



Universidad Nacional de Rosario
Facultad de Ciencias Económicas y Estadística
Lic. en Estadística

Estimación de funciones y pruebas no paramétricas para datos de supervivencia utilizando R

Material elaborado por la cátedra: Análisis de Datos de Supervivencia

Autoras: Harvey G.; Arnesi N.; Boggio G.

Año: 2019

Índice

1. Introducción2
2. Presentación del problema2
3. Aspectos específicos del software R3
4. Procedimientos no paramétricos para datos de supervivencia en R6
5. Estimación de la función de supervivencia6
 - 5.1 Método de Kaplan-Meier6
 - 5.2 Método de Nelson-Aalen11
 - 5.3 Método de Tablas de Vida14
6. Estimación de la función hazard a través del método de Nelson-Aalen19
7. Comparación de curvas de supervivencia estimadas23
 - 7.1 Comparación de dos grupos23
 - 7.2 Test log-rank de tendencia27
 - 7.3 Test estratificado28
8. Referencias bibliográficas30

1. Introducción

El presente material se elabora con el objetivo de exponer las principales herramientas para llevar a cabo un análisis de supervivencia básico mediante del software libre **R**¹. Es importante mencionar que por las características propias de dicho software, en forma permanente se incorporan nuevos paquetes y opciones; en esta oportunidad se muestra una forma de analizar los datos de supervivencia pero queda propuesto al alumno explorar otros códigos.

A fin de ejemplificar los distintos procedimientos se presenta el análisis de un problema real.

En cada uno de los análisis realizados se presentan los códigos utilizados junto con una breve descripción y la correspondiente salida. Además, con el fin de comparar con lo visto durante el desarrollo de las clases prácticas, se presentan las sentencias y las salidas obtenidas utilizando SAS. Se sugiere trabajar con el editor R Studio de manera de que resulte más amigable la programación.

2. Presentación del problema

Se cuenta con una base de datos con información de 84 pacientes con enfisema que recibieron trasplante pulmonar (TP) entre junio de 1994 y marzo de 2012. Se registraron características demográficas para cada paciente, entre las cuales se destacan:

- **Sexo:** género del paciente, con niveles: hombre y mujer.
- **Edad:** edad a la que se realizó el TP, expresada en años.

Además, se registró el **Estado clínico** de los pacientes el cual indica si fueron trasplantados bajo el estado de urgencia/emergencia o si lo hicieron en forma electiva.

Como variable respuesta se considera el tiempo transcurrido (expresado en años) desde el TP hasta el fallecimiento del paciente o hasta la finalización del estudio (**Tiempo**). La variable que indica si el paciente falleció o si se trata de una observación censurada se denomina **evento** y presenta los códigos 0 para las censuras y 1 para los eventos.

Resulta de interés evaluar el efecto de las variables demográficas mencionadas sobre la probabilidad de sobrevivir más allá de cierto tiempo excluyendo el primer año pos-trasplante. Los datos correspondientes se encuentran en un archivo con formato delimitado por comas, bajo el

1R Core Team (2013). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria.URL <http://www.R-project.org/>.

nombre EPOC.csv. En la sección siguiente se explica cómo se realiza la lectura y visualización de este tipo de archivos.

3. Aspectos específicos del software R

R es un lenguaje orientado a objetos, lo que significa que las variables, los datos, las funciones, los resultados, etc. se guardan en la memoria activa en forma de objetos con nombres específicos. El usuario puede modificar y manipular estos objetos con diversos operadores (aritméticos, lógicos y comparativos) y funciones (que son, a su vez, objetos). Los objetos pueden ser creados con el operador de asignación, el cual se denota con el signo menos (-) y el símbolo de menor (<) o mayor (>), dependiendo de la dirección en que se asigna el objeto.

En lo referido a la ayuda, el lenguaje R ofrece varios niveles de ayuda y éstas pueden activarse a través de la línea de comandos o a través del menú **Help**. A través de la línea de comandos pueden utilizarse las instrucciones:

- `help("tema")`, para obtener ayuda acerca del tópico específico, el inconveniente de esta instrucción es que hay que colocar exactamente el nombre del tema,
- `help.search("tema")`, es más flexible porque realiza una búsqueda del tema en todas los paquetes instalados.

Un aspecto que debe considerarse cuando se trabaja en R es que el lenguaje diferencia entre las mayúsculas y las minúsculas, por lo que cada comando u objeto debe ser escrito de manera exacta.

Si se desea incluir un comentario dentro de un código de R, el mismo debe iniciarse con el símbolo numeral (#). Todo lo que aparece en la línea de comando a la derecha de dicho símbolo es tomado como un comentario y, por lo tanto, no interfiere en la programación.

3.1. Lectura de datos

En R se pueden leer datos que se encuentren guardados bajo diversos formatos. Por ejemplo, los datos pueden estar almacenados en una hoja de cálculo Excel, en un archivo de datos de SPSS, en un archivo de texto, en un archivo donde los datos se encuentran delimitados por espacios o por comas, entre otros.

Los datos de la aplicación se encuentran guardados en un archivo con extensión csv, del inglés *commaseparatedvalues*, (EPOC.csv). Para su lectura y visualización debe ejecutarse el siguiente código.

Código 3.1

```
setwd("...:/.../carpeta_de_trabajo")
# establece el directorio de trabajo, donde estarán
# guardados los archivos a utilizar.

Base_EPOC<-read.csv('EPOC.csv')
# lee un conjunto de datos a partir de un
# archivo delimitado por comas y lo guarda en un
# objeto de R bajo el nombre asignado por el
# usuario, en este caso Base_EPOC, guardado en la
# carpeta de trabajo correspondiente.

Base_EPOC
# permite visualizar el contenido del objeto
# Base_EPOC, es decir, los datos del archivo.

attach(Base_EPOC)
# indica que todas las variables mencionadas
# en los sucesivos códigos se encuentran en
# el objeto Base_EPOC. De no utilizarse esta
# sentencia, cada vez que se nombra una
# variable debería indicarse como:
# Base_EPOC$nombre_variable. Esta función se
# anula con el comando detach(nombre_objeto).
```

La salida correspondiente a la ejecución de este código es la Salida 3.1.

Salida 3.1

Se muestran los primeros 15 valores.

	id	Edad	sexo	evento	tiempo	estado_clinico
1	1	45	F	0	0.901369863	Electivo
2	2	50	F	1	0.002739726	Electivo
3	3	51	M	0	0.904109589	Electivo
4	5	52	F	1	3.745205479	Electivo
5	6	55	M	1	4.791780822	Electivo
6	7	53	F	1	1.712328767	Electivo
7	8	58	F	1	0.021917808	Electivo
8	9	61	F	1	0.273972603	Electivo
9	10	59	F	1	0.005479452	Electivo
10	11	61	M	1	3.994520548	Electivo
11	12	49	M	1	0.128767123	Electivo
12	13	56	M	1	9.504109589	Electivo
13	14	43	M	0	1.339726027	Electivo
14	15	42	F	0	9.356164384	Electivo
15	16	62	M	0	5.512328767	Urgencia/Emergencia

...

3.2. Instalación y carga de paquetes

Al descargar R, éste incluye características y “paquetes” básicos pero existe una gran cantidad de paquetes (también llamados librerías) que se deben agregar para poder ejecutar funciones particulares.

Utilizando el comando `library()` se obtiene la lista de los paquetes que ya están instalados. Si no está disponible el paquete que se pretende utilizar, la instalación del mismo puede hacerse de dos maneras: mediante el menú, siguiendo las instrucciones que se encuentran en: Packages→Install, o bien mediante códigos utilizando el comando:

```
install.packages("nombre_del_paquete") # no omitir comillas.
```

Una vez que el paquete está instalado, debe cargarse para poder utilizar sus funciones y datos. Al igual que la instalación, la carga puede realizarse desde el menú o mediante el comando:

```
library(nombre_del_paquete)
```

4. Procedimientos no paramétricos para datos de supervivencia

Existen varios paquetes en R que permiten llevar a cabo análisis de supervivencia. Entre los más ampliamente utilizados se encuentran: `survival` y `KMsurv`.

Las rutinas en la librería `survival` trabajan con objetos contruidos usando la función `Surv`, la cual crea un objeto que combina la información del tiempo en el que se producen los eventos o el tiempo de censura junto con la información referida al estado de cada observación (evento o censura).

Para el ejemplo bajo estudio, ejecutando el código 4.1 se obtiene la salida 4.1 donde se muestra el contenido de dicho objeto.

Código 4.1

```
library(survival)

Surv(tiempo, evento==1) # muestra los tiempos en los cuales se dieron
                        # los eventos (en este caso codificados como 1)
                        # y las censuras. Los tiempos censurados se
                        # representan en la salida con el signo +.
                        # puede omitirse la especificación "=="
```

Salida 4.1

Se muestran los primeros 10 valores.

```
0.901369863+ 0.002739726 0.904109589+ 3.745205479 4.791780822
1.712328767 0.021917808 0.273972603 0.005479452 3.994520548
...
```

5. Estimación de la función de supervivencia

5.1 Método de Kaplan-Meier

El estimador de Kaplan-Meier utiliza información de todas las observaciones disponibles (censuradas y no censuradas) considerando la supervivencia en un momento dado como una serie de “pasos” definidos por los tiempos de supervivencia observados y censurados. Se supone que hay n individuos con tiempos de supervivencia observados $t_1, t_2, \dots, t_n$; algunos de estas observaciones pueden ser censuradas por derecha y además puede haber más de un individuo con el mismo tiempo de supervivencia observado. Por lo tanto, se supone que hay r tiempos distintos en los cuales se producen los eventos ($r \leq n$). Los tiempos de eventos se consideran ordenados en forma ascendente;

se indica con $t_{(j)}$ el j -ésimo tiempo ordenado con $j = 1, 2, \dots, r$. De esta forma se tiene que: $t_{(1)} < \dots < t_{(j)} < \dots < t_{(r)}$.

El estimador de la función de supervivencia $\hat{S}(t)$ se obtiene luego de definir los intervalos de manera que cada uno comience en un tiempo observado y termine en el momento anterior al próximo tiempo de evento ordenado, de forma que contenga un único tiempo de evento. Así, se tiene:

$$\hat{S}(t) = \prod_{j=1}^k \frac{n_j - d_j}{n_j} \quad k = 1, 2, \dots, r \text{ y } t_{(k)} \leq t < t_{(k+1)}$$

donde d_j representa el número de eventos en el momento $t_{(j)}$ y n_j el número de sujetos en riesgo justo antes del momento $t_{(j)}$. Notar que $\hat{S}(t) = 1$ para $t < t_{(1)}$ y $t_{(r+1)}$ se considera ∞ .

La función de supervivencia a través del método de Kaplan-Meier puede obtenerse mediante la función `survfit`, tal como se muestra en el código 5.1. Notar que la función se aplica a lo que se denomina un “objeto fórmula”, `Surv(tiempo, evento) ~ 1`, el cual tiene un objeto `Surv` como la respuesta a la izquierda del operador `~`. Una curva de supervivencia simple (sin distinguir entre niveles de ninguna covariable) se obtiene utilizando `~ 1`.

Código 5.1

```
KM.surv<-survfit(Surv(tiempo,evento==1)~1, type="kaplan-meier",
  conf.int=.95, conf.type='log-log', error="greenwood")
  # crea y asigna al objeto KM.surv, la
  # función de supervivencia utilizando el método
  # de Kaplan-Meier con sus respectivos intervalos de
  # confianza del 95%.
  # La opción conf.type permite indicar la forma
  # de cálculo de los intervalos de confianza. En
  # este caso se elige log(-log(survival)).
  # La opción error permite especificar la forma de
  # cálculo del error estándar de la estimación.
  # En este caso tanto "type" como "error" pueden
  # omitirse ya que son las que utiliza por defecto.

summary(KM.surv)      # permite visualizar la función de
                      # supervivencia estimada.
```


Salida 5.1

Se muestran los primeros 10 registros.

time	n.risk	n.event	survival	std.err	lower 95% CI	upper 95% CI
0.00000	84	3	0.9643	0.0202	0.89336	0.988
0.00274	81	2	0.9405	0.0258	0.86290	0.975
0.00548	79	2	0.9167	0.0302	0.83314	0.959
0.00822	77	1	0.9048	0.0320	0.81858	0.951
0.02192	76	1	0.8929	0.0337	0.80422	0.943
0.03562	75	1	0.8810	0.0353	0.79004	0.934
0.06027	74	1	0.8690	0.0368	0.77603	0.925
0.06575	73	1	0.8571	0.0382	0.76217	0.916
0.06849	72	1	0.8452	0.0395	0.74845	0.907
0.08767	71	1	0.8333	0.0407	0.73487	0.898

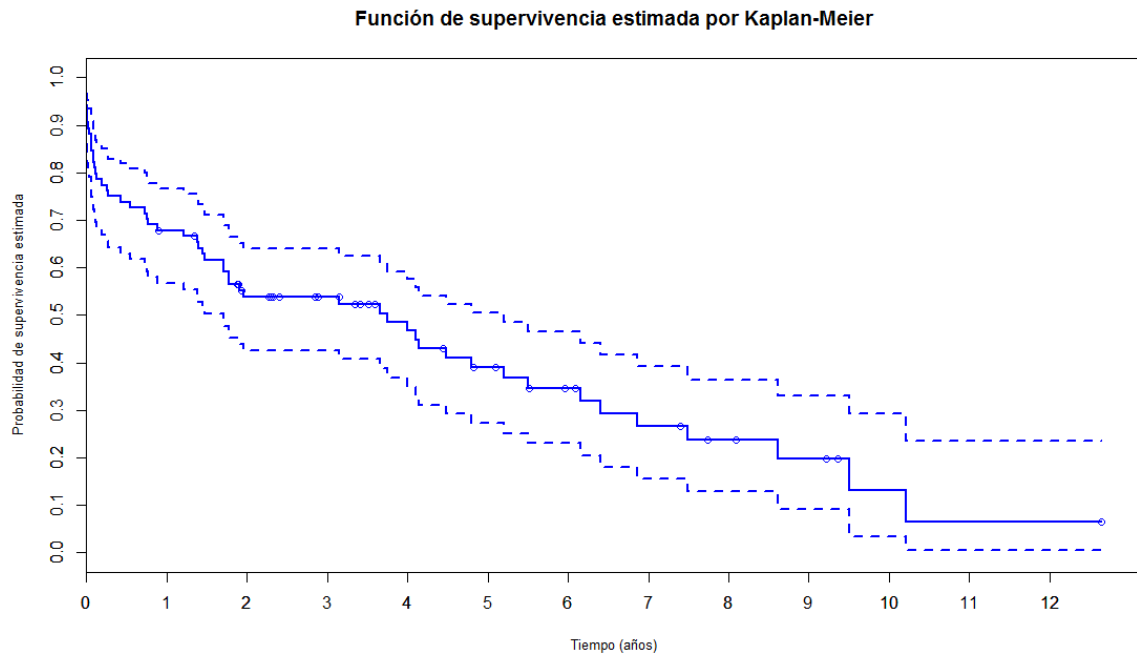
...

Para obtener el gráfico de la función de supervivencia se utiliza el comando **plot**, aplicado al objeto de R que contiene la función estimada, tal como se muestra en el Código 5.2. Notar que por defecto el gráfico incluye los intervalos de confianza junto con la curva de supervivencia estimada.

Código 5.2

```
plot(KM.surv,
     main="Función de supervivencia estimada por Kaplan-Meier",
     xlab="Tiempo (años)",
     ylab="Probabilidad de supervivencia estimada",
     lty=1, lwd=2, cex=1, cex.lab=0.8, mark=1, col="blue")
# En primer lugar se indica el objeto que
# contiene la función estimada (KM.surv en
# este caso).
# El título del gráfico se indica con main y
# los nombres de los ejes con xlab e ylab.
# lty: indica el tipo de línea de la curva (en
# este caso línea sólida)
# lwd: indica el ancho de la línea.
# cex: indica el tamaño de los puntos
# indicadores de censura.
# cex.lab: indica el tamaño de los nombres de
# los ejes.
# mark: indica la forma de los puntos
# indicadores de censura (en este caso círculos
# sin relleno.
# col: indica el color de la curva

axis(1, seq(0,12,1))
axis(2, seq(0,1,.1))
# en la sentencia axis se identifica con el 1
# al eje de las abscisas y con 2 al de las
# ordenadas. Luego, con seq() se define el
# mínimo, máximo y la escala del eje en cuestión.
```

Salida 5.2

Si se desea obtener información sobre los cuartiles, una forma es hacerlo a través de la utilización de la función **quantile** aplicada al objeto que contiene la función de supervivencia estimada.

Código 5.3

```
quantile(KM.surv,
  quantiles=c(0.25, 0.5, 0.75),
  conf.int=.95) # en la opción quantiles se indican separados
                # por comas los cuantiles para los cuales se
                # quiere conocer la estimación.
                # Al igual que en la función survfit, en la
                # opción conf.int se especifica el
                # coeficiente de confianza.
```

Salida 5.3

```
$quantile
      25      50      75
0.3506849 3.7452055 7.4904110

$lower
      25      50      75
0.06849315 1.70684931 5.50410959

$upper
      25      50      75
1.386301  5.202740 10.213699
```

Recordar:

En el software SAS la estimación de la función de supervivencia se obtiene mediante el procedimiento **lifetest**. También forman parte de la salida los cuartiles y el gráfico de la función estimada (si se utiliza la opción `plots=survival`).

```
Proc lifetest data=EPOC plots=(s);
time TIEMPO*EVENTO(0);
title "Funcion de supervivencia estimada por Kaplan-Meier";
run;
```

The LIFETEST Procedure

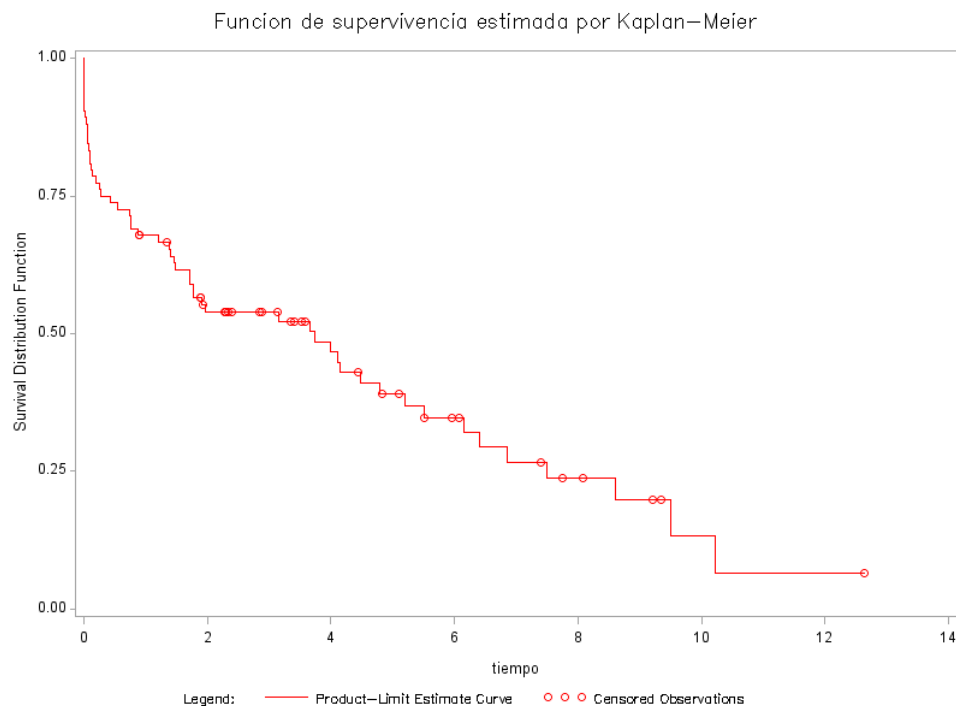
Product-Limit Survival Estimates

years	Survival	Failure	Survival Standard Error	Number Failed	Number Left
0.0000	1.0000	0	0	0	84
0.0000	.	.	.	1	83
0.0000	.	.	.	2	82
0.0000	0.9643	0.0357	0.0202	3	81
0.0027	.	.	.	4	80
0.0027	0.9405	0.0595	0.0258	5	79
0.0055	.	.	.	6	78
0.0055	0.9167	0.0833	0.0302	7	77
0.0082	0.9048	0.0952	0.0320	8	76
0.0219	0.8929	0.1071	0.0337	9	75
0.0356	0.8810	0.1190	0.0353	10	74
0.0603	0.8690	0.1310	0.0368	11	73
0.0658	0.8571	0.1429	0.0382	12	72
0.0685	0.8452	0.1548	0.0395	13	71
0.0877	0.8333	0.1667	0.0407	14	70
...	...	...	...	...	...

Summary Statistics for Time Variable years

Quartile Estimates

Percent	Point Estimate	95% Confidence Interval [Lower Upper)	
75	7.4904	5.2027	10.2137
50	3.7452	1.7068	5.2027
25	0.3507	0.0932	1.4027



5.2 Método de Nelson-Aalen (Nelson-Altschuler)

Otra forma de estimar la función de supervivencia es utilizando el método de Nelson-Aalen. En este caso el estimador está dado por:

$$\hat{S}(t) = \prod_{j=1}^k \exp\left(\frac{-d_j}{n_j}\right) \quad k = 1, 2, \dots, \text{ryt}_{(k)} \leq t < t_{(k+1)}$$

donde, d_j y n_j se definen tal como se detalló para el estimador de Kaplan-Meier.

La estimación por Nelson-Aalen se obtiene utilizando en la función `survfit` la opción `type="fleming-harrington"`, tal como se observa en el código 5.4.

Código 5.4

```
NAalen.surv<- survfit(Surv(tiempo, evento==1) ~ 1,
  type="fleming-harrington", error="tsiatis", conf.int = .95)
  # crea la función de supervivencia utilizando
  # el método de Nelson Aalen, con sus
  # respectivos intervalos de confianza.
summary(NAalen.surv)
```

Salida 5.4

Se muestran los primeros 11 registros.

time	n.risk	n.event	survival	std.err	lower	95% CI upper	95% CI
0.00000	84	3	0.9649	0.0199	0.9267	1.000	
0.00274	81	2	0.9414	0.0254	0.8928	0.993	
0.00548	79	2	0.9178	0.0297	0.8614	0.978	
0.00822	77	1	0.9060	0.0316	0.8461	0.970	
0.02192	76	1	0.8942	0.0334	0.8311	0.962	
0.03562	75	1	0.8823	0.0350	0.8164	0.954	
0.06027	74	1	0.8705	0.0364	0.8019	0.945	
0.06575	73	1	0.8586	0.0378	0.7876	0.936	
0.06849	72	1	0.8468	0.0391	0.7735	0.927	
0.08767	71	1	0.8349	0.0403	0.7596	0.918	
0.09315	70	1	0.8231	0.0414	0.7457	0.908	

...

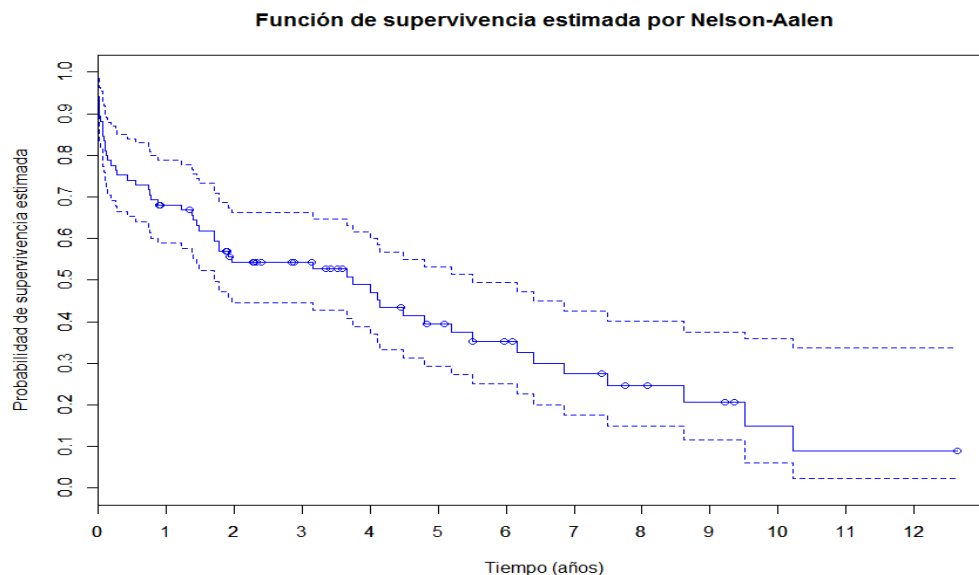
Una vez estimada la función de supervivencia, puede graficarse de la misma forma que se procedió con el estimador de Kaplan-Meier.

Código 5.5

```
plot(NAalen.surv,
     main="Función de supervivencia estimada por Nelson-Aalen",
     xlab="Tiempo (años)",
     ylab="Probabilidad de supervivencia estimada",
     lty=1, lwd=2, cex=1, cex.lab=0.8, mark=1, col="blue")

axis(1, seq(0,12,1))
axis(2, seq(0,1,.1))
```

Salida 5.5

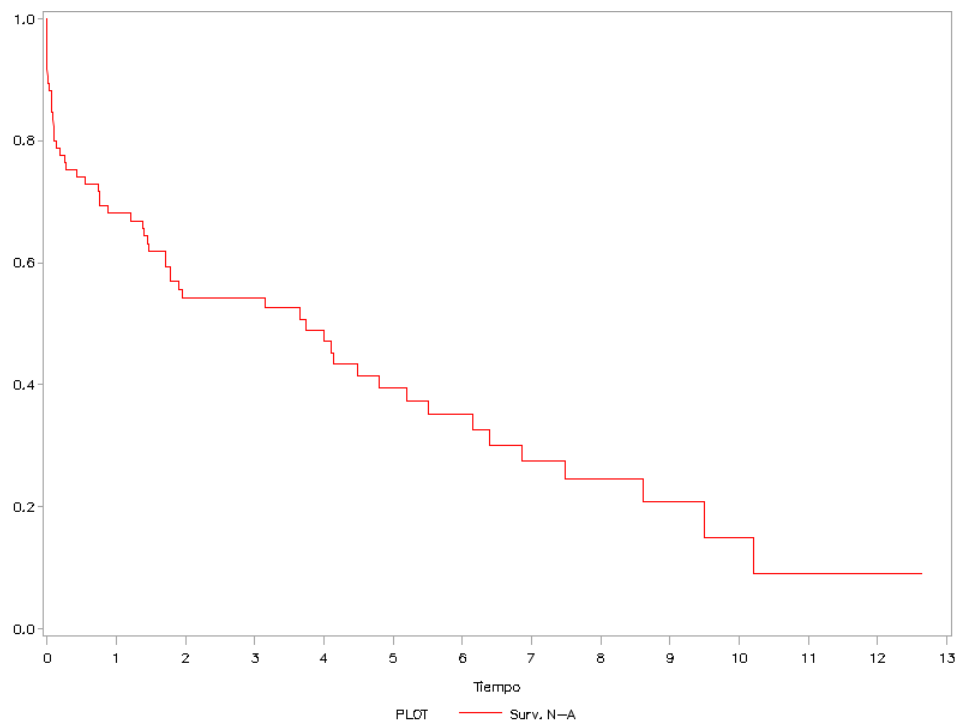


Recordar:

En el software SAS se puede obtener este estimador a través de la macro ***nelaalen*** (la cual debe leerse indicando el directorio donde está guardada y luego ejecutarse).

```
%include "C:\Nelaalen.sas";
%nelaalen(data=EPOC, time=tiempo, cens=evento);
```

Estimacion de Funciones de Supervivencia							
Tiempo	En Riesgo	Eventos	Freq.	Hazard N-A	Hazard Acumulado N-A	Surv. N-A	Surv. K-M
0.00000	84	3	3	0.035714	0.03571	1.00000	1.00000
0.00274	81	2	2	0.024691	0.06041	0.94138	0.94048
0.00548	79	2	2	0.025316	0.08572	0.91785	0.91667
0.00822	77	1	1	0.012987	0.09871	0.90601	0.90476
0.02192	76	1	1	0.013158	0.11187	0.89416	0.89286
0.03562	75	1	1	0.013333	0.12520	0.88232	0.88095
0.06027	74	1	1	0.013514	0.13871	0.87048	0.86905
0.06575	73	1	1	0.013699	0.15241	0.85863	0.85714
0.06849	72	1	1	0.013889	0.16630	0.84679	0.84524
0.08767	71	1	1	0.014085	0.18039	0.83495	0.83333
...	...	...	...	...	...	...	...



5.3 Método de Tablas de Vida

Si los datos con los que se cuenta se presentan agrupados por intervalos, tal como se muestran en la Tabla 1, es necesario aplicar el Método de Tablas de vida para su análisis. A fin de ejemplificar este método se creó un archivo con los datos agrupados: "EPOC_agrup.txt". En el código 5.6 se muestra cómo acceder a los datos en este caso.

Tabla 1: Tiempos de supervivencia de pacientes con TP agrupados por intervalos.

Intervalo de tiempo	Nº de individuos en riesgo	Nº de eventos	Nº de censuras
[0,2)	84	6	38
[2,4)	40	11	4
[4,6)	25	5	6
[6,8)	14	3	4
[8,10)	7	3	2
[10,12)	2	0	1
[12,14)	1	1	0

Código 5.6

```
EPOC_agrup<-read.table('EPOC_agrup.txt',header=T)
# lee un conjunto de datos a partir de un archivo de
# texto y lo asigna al objeto de R denominado
# EPOC_agrup.
# La opción header=T indica que la primera fila
# contiene los nombres de las variables.

attach(EPOC_agrup)
# indica que todas las variables mencionadas en los
# sucesivos códigos se encuentran en el
# objeto EPOC_agrup.

EPOC_agrup
# permite visualizar el contenido del objeto
# EPOC_agrup, recién creado.
```

Salida 5.6

Nriesgo	Ncens	Neven
1	84	6
38		
2	40	11
4		
3	25	5
6		
4	14	3
4		

En la Salida 5.6 puede observarse que el objeto `EPOC_agrup` contiene: el número de individuos en riesgo en cada intervalo (`Nriesgo`), el número de eventos en cada intervalo (`Neven`) y el número de censuras en cada intervalo (`Ncens`).

Para indicar los intervalos es necesario completar el código anterior creando un vector que contenga el valor inicial de cada uno de los intervalos de tiempo y además un valor adicional que indica el valor final del último intervalo. Es decir que tendrá una longitud mayor en una unidad en comparación con el archivo que contiene los datos.

```
intervalos<-c(0,2,4,6,8,10,12,14)
intervalos#permite visualizar el contenido del objeto creado
```

La función de supervivencia estimada por medio del método de Tablas de Vida, puede obtenerse mediante la función **lifetab**, perteneciente al paquete `KMsurv`. En el caso de la aplicación, el código correspondiente es el siguiente:

#Código 5.7

```
library(KMsurv)

TVida.surv<-lifetab(intervalos, Nriesgo[1], Ncens, Neven)
# crea, y asigna al objeto TVida.surv,
# la función de supervivencia
# utilizando el método de tablas de vida.
# Nriesgo[1]: indica el número de sujetos que
# inicialmente ingresaron al estudio, el primer
# valor de la variable Nriesgo.

TVida.surv[,1:5]
# permite visualizar las primeras 5 columnas del
# objeto TVida.surv, las cuales contienen la
# información necesaria para construir la curva
# de supervivencia.
```


Salida 5.7

	nsub	snlost	nrisk	nevent	surv
0-2	84	6	81.0	38	1.0000000
2-4	40	11	34.5	4	0.5308642
4-6	25	5	22.5	6	0.4693147
6-8	14	3	12.5	4	0.3441641
8-10	7	3	5.5	2	0.2340316
10-12	2	0	2.0	1	0.1489292
12-14	1	1	0.5	0	0.0744646

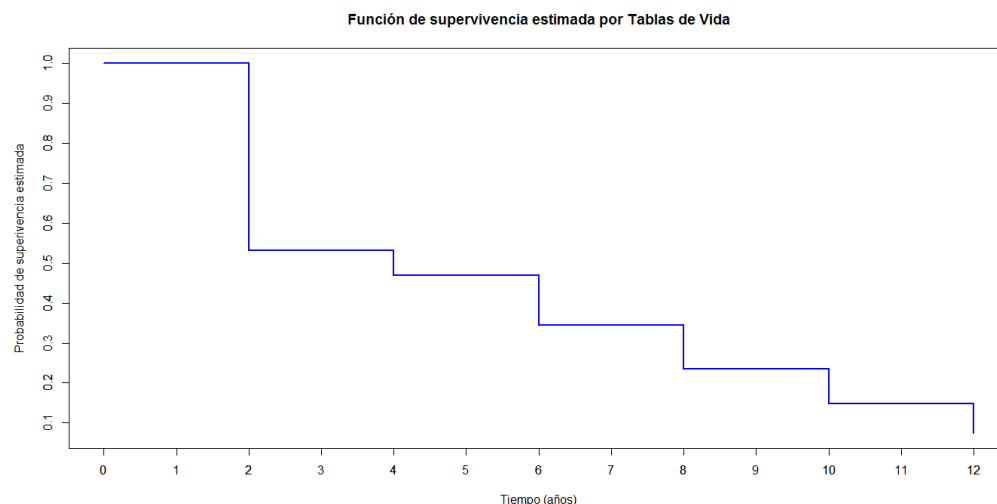
Nuevamente para construir el gráfico de la curva de supervivencia estimada por este método se utiliza la función **plot**.

Código 5.8

```
plot(intervalos[1:7],TVida.surv[,5], type="s",
     main="Función de supervivencia estimada por Tablas de Vida",
     xlab="Tiempo (años)",
     ylab="Probabilidad de supervivencia estimada",
     lty=1, lwd=2, cex=1, cex.lab=0.8,col ="blue")
# Al indicar como primer parámetro de la función
# plot "intervalos", se obtiene en el eje del tiempo
# la escala en función de los límites de los
# intervalos.
# TVida.surv[,5]: indica que del objeto "TVida.surv"
# se utilice la quinta columna, la cual contiene
# la supervivencia estimada.
# type="s":indica que los puntos
# correspondientes a la función estimada
# aparezcan unidos y en forma escalonada.

axis(1, seq(0,12,1))
axis(2, seq(0,1,.1))
```

Salida 5.8



Recordar:

En SAS, la estimación de la función de supervivencia se realiza por medio del procedimiento **lifetest**, especificando el método a utilizar mediante `method=lt`.

```
data intervalos;
input tiempo censura freq;
cards;
1 0 6
1 1 38
3 0 11
3 1 4
5 0 5
5 1 6
7 0 3
7 1 4
9 0 3
9 1 2
11 0 0
11 1 1
13 0 1
13 1 0
;
run;

ods graphics on;
proclifetest data=intervalos method=lt intervals=(0 to 14 by 2) plots=(s);
time tiempo*censura(0);
freq freq;
run;
ods graphics off;
```

Life Table Survival Estimates

Interval [Lower, Upper)		Number Failed	Number Censored	Effective Sample Size	Conditional Probability of Failure	Conditional Probability Standard Error
0	2	38	6	81.0	0.4691	0.0554
2	4	4	11	34.5	0.1159	0.0545
4	6	6	5	22.5	0.2667	0.0932
6	8	4	3	12.5	0.3200	0.1319
8	10	2	3	5.5	0.3636	0.2051
10	12	1	0	2.0	0.5000	0.3536
12	14	0	1	0.5	0	0

Interval [Lower, Upper)		Survival	Failure	Survival Standard Error	Median Residual Lifetime	Median Standard Error
0	2	1.0000	0	0	3.0029	1.8052
2	4	0.5309	0.4691	0.0554	5.4298	0.8207
4	6	0.4693	0.5307	0.0569	3.9886	0.8984
6	8	0.3442	0.6558	0.0605	3.4559	1.1438
8	10	0.2340	0.7660	0.0613	2.8571	1.3401
10	12	0.1489	0.8511	0.0618	2.0000	1.4142
12	14	0.0745	0.9255	0.0611	.	.



6. Estimación de la función hazard a través del método de Nelson-Aalen

La función hazard es una función fundamental en análisis de supervivencia; especifica la tasa instantánea a la cual los eventos ocurren para individuos que sobrevivieron hasta el tiempo t . Si T es una variable continua se tiene que:

$$h(t) = \frac{f(t)}{S(t)}$$

Esta función sirve para determinar la distribución de los eventos utilizando información cualitativa acerca del mecanismo de falla (o de generación de eventos) y para describir cómo dicho mecanismo cambia a través del tiempo.

Una cantidad relacionada es la función hazard acumulada, $H(t)$, definida como:

$$H(t) = \int_0^t h(u) du = -\ln[S(t)]$$

Haciendo uso de esta última relación, la función hazard acumulada estimada a través del método de Nelson-Aalen se puede obtener a partir de la función de supervivencia calculada por el mismo método en la sección 5.2.

La función hazard acumulada estimada se obtiene tal como se muestra en el código 6.1.

Código 6.1

```
h.aalen<-(-log (NAalen.surv$surv) )
# Se le asigna al objeto llamado h.aalen el valor
# negativo del logaritmo de los valores de la función
# de supervivencia estimada por Nelson-Aalen para
# obtener la estimación de la función hazard
# acumulada.

h.aalen
```

Salida 6.1

Se muestran los primeros 42 valores.

```
[1] 0.03571429 0.06040564 0.08572210 0.09870911 0.11186701 0.12520034 0.13871385
[8] 0.15241248 0.16630137 0.18038588 0.19467159 0.20916435 0.22387023 0.23879560
[15] 0.25394712 0.26933173 0.28495673 0.30082975 0.31695878 0.33335222 0.35001889
[22] 0.36696804 0.38420942 0.38420942 0.38420942 0.40239124 0.40239124 0.42125917
[29] 0.44048994 0.46009778 0.48009778 0.50050594 0.52133927 0.56389247 0.56389247
[36] 0.56389247 0.58714828 0.58714828 0.61153852 0.61153852 0.61153852 0.61153852
```

...

Si se quiere visualizar de forma conjunta la función hazard acumulada recientemente estimada y la función de supervivencia, se puede utilizar la siguiente sentencia.

Código 6.2

```
attach(NAalen.surv)

aalen.est<- data.frame(time=time, d=n.event,
  n=n.risk, h.aalen, superv_NA=surv)
  #data.frame: crea un data a partir de una lista
  # de variables con mismo número de filas,
  # time=time: construye el tiempo a partir del vector
  # denominado time,
  # d=n.event: construye el número de eventos
  # en cada tiempo a partir del vector n.event,
  # n=n.risk: construye el número de eventos en cada tiempo
  # a partir del vector n.event,
  # h.aalen: objeto que contiene el negativo del
  # logaritmo de los valores de la función de
  # supervivencia estimada por Nelson-Aalen,
  # superv_NA=surv: construye la función
  # de supervivencia estimada por Nelson-Aalen a
  # partir del vector surv.

aalen.est

detach(NAalen.surv)
```

Salida 6.2*Se muestran las primeras 8filas.*

```

      time d  n    h.aalen  superv_NA
[1,] 0.000000000 3 84 0.03571429 0.96491594
[2,] 0.002739726 2 81 0.06040564 0.94138259
[3,] 0.005479452 2 79 0.08572210 0.91784927
[4,] 0.008219178 1 77 0.09870911 0.90600622
[5,] 0.021917808 1 76 0.11186701 0.89416317
[6,] 0.035616438 1 75 0.12520034 0.88232012
[7,] 0.060273973 1 74 0.13871385 0.87047708
[8,] 0.065753425 1 73 0.15241248 0.85863403

```

...

Para realizar el gráfico de la función hazard estimada por Nelson-Aalen junto con una curva loess se pueden utilizar las funciones **plot** para el gráfico de dispersión y **lines** para ajustar la curva loess de la siguiente manera.

Código 6.3

```

attach(aalen.est)

h<-d/n                                # asigna al vector h el cociente entre el número
                                      # de evento y el número de individuos en riesgo
                                      # en cada tiempo.

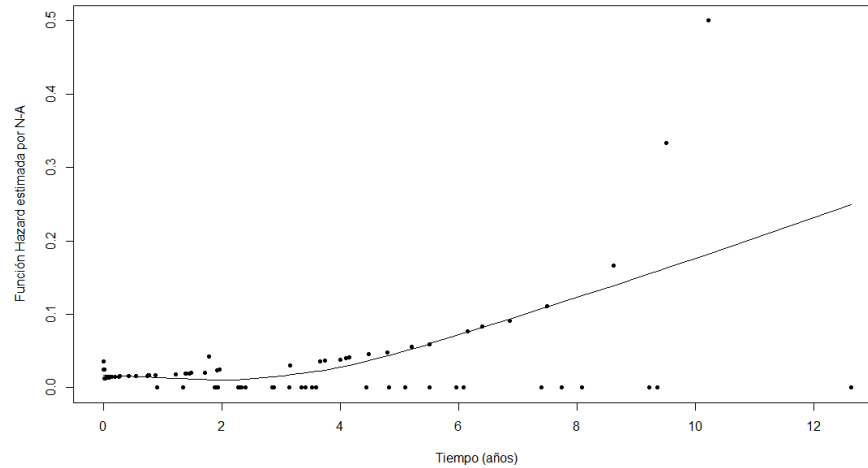
plot(time, h, type="p", pch=20,
      xlab="Tiempo (años)",
      ylab="Función Hazard estimada por N-A")
lines(lowess(time, h))

# time: son los tiempos de supervivencia
# que forman el eje X del # gráfico,
# h: son los valores del elemento h que forman el eje
# Y del gráfico,
# type="p": indica que es un gráfico de puntos,
# pch=20: indica la forma de los puntos,
# lines(lowess(time, h)): agrega una curva loess
# al gráfico de dispersión.

detach(aalen.est)

```

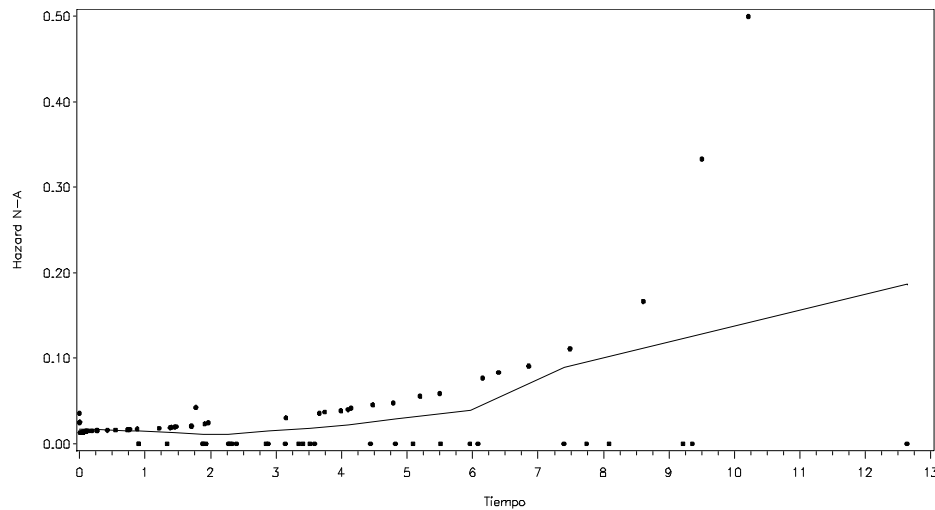
Salida 6.4

**Recordar:**

En SAS, como resultado de la macro **nelaalen** se obtiene el data `hazard`, el cual contiene las estimaciones de la función hazard por Nelson-Aalen tanto en forma puntual como acumulada, bajo los nombres `hazna` y `hazcum` respectivamente. Por lo tanto, a través de los procedimientos **proc loess** y **procgplot** se puede obtener el gráfico correspondiente, utilizando el data `hazard`.

```
Proc loess data=hazard;
modelhazna=tiempo / smooth=0.6;
ods outputOutputStatistics=Results;
run;
axis1 label=(angle=90rotate=0) minor=none;
symbol1 color=black value=dot i=none h=0.5;
symbol2 color=black interpol=join value=none;

procgplotdata=Results;
plotDepVar*tiempo=1Pred*tiempo / vaxis=axis1 hm=3vm=3 overlay;
formatdepvar5.2 tiempo 2.;
run; quit;
```



7. Comparación de curvas de supervivencia estimadas

7.1 Comparación de dos grupos

La forma más simple de comparar los tiempos de supervivencia obtenidos a partir de dos grupos de individuos es presentar las estimaciones de las funciones de supervivencia en un mismo gráfico.

En el código 7.1 se presenta la sentencia necesaria para obtener la estimación de la función de supervivencia en forma separada para los pacientes que fueron trasplantados bajo el estado de urgencia o emergencia y para aquellos que lo hicieron en forma electiva. Luego, en el código 7.2 se muestra cómo obtener el gráfico correspondiente, a través de la función `plot`.

Código 7.1

```
KM_estadoclinico<-survfit(Surv(tiempo, evento==1)~estado_clinico)
# estima la función de supervivencia
# según la variable que se quiere comparar, la cual
# se especifica luego del símbolo "~", en este caso
# "estado_clínico".
```

```
summary(KM_estadoclinico)
```


Salida 7.1

Se muestran los primeros 7 registros para el grupo de trasplante electivo.

```

                                estado_clinico=Electivo
time n.risk n.event survival std.err lower 95% CI upper 95% CI
0.00000    70      2   0.9714  0.0199   0.9332    1.000
0.00274    68      2   0.9429  0.0277   0.8900    0.999
0.00548    66      2   0.9143  0.0335   0.8510    0.982
0.00822    64      1   0.9000  0.0359   0.8324    0.973
0.02192    63      1   0.8857  0.0380   0.8142    0.963
0.03562    62      1   0.8714  0.0400   0.7964    0.953
0.06027    61      1   0.8571  0.0418   0.7790    0.943

...

                                estado_clinico=Urgencia/Emergencia
time n.risk n.event survival std.err lower 95% CI upper 95% CI
0.0000    14      1   0.929  0.0688   0.803    1.000
0.0959    13      1   0.857  0.0935   0.692    1.000
0.8822    12      1   0.786  0.1097   0.598    1.000
1.4767    11      1   0.714  0.1207   0.513    0.995
1.7068    10      1   0.643  0.1281   0.435    0.950
4.1014     4      1   0.482  0.1691   0.242    0.959

```

Código 7.2

```

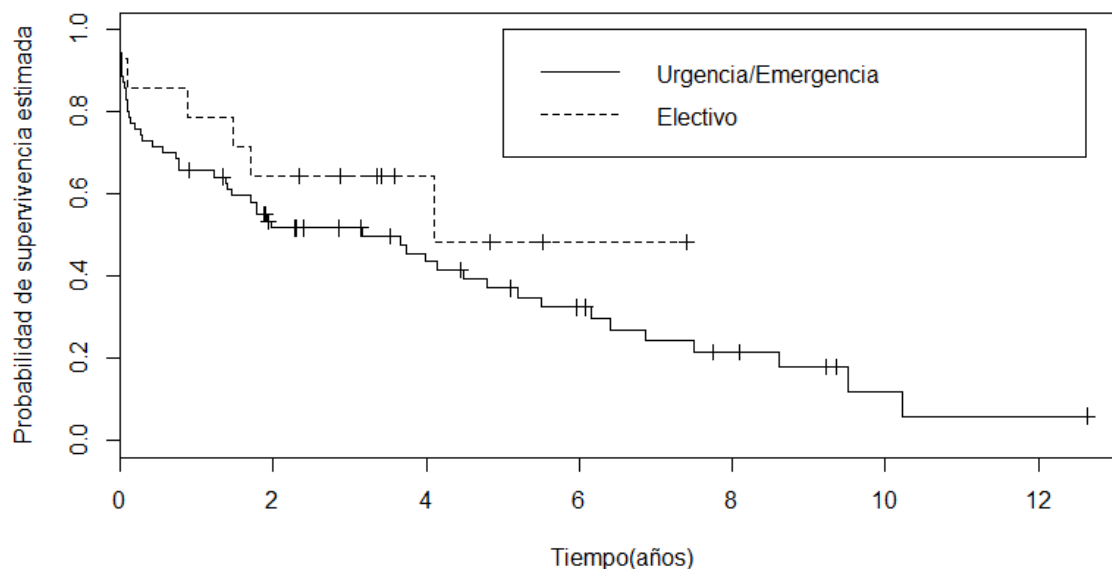
plot(KM_estadoclinico,
     xlab="Tiempo(años)",
     ylab="Probabilidad de supervivencia estimada",
     main="Estimador de Kaplan-Meier según estado clínico",
     lty=c(1,2))      # lty=c(1,2): es un vector que indica el
                     # tipo de línea, en este caso 1=línea sólida y
                     # 2=línea punteada.

legend(x=5,y=1,
       legend=c("Urgencia/Emergencia","Electivo"),lty=c(1,2))
                     # legend: esta función se puede utilizar para
                     # agregar leyenda al gráfico.
                     # x e y son las coordenadas que se utilizan para
                     # colocar la leyenda.

```

Salida 7.2

Estimador de Kaplan-Meier según estado clínico



Existen dos explicaciones posibles para la diferencia observada entre las funciones de supervivencia estimadas; puede ocurrir que exista una diferencia real en la supervivencia entre los dos grupos o bien que no exista tal diferencia y que la diferencia observada sea solamente el resultado del azar. Para decidir entre las dos explicaciones se utiliza un test de hipótesis. Dicho test puede realizarse a través de la función `survdif`, que permite efectuar el contraste de hipótesis para verificar la igualdad o diferencia de dos o más curvas de supervivencias en base a la familia de pruebas G-rho propuestas por Harrington y Fleming (1982). La familia de tests G-rho de Fleming y Harrington es otra forma de estudiar los test *log-rank* de Peto-Peto. Esos dos autores sugieren pesos de la forma: $w_i = [\hat{S}_{KM}(t_{i-1})]^\rho$; haciendo $\rho = 0$ se tiene que $w_i = 1$ (test *log-rank*) y, si $\rho = 1$, se obtiene el test de Peto-Peto. Por lo tanto, es fundamental especificar el argumento “rho” en el código de R.

Por ejemplo, si se quiere obtener el test log-rank para evaluar si la diferencia observada en el gráfico según estado clínico es significativa puede utilizarse el código 7.3.

Código 7.3

```
survdif(Surv(tiempo, evento==1)~estado_clinico, rho=0)
# survdif: comprueba si hay una diferencia entre dos
# o más curvas de supervivencia; con rho=0 se
# obtiene el test log-rank.
```

Salida 7.3

	N	Observed	Expected	$(O-E)^2/E$	$(O-E)^2/V$
estado_clinico=Electivo	70	49	46.19	0.172	1.1
estado_clinico=Urgencia/Emergencia	14	6	8.81	0.899	1.1

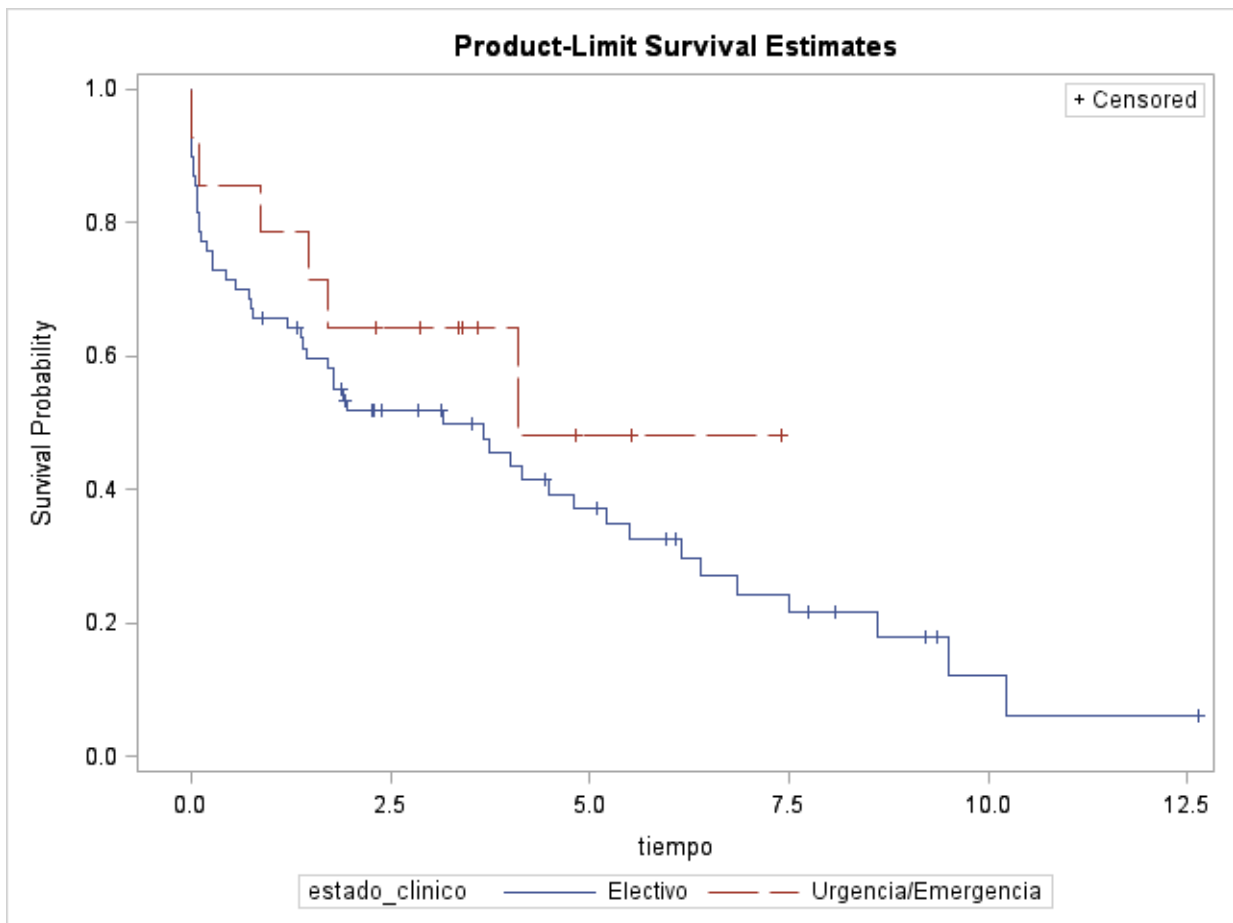
chisq= 1.1 on 1 degrees of freedom, p= 0.295

Recordar:

En SAS se obtienen las estimaciones de las funciones de supervivencia para cada grupo junto con los tests de hipótesis mediante la sentencia `strata` dentro del procedimiento `lifetest`.

```
symbol1i=join v=none line=2 color=darkblue;
symbol3i=join v=none line=1 color=red;

proclifetestdata=EPOC plots=s cs=dot;
time tiempo*evento(0);
strataestado_clinico;
title"Función de supervivencia estimada por Kaplan-Meier según estado
clínico";
run;
```



Test of Equality over Strata

Test	Chi-Square	DF	Pr > Chi-Square
Log-Rank	1.0988	1	0.2945
Wilcoxon	0.9804	1	0.3221
-2Log(LR)	1.4813	1	0.2236

7.2 Test log-rank de tendencia

El test log-rank de tendencia se utiliza para detectar un ordenamiento en las curvas de supervivencia estimadas cuando se comparan tres o más grupos que presentan un orden. Por ejemplo, si se quiere comparar distintas dosis de un tratamiento o bien distintos grupos etarios. En **R** para obtener este test se suele recurrir al planteo de un modelo de Cox, considerando la variable que se quiere analizar en el predictor. La función que permite ajustar este modelo es `coxph` la cual se utiliza igual que la función `survdiff` ya explicada.

En el caso de la aplicación, supongamos que resulta de interés estudiar la experiencia de supervivencia de los individuos de acuerdo a ciertos grupos etarios. En primer lugar es necesario definir estos grupos, lo cual se muestra en el código 7.4; luego, en el código 7.5 se muestra cómo obtener el test de tendencia correspondiente.

Código 7.4

```
Base_EPOC[, "edad_cat"] <- cut(Base_EPOC$Edad,
breaks = c(0, 52, 58, 70), labels= FALSE)
# cut: divide una variable continua en intervalos.
# breaks: es un vector que contiene los puntos
# de corte. En este caso quedan definidos 4 grupos:
# (0;52], (52;58], (58;70] y más de 70 años.
# labels=FALSE: asigna códigos con números
# enteros en lugar de un factor.
```

Código 7.5

```
test_tendencia <- coxph(Surv(tiempo, evento==1) ~ Base_EPOC$edad_cat)
summary(test_tendencia)
```

Salida 7.5

```

n= 84, number of events= 55

            coef exp(coef) se(coef)      z Pr(>|z|)
edad_cat  0.2267    1.2545   0.1828  1.24   0.215

            exp(coef) exp(-coef) lower .95 upper .95
edad_cat    1.254    0.7971    0.8767    1.795

Concordance= 0.547 (se = 0.042 )
Rsquare= 0.018 (max possible= 0.992 )
Likelihood ratio test= 1.54 on 1 df,  p=0.2145
Wald test               = 1.54 on 1 df,  p=0.2149
Score (logrank) test = 1.55 on 1 df,  p=0.2136

```

Recordar:

En SAS el test de tendencia se obtiene mediante la sentencia `test` en el procedimiento `lifetest`.

```

Proc lifetest data=EPOC_2 ;
  time tiempo*evento(0);
  testedad_cat;
run;

```

```

Univariate Chi-Squares for the Log-Rank Test

Variable      Test      Standard      Chi-Square      Pr >
Statistic      Error
edad_cat      -6.7997      5.4620      1.5498      0.2132

```

7.3 Test estratificado

El test estratificado se utiliza para comparar dos o más conjuntos de datos de supervivencia teniendo en cuenta una variable adicional. Por ejemplo, si se quisiera evaluar la existencia de diferencias en la supervivencia de los pacientes según su estado clínico, estratificando (o controlando) por grupo etario. Este test puede llevarse a cabo corriendo la función `survdiff` (**# Código 7.6**) o recurriendo nuevamente a un modelo de Cox (**# Código 7.7**).

Código 7.6

```

Estratificado1<-survdiff(Surv(tiempo, evento==1)~estado_clinico +
                        strata(Base_EPOC$edad_cat)

```

Salida 7.6

	N	Observed	Expected	(O-E)^2/E	(O-E)^2/V
estado_clinico=Electivo	70	49	45.79	0.226	1.43
estado_clinico=Urgencia/Emergencia	14	6	9.21	1.121	1.43

Chisq= 1.4 on 1 degrees of freedom, p= 0.2

Código 7.7

```
Estratificado<-coxph(Surv(tiempo,evento==1)~
                      strata(Base_EPOC$edad_cat)+estado_clinico)

summary(estratificado)
```

Salida 7.7

```
n= 84, number of events= 55
```

	coef	exp(coef)	se(coef)	z	Pr(> z)
estado_clinicourgencia/Emergencia	-0.5226	0.5930	0.4408	-1.186	0.236

	exp(coef)	exp(-coef)	lower .95	upper .95
estado_clinicourgencia/Emergencia	0.593	1.686	0.2499	1.407

```
Concordance= 0.529 (se = 0.052 )
Rsquare= 0.019 (max possible= 0.968 )
Likelihood ratio test= 1.59 on 1 df, p=0.2073
Wald test = 1.41 on 1 df, p=0.2358
Score (logrank) test = 1.44 on 1 df, p=0.2306
```

Recordar:

Para obtener el test estratificado en SAS se utilizan las sentencias `test` y `strata` de forma conjunta. De esta manera, las estadísticas producidas por la sentencia `test` son estadísticas estratificadas que controlan por las variables listadas en la sentencia `strata`.

```
Proc lifetest data=EPOC 2;
  time tiempo*evento(0);
  strata edad_cat;
  test estado_clinico_;
run;
```

Chi-cuadrado de una variable para el test log-rango

Variable	Estadísticos del test	Error estándar	Chi-cuadrado	Pr > Chi-cuadrado
estado_clinico_	3.2145	2.6896	1.4284	0.2320

8. Referencias bibliográficas

- **Collett, David (2003)**

“Modelling Survival Data in Medical Research”. Segunda Edición. United State of America: Chapman & Hall/CRC.

- **Borges Peña, Rafael Eduardo (2005)**

“Análisis de Supervivencia utilizando el lenguaje R”. Disponible en: <http://webdelprofesor.ula.ve/economia/borgesr/PaipaREBP.pdf>. [Último acceso: Julio 2015].

- **Moore, Dirk (2016)**

“Applied Survival Analysis Using R”. Springer.

- **Hosmer, David; Lemeshow, Stanley**

“R Textbook Examples: Applied Survival Analysis”. Chapter 2: “Descriptive Methods for Survival Data”. Disponible en: http://www.ats.ucla.edu/stat/sas/examples/asa/asa2_may_2006.htm. [Último acceso: Julio 2015].

- **Fox, John; Weisberg, Sanford (2011)**

“Cox Proportional-Hazards Regression for Survival Data”. Disponible en: <http://socserv.socsci.mcmaster.ca/jfox/Books/Companion/appendix/Appendix-Cox-Regression.pdf>. [Último acceso: Julio 2015].

- **Therneau, Terry M (2015)**

Packages “survival”. The R Project for Statistical Computing. Disponible en: <https://cran.r-project.org/web/packages/survival/survival.pdf>. [Último acceso: Julio 2015].

- **Klein; Moeschberger; Yan, Jun (2015)**

Packages “KMsurv”. The R Project for Statistical Computing. Disponible en: <https://cran.r-project.org/web/packages/KMsurv/KMsurv.pdf>. [Último acceso: Julio 2015].

- **Borges, Rafael (2004)**

Análisis de supervivencia básico utilizando el lenguaje R. Disponible en: http://www.academia.edu/3089513/Analisis_de_supervivencia_basico_utilizando_el_lenguaje_R. [Último acceso: Julio 2015].