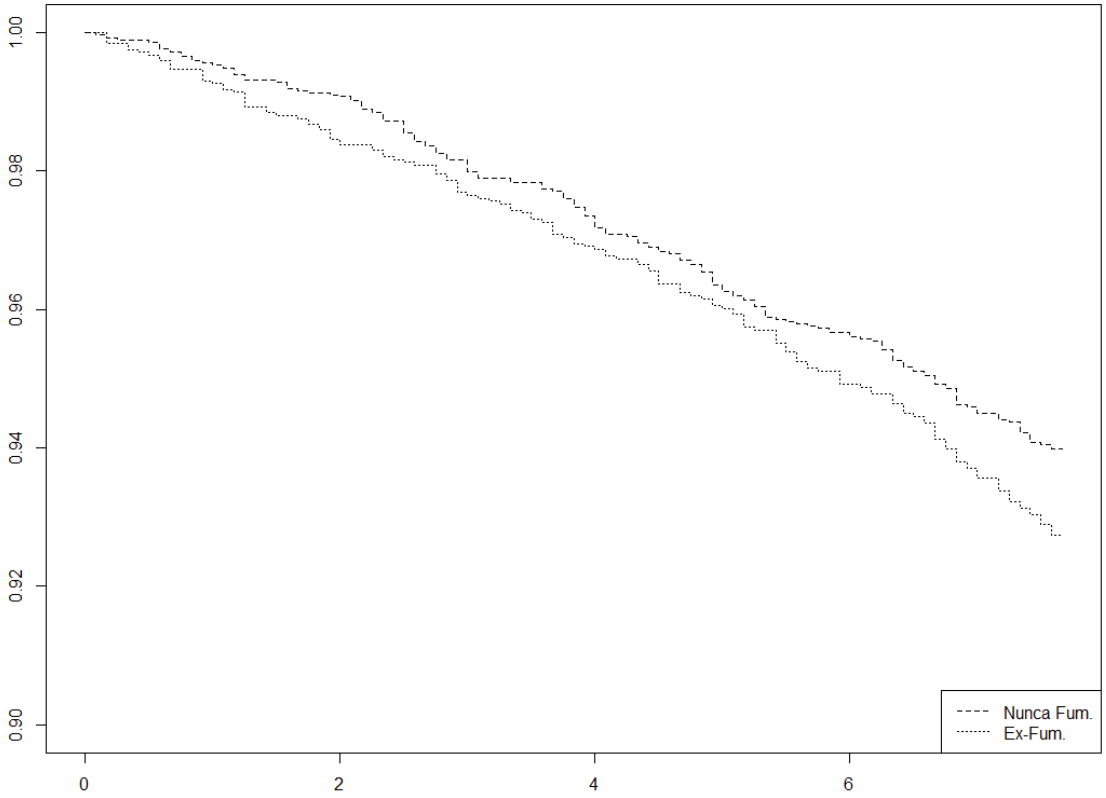

Figura 25: En este segundo gráfico podemos ver cómo, tanto como para fumadores como para exfumadores, la probabilidad de supervivencia disminuye a lo largo del tiempo.

Aún así, observar que la probabilidad de supervivencia de los individuos que nunca han fumado es siempre mayor a la de los exfumadores.

```
plot(km, ylim=c(0.90,1), lty=2:3, conf.int=T)
legend("bottomright", c("Nunca Fum.", "Ex-Fum."), lty = 2:3)
```

Figura 26: Se añaden los intervalos de confianza con el argumento "conf.int=T".

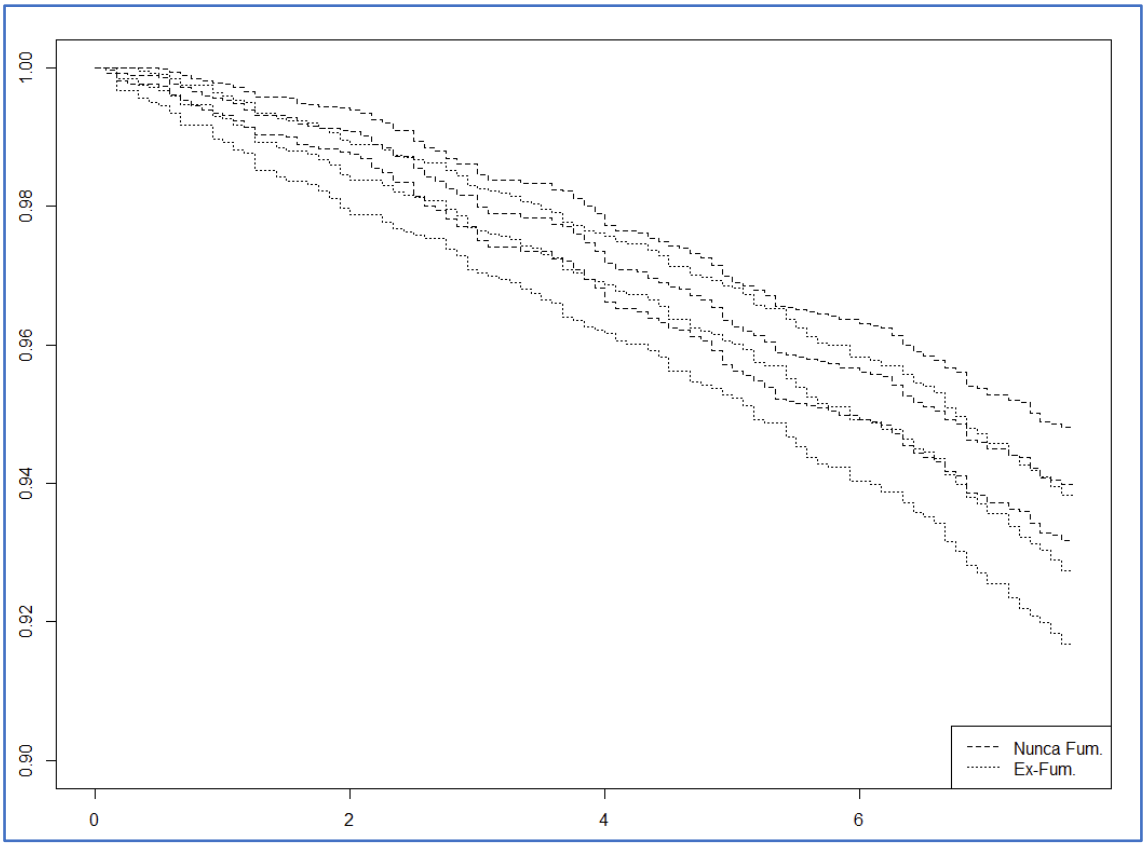

Figura 27: Esta última gráfica representa los estimadores de los mismos grupos de individuos, junto con sus respectivos intervalos de confianza. Dichos intervalos se representan como dos líneas más, una por encima y otra por debajo del grupo al que se refieren.

A continuación pedimos a R que muestre en pantalla el contenido del objeto y obtenemos que la salida de survfit() sólo nos devuelve el número de eventos.
Nótese que en este caso la mediana de supervivencia es NA, porque el comando sólo devuelve este valor cuando el 50% de los individuos han tenido el evento antes de la censura, lo cual no ocurre en este caso. El objeto survfit tiene mucha información escondida que se podría extraer. str() nos ayuda a ver el contenido del objeto:

```
> km
Call: survfit(formula = Surv(peryr.exm.8yr, heart.8yr == 1) ~ as.factor(smoking))

                         n events median 0.95LCL 0.95UCL
as.factor(smoking)=1 3481    198     NA      NA      NA
as.factor(smoking)=2 2448    163     NA      NA      NA
```

Figura 28: Salida de la función survfit.

Podemos ver que, en el primer grupo, de 3481 no fumadores, el número sucesos es de 198; mientras que en el grupo de los exfumadores, de 2448, se dan 163 muertes.

```
> str(km)
List of 17
 $ n        : int [1:2] 3481 2448
 $ time     : num [1:183] 0.0833 0.1667 0.25 0.3333 0.4167 ...
 $ n.risk   : num [1:183] 3481 3479 3477 3474 3472 ...
 $ n.event  : num [1:183] 1 2 1 0 0 1 3 2 2 2 ...
 $ n.censor : num [1:183] 1 0 2 2 5 2 4 0 1 5 ...
 $ surv     : num [1:183] 1 0.999 0.999 0.999 0.999 ...
 $ std.err  : num [1:183] 0.000287 0.000498 0.000575 0.000575 0.000575 ...
 $ cumhaz   : num [1:183] 0.000287 0.000862 0.00115 0.00115 0.00115 ...
 $ std.chaz : num [1:183] 0.000287 0.000498 0.000575 0.000575 0.000575 ...
 $ strata   : Named int [1:2] 92 91
  ..- attr(*, "names")= chr [1:2] "as.factor(smoking)=1" "as.factor(smoking)=2"
 $ type     : chr "right"
 $ logse    : logi TRUE
 $ conf.int : num 0.95
 $ conf.type: chr "log"
 $ lower    : num [1:183] 0.999 0.998 0.998 0.998 0.998 ...
 $ upper    : num [1:183] 1 1 1 1 1 ...
 $ call     : language survfit(formula = Surv(peryr.exm.8yr, heart.8yr == 1) ~ as.factor(smoking))
 - attr(*, "class")= chr "survfit"
```

Existe una función extractora para obtener los valores concretos de la función de supervivencia en cada tiempo que ha ocurrido al menos un evento.

```
> summary(km)
Call: survfit(formula = Surv(peryr.exm.8yr, heart.8yr == 1) ~ as.factor(smoking))

                as.factor(smoking)=1
  time n.risk n.event survival  std.err lower 95% CI upper 95% CI
0.0833   3481       1    1.000 0.000287        0.999        1.000
0.1667   3479       2    0.999 0.000497        0.998        1.000
0.2500   3477       1    0.999 0.000574        0.998        1.000
0.5000   3467       1    0.999 0.000642        0.997        1.000
0.5833   3464       3    0.998 0.000813        0.996        0.999
0.6667   3457       2    0.997 0.000909        0.995        0.999
0.7500   3455       2    0.997 0.000996        0.995        0.998
0.8333   3452       2    0.996 0.001076        0.994        0.998
0.9167   3445       1    0.996 0.001114        0.993        0.998
1.0000   3440       1    0.995 0.001150        0.993        0.998
1.0833   3437       2    0.995 0.001221        0.992        0.997
1.1667   3432       3    0.994 0.001319        0.991        0.997
1.2500   3426       3    0.993 0.001410        0.990        0.996
1.5000   3410       1    0.993 0.001439        0.990        0.996
...
                as.factor(smoking)=2
  time n.risk n.event survival  std.err lower 95% CI upper 95% CI
 0.167   2445       4    0.998 0.000817        0.997        1.000
 0.333   2437       2    0.998 0.001001        0.996        1.000
 0.417   2430       1    0.997 0.001082        0.995        0.999
 0.500   2427       1    0.997 0.001157        0.994        0.999
 0.583   2424       2    0.996 0.001294        0.993        0.998
 0.667   2419       3    0.995 0.001475        0.992        0.998
 0.917   2402       4    0.993 0.001690        0.990        0.996
 1.000   2394       1    0.993 0.001739        0.989        0.996
 1.083   2388       2    0.992 0.001834        0.988        0.995
 1.167   2383       1    0.991 0.001880        0.988        0.995
 1.250   2380       5    0.989 0.002094        0.985        0.993
 1.417   2364       2    0.988 0.002174        0.984        0.993
 1.500   2359       1    0.988 0.002213        0.984        0.992
 1.667   2347       1    0.988 0.002252        0.983        0.992
```

Figura 29: En estos fragmentos de salidas vemos los valores de cada uno de los los elementos de "km" en función del tiempo.

La estimación de Kaplan-Meier en esta salida es la función escalonada cuyos puntos de salto ocurren en "tiempo" y cuyos valores justo después de un salto están dados la columna "survival".
A continuación vamos a realizar el long-rank test para obtener un p-valor que nos dará la probabilidad de que las curvas de supervivencia sean distintas en nunca fumadores y ex-fumadores sean distintas por azar, asumiendo que hipótesis nula de que las curvas de supervivencia son iguales sea cierta.

```
> survdiff(Surv(peryr.exm.8yr, heart.8yr==1)~as.factor(smoking))
Call:
survdiff(formula = Surv(peryr.exm.8yr, heart.8yr == 1) ~ as.factor(smoking))

                        N Observed Expected (O-E)^2/E (O-E)^2/V
as.factor(smoking)=1 3481      198      215      1.33      3.28
as.factor(smoking)=2 2448      163      146      1.95      3.28

 Chisq= 3.3  on 1 degrees of freedom, p= 0.07
```

3.2.2. Modelo de riesgos proporcionales de Cox

El modelo de riesgos proporcionales permite el análisis de datos de supervivencia por modelosde regresión similares a los de lm y glm. El modelo de regresión de Cox la función de riesgo (hazard) se construye como:

$$h_i(t) = h_0(t)\ \exp(\beta_1 x_{i1} + ... + \beta_p x_{ip})$$

Este modelo se llama ”semiparamétrico” ya que la función de riesgo base o de referencia h0(t) puede tomar cualquier forma. Si hay dos perfiles de riesgo (P R) diferentes:

$$PR_i = \beta_1 x_{i1} + ... + \beta_p x_{ip}$$
$$PR_k = \beta_1 x_{k1} + ... + \beta_p x_{kp}$$

La razón de riesgo de un perfil comparado con el otro es:

$$\frac{h_i(t)}{h_k(t)} = \frac{h_0(t)exp(PR_i)}{h_0(t)exp(PR_k)} = \frac{exp(PR_i)}{exp(PR_k)}$$

por tanto, dicho cociente no depende del tiempo, y por ello a los modelos de Cox se les denomina de riesgo proporcionales. Como primer ejemplo, consideremos un modelo con el estatus de fumador como única variable explicativa:

```
library(survival)
fit <- coxph(Surv(peryr.exm.8yr, heart.8yr)~as.factor(smoking), subset(data, prev.cvd==0))
summary(fit)
```

```
> library(survival)
> fit <- coxph(Surv(peryr.exm.8yr, heart.8yr)~as.factor(smoking), subset(data, prev.cvd==0))
> summary(fit)
Call:
coxph(formula = Surv(peryr.exm.8yr, heart.8yr) ~ as.factor(smoking),
    data = subset(data, prev.cvd == 0))

  n= 5116, number of events= 206

                        coef exp(coef) se(coef)      z Pr(>|z|)
as.factor(smoking)2 -0.05233   0.94902  0.14364 -0.364    0.716

                    exp(coef) exp(-coef) lower .95 upper .95
as.factor(smoking)2     0.949      1.054    0.7162     1.258

Concordance= 0.507  (se = 0.017 )
Likelihood ratio test= 0.13  on 1 df,   p=0.7
Wald test            = 0.13  on 1 df,   p=0.7
Score (logrank) test = 0.13  on 1 df,   p=0.7
```

Nótese que se ha excluido a individuos con antecedentes de enfermedad cardiovascular con la opción subset(data, prev.cvd==0). Es importante tener en cuenta que para que el análisis de supervivencia sea correcto sólo puede incluirse en el análisis tiempo de seguimiento en el que se está ”a riesgo”. Técnicamente hablando una persona ”viva” está a riesgo de morir de enfermedad cardiovascular. Sin embargo se podría argumentar que aquellos que ya tienen antecedentes podrían tener un riesgo basal más elevado que los que no, y por tanto en este caso se ha decidido excluirlos. La interpretación de la salida sería: coef es el logaritmo estimado de la razón de riesgo (”hazard”) entre los dos grupos (nunca y ex-fumadores). Por conveniencia también se da como la relación de riesgo (”hazard ratio”) obtenida al exponenciar dicho coeficiente (exp(coef)). La línea que sigue también da la relación invertida (intercambio los grupos) y los intervalos de confianza para la razón de riesgo. Por fin, se dan tres pruebas generales para efectos significativos en el modelo. Estos son todos equivalentes en muestras grandes pero pueden diferir algo en muestras pequeñas.

Un ejemplo más elaborado, permitiría incluir a individuos con y sin antecedentes cardiovasculares, y permitir que el modelo asumiera un riesgo basal diferencial:

```
fit <- coxph(Surv(peryr.exm.8yr, heart.8yr)~as.factor(smoking)+strata(as.factor(prev.cvd)))
summary(fit)
```

```
> fit <- coxph(Surv(peryr.exm.8yr, heart.8yr)~as.factor(smoking)+strata(as.factor(prev.cvd)))
> summary(fit)
Call:
coxph(formula = Surv(peryr.exm.8yr, heart.8yr) ~ as.factor(smoking) +
    strata(as.factor(prev.cvd)))

  n= 5929, number of events= 361

                      coef exp(coef) se(coef)     z Pr(>|z|)
as.factor(smoking)2 0.0391    1.0399   0.1065 0.367    0.714

                    exp(coef) exp(-coef) lower .95 upper .95
as.factor(smoking)2      1.04     0.9617     0.844     1.281

Concordance= 0.496  (se = 0.015 )
Likelihood ratio test= 0.13  on 1 df,   p=0.7
Wald test            = 0.13  on 1 df,   p=0.7
Score (logrank) test = 0.13  on 1 df,   p=0.7
```

Figura 30: En esta salida se tiene en cuenta el riesgo basal gracias a la inclusión de individuos con o sin antecedentes cardiovasculares.

Por supuesto, el modelo de regresión permite incorporar ajuste por otros factores de riesgo:

```
fit <- coxph(Surv(peryr.exm.8yr, heart.8yr)~as.factor(smoking)+as.factor(riagendr)+ridageyr+
               bmxbmi+highchol+diab+ckd+sedent+hbp+strata(as.factor(prev.cvd)))
summary(fit)
```

```
> fit <- coxph(Surv(peryr.exm.8yr, heart.8yr)~as.factor(smoking)+as.factor(riagendr)+ridageyr+
+               bmxbmi+highchol+diab+ckd+sedent+hbp+strata(as.factor(prev.cvd)))
> summary(fit)
Call:
coxph(formula = Surv(peryr.exm.8yr, heart.8yr) ~ as.factor(smoking) +
    as.factor(riagendr) + ridageyr + bmxbmi + highchol + diab +
    ckd + sedent + hbp + strata(as.factor(prev.cvd)))

  n= 5929, number of events= 361

                          coef exp(coef)  se(coef)      z Pr(>|z|)
as.factor(smoking)2  -0.098549  0.906151  0.117783 -0.837  0.40276
as.factor(riagendr)2 -0.607241  0.544852  0.120717 -5.030 4.90e-07 ***
ridageyr              0.098924  1.103982  0.006541 15.125  < 2e-16 ***
bmxbmi               -0.035788  0.964845  0.012298 -2.910  0.00361 **
highchol              0.043477  1.044436  0.114619  0.379  0.70445
diab                  0.697491  2.008706  0.115882  6.019 1.76e-09 ***
ckd                   0.549228  1.731916  0.112930  4.863 1.15e-06 ***
sedent                0.457373  1.579918  0.109121  4.191 2.77e-05 ***
hbp                   0.325227  1.384344  0.130521  2.492  0.01271 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

                     exp(coef) exp(-coef) lower .95 upper .95
as.factor(smoking)2     0.9062     1.1036    0.7194    1.1415
as.factor(riagendr)2    0.5449     1.8354    0.4301    0.6903
ridageyr                1.1040     0.9058    1.0899    1.1182
bmxbmi                  0.9648     1.0364    0.9419    0.9884
highchol                1.0444     0.9575    0.8343    1.3075
diab                    2.0087     0.4978    1.6006    2.5209
ckd                     1.7319     0.5774    1.3880    2.1610
sedent                  1.5799     0.6329    1.2757    1.9567
hbp                     1.3843     0.7224    1.0719    1.7879

Concordance= 0.879  (se = 0.008 )
Likelihood ratio test= 595.1  on 9 df,   p=<2e-16
Wald test            = 474.5  on 9 df,   p=<2e-16
Score (logrank) test = 587.9  on 9 df,   p=<2e-16
```

El modelo de Cox no estima un intercepto como en el caso de la regresión lineal, pero permite una función de riesgo de basal h0(t) con una curva de supervivencia correspondiente. En un análisis estratificado, habrá una curva de supervivencia basal para cada estrato. Se pueden extraer utilizando survfit() a partir de la la salida de coxph() y, por supuesto, utilizar el método plot para objetos de supervivencia. Ejecutar plot(survfit(fit), ylim = c(0.95,1)) nos mostrará el riesgo basal (”Baseline hazard”) en individuos con y sin antecedentes de enfermedad cardiovascular (ver Figura 31). Hay que tener en cuenta que el valor predeterminado para survfit() es generar curvas para un pseudoindividuos en los que las covariables están fijadas en sus valores medios.

```
plot(survfit(fit), ylim = c(0.95,1))
```

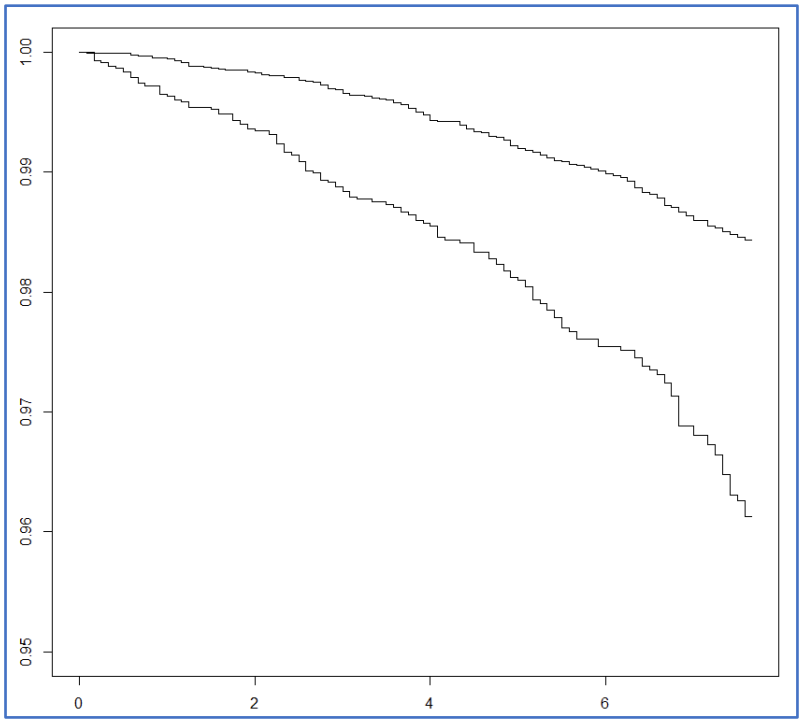

Figura 31: Hay menor probabilidad de supervivencia en personas con antecedentes cardiovasculares que en personas sin antecedentes.

La interpretación del coeficiente exponenciado sería: riesgo relativo de que ocurra el evento de interés cuando la variable explicativa continua aumenta una unidad (en caso de las variables categóricas, comparando con la referencia), ajustando por (manteniendo constantes) las demás variables explicativas. En este caso, el riesgo de morir por causa cardiovascular es 9% más bajo en ex-fumadores comparando con nunca fumadores, ajustando por todos los otros factores de riesgo cardiovascular, aunque el intervalo de confianza incluye el valor nulo. El tabaquismo activo es un factor de riesgo cardiovascular reconocido. La asociación inversa atribuida al hecho de dejar de fumar podría tener que ver con que ser ex-fumador se relaciona con otros estilos de vida saludables. Alternativamente, si ejecutamos el código: plot(cox.zph(fit)[1]) visualizaremos un tipo de residuos específico para la regresión de Cox (”Residuos de Schoenfeld”, donde el índice [1] indica que nos enseñe los residuos correspondientes al primer coeficiente del modelo en este caso) que nos ayuda a apreciar visualmente la asunción de proporcionalidad que da nombre al modelo de riesgos proporcionales de Cox (ver Figura 32).

Si se cumple la hipótesis de riesgos proporcionales los residuos debieran agruparse de forma aleatoria a ambos lados del valor 0 del eje Y, y la curva ajustada debería ser próxima a una línea recta. En este caso vemos como los residuos tienen una tendencia decreciente hasta aproximadamente los 4 años. Esto es consistente con estudios que sugieren que abandonar el hábito tabáquico produce un beneficio rápido en términos de riesgo de desarrollar eventos cardiovasculares (riesgo asociado al tabaquismo activo comienza a disminuir inmediatamente después de dejar de fumar, hasta que se iguala en 3-4 años con el de los nunca fumadores). Ésto es un incentivo para dejar de fumar, pero al mismo tiempo una indicación clara de que, en el caso del coeficiente para el status de ex-fumador en el modelo, la asunción de riesgos proporcionales no se cumple, ya que el riesgo asociado a dejar de fumar cambia en el tiempo.

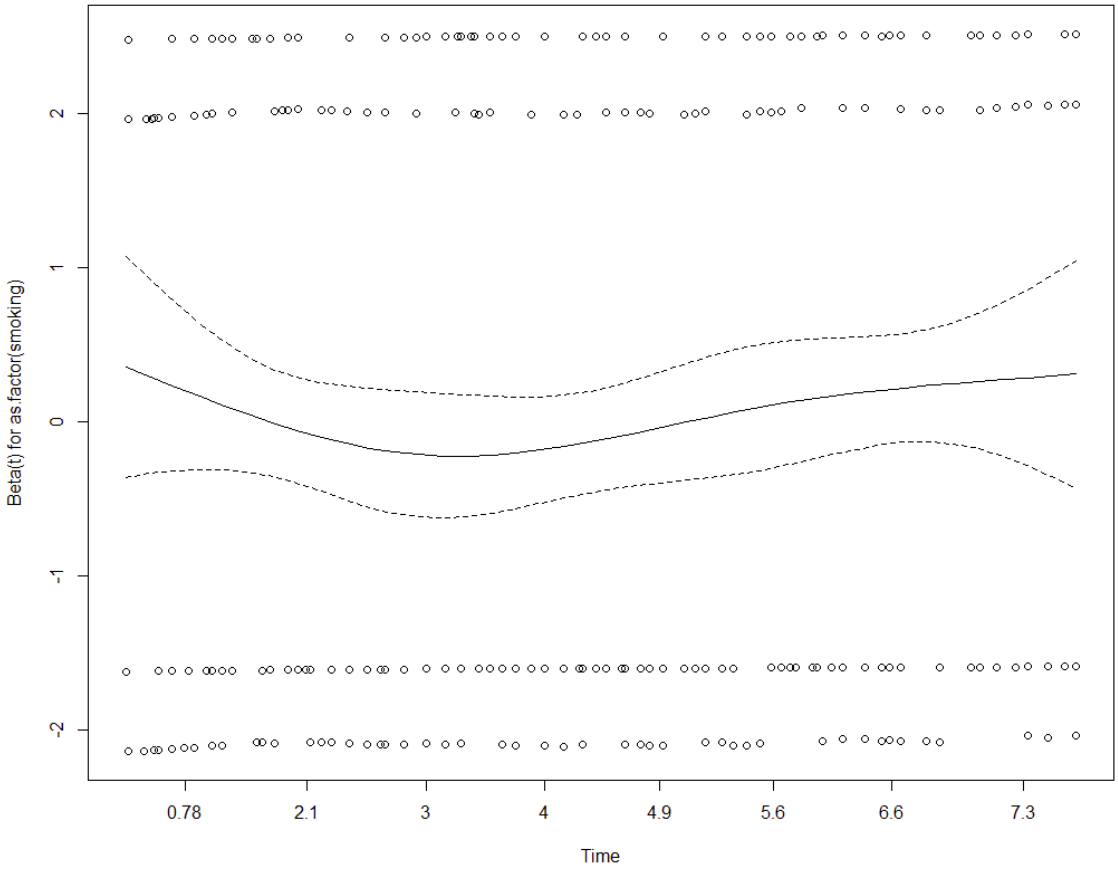

Figura 32: Residuos de Schoenfeld para visualizar la asunción de riesgos proporcionales asociada al coeficiente de estatus de ex-fumador

Desafortunadamente, para otras enfermedades como el cáncer, el riesgo residual más elevado de los ex-fumadores comparando con los nunca fumadores puede durar hasta 14 años. A pesar de que en los datos de esta práctica sólo tenemos 8 años de seguimiento y no estamos teniendo en cuenta el tiempo desde que se dejó de fumar, se puede intuir claramente en los datos de la Encuesta Nacional Americana que los ex-fumadores tienen casi un 40% mayor riesgo de morir por cáncer que los nunca fumadores, lo que apoya que nunca se debe empezar a fumar.

```
fit <- coxph(Surv(peryr.exm.8yr, cancer.8yr)~as.factor(smoking)+as.factor(riagendr)+ridageyr,
             subset(data, prev.cancer == 0))
summary(fit)
```

```
> fit <- coxph(Surv(peryr.exm.8yr, cancer.8yr)~as.factor(smoking)+as.factor(riagendr)+ridageyr,
subset(data, prev.cancer == 0))
> summary(fit)
Call:
coxph(formula = Surv(peryr.exm.8yr, cancer.8yr) ~ as.factor(smoking) +
    as.factor(riagendr) + ridageyr, data = subset(data, prev.cancer ==
    0))

  n= 5548, number of events= 169

                          coef exp(coef)  se(coef)      z Pr(>|z|)
as.factor(smoking)2   0.312712  1.367128  0.167155  1.871   0.0614 .
as.factor(riagendr)2 -0.263237  0.768560  0.167111 -1.575   0.1152
ridageyr              0.071103  1.073692  0.006668 10.663   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

                     exp(coef) exp(-coef) lower .95 upper .95
as.factor(smoking)2     1.3671     0.7315    0.9852     1.897
as.factor(riagendr)2    0.7686     1.3011    0.5539     1.066
ridageyr                1.0737     0.9314    1.0598     1.088

Concordance= 0.758  (se = 0.017 )
Likelihood ratio test= 142.6  on 3 df,   p=<2e-16
Wald test            = 120.9  on 3 df,   p=<2e-16
Score (logrank) test = 137.2  on 3 df,   p=<2e-16
```