



Universidad Central de Venezuela  
Facultad de Ciencias Económicas y Sociales  
Escuela de Estadística y Ciencias Actuariales  
Área de Postgrado en Estadística y Actuario

# VI Jornadas Académicas

A square logo with a grid pattern, containing a large Greek letter sigma ( $\Sigma$ ) and the year 2004 below it.

$\Sigma$

2004

ANÁLISIS DE SUPERVIVENCIA BÁSICO UTILIZANDO EL LENGUAJE R  
Profesor Rafael Borges

**14 de Octubre de 2004**

Tercer piso de la Escuela de Estadística y Ciencias Actuariales, salón 60-313,  
Res. 3B, Ciudad Universitaria de Caracas

# ÍNDICE

Introducción.....	1
Parte 1: Introducción al análisis de supervivencia.....	2
1.1. Censura y truncamiento.....	2
1.2. Definiciones básicas.....	3
1.2.1. Función de supervivencia.....	3
1.2.2. Funciones de riesgos (hazard).....	3
1.3. Estimador de Kaplan y Meier.....	4
1.4. Comparación de las funciones de supervivencia.....	5
1.5. Sobrevida media y mediana.....	7
1.5.1. Sobrevida media.....	7
1.5.2. Sobrevida mediana.....	7
1.6. El modelo de regresión de Cox.....	8
1.7. Contrastes de hipótesis para el modelo de Cox.....	9
1.7.1. Test de razón de verosimilitud.....	9
1.7.2. Test de Wald.....	9
1.7.3. Test de puntajes (score test).....	9
1.8. Modelos de Cox estratificados.....	10
1.9. Estudio de residuos en el análisis de supervivencia.....	10
1.9.1. Residuos de martingala.....	11
1.9.2. Residuos de desvíos (deviances).....	11
1.9.3. Residuos de puntajes (scores).....	12
1.9.4. Residuos de Schoenfeld.....	12
1.10. Interpretación del modelo de Cox.....	13
Parte 2. Aspectos relacionados con el lenguaje R.....	14
2.1. Introducción al lenguaje R.....	14
2.1.1. Instalación del lenguaje R.....	15
2.1.2. Instalación de los paquetes adicionales.....	15
2.1.3. Ayudas y documentación del R.....	15
2.1.4. Acceso a datos internos disponibles.....	16
2.1.5. Acceso a datos externos disponibles.....	16
2.1.6. La opción de asignación.....	16
2.1.7. Verificación de objetos disponibles.....	17
2.1.8. Eliminación de objetos no deseados.....	17
2.1.9. R diferencia las mayúsculas de las minúsculas.....	17
2.1.10. Datos faltantes en R.....	17
2.1.11. Comentarios en R.....	17
2.1.12. Creación de datos en R.....	17
2.1.13. Carga y descarga de objetos.....	18
2.1.14. Envío de gráficos a otros programas.....	18
2.1.15. Salida del lenguaje R.....	18
2.2. Análisis de supervivencia utilizando el lenguaje R.....	18
2.2.1. El paquete <i>survival</i> .....	19
2.2.1.1 La función <code>surv</code> .....	19
2.2.1.2 La función <code>survfit</code> .....	19

2.2.1.3. La función <code>survdiff</code> .....	20
2.2.1.4. La función <code>coxph</code> .....	20
2.2.1.5. La función <code>cox.zph</code> .....	21
2.2.1.6. La función <code>residuals</code> .....	21
2.3. Ejemplo de análisis de supervivencia utilizando el lenguaje R.....	22
2.3.1. Estructura del archivo de datos <code>dpa.txt</code> .....	22
2.3.2. Lectura de los datos en R.....	22
2.3.3. Estimación de la función de supervivencia a través del estimador de Kaplan y Meier.....	23
2.3.4. Comparación de funciones de supervivencia.....	25
2.3.5. Ajuste del modelo de Cox.....	26
2.3.6. Verificación de los supuestos del modelo de Cox.....	29
2.3.6.1. Supuesto de riesgos proporcionales.....	29
2.3.6.2. Residuos tipo deviance.....	31
2.3.6.3. Gráficos de influencias sobre la estimación de cada coeficiente.....	31
2.3.6.4. Forma funcional de las variables continuas.....	33
Referencias.....	36

## **Introducción.**

Este material ha sido elaborado para el curso titulado “Análisis de supervivencia básico utilizando el lenguaje R” que ha sido programado en las VI Jornadas Académicas de la Escuela de Estadística y Ciencias Actuariales (EECA) y del Área de Postgrado en Estadística y Actuariado (APGEA) de la Universidad Central de Venezuela (UCV).

Como el curso está diseñado para ser tomado por personas que no tienen conocimientos acerca del tópico estadístico conocido como análisis de supervivencia ni del lenguaje de procesamiento estadístico R, éste ha sido estructurado de tal manera de dar una introducción tanto al análisis de supervivencia, como del lenguaje R, y exponer las principales herramientas para llevar un análisis de supervivencia básico mediante el uso del lenguaje R.

El presente material ha sido dividido en dos secciones claramente diferenciadas una de la otra.

En la primera parte del material se presenta una visión introductoria del análisis de supervivencia en el cual se presenta mecanismos de censura y truncamiento, algunas definiciones básicas, la estimación del análisis de supervivencia a través del estimador de Kaplan y Meier, el modelo de regresión de Cox acompañado de la verificación de sus supuestos.

La segunda parte del material se refiere a una breve introducción al lenguaje R y a su uso en el análisis de supervivencia. En esta sección se presentan aspectos generales del lenguaje, aspectos específicos como herramienta para el análisis de supervivencia acompañado de un ejemplo completo.

Finalmente debemos enfatizar que este es un material que este curso pretende servir de motivación para que algunos de los asistentes decidan seguir el estudio de los métodos de análisis de supervivencia y del lenguaje R.

## Parte 1: Introducción al análisis de supervivencia.

El análisis de supervivencia consiste en un conjunto de técnicas para analizar el tiempo de seguimiento hasta la ocurrencia de un evento de interés. Este tiempo de seguimiento hasta que ocurra el evento de interés, también denominado tiempo de vida puede observarse completa o parcialmente. Un caso poco frecuente en la práctica es aquel en que se observan los individuos desde un evento inicial hasta el evento de final o de ocurrencia del fenómeno que se desea observar. A la ocurrencia del evento de interés se le suele denominar falla o muerte.

Ahora bien, es posible, y muy frecuente en la práctica encontrarse con situaciones en que se cuenten con observaciones incompletas de los períodos que transcurren entre el tiempo inicial y el tiempo final. Esto puede darse por censura o por truncamiento, y es precisamente bajo la presencia de censura o truncamiento que el análisis de supervivencia cobra una importancia primordial.

Debido a la presencia de censura y/o truncamiento al análisis de supervivencia se le conoce también como análisis de datos censurados y/o truncados. Otro de los nombre que recibe este análisis es análisis de confiabilidad (reliability), derivados de estudios de distribuciones del tiempo de vida en procesos industriales o de ingeniería. También puede ser denominado como análisis del tiempo hasta la ocurrencia de un evento.

En esta sección se presenta una visión introductoria del análisis de supervivencia, incluyéndose aspectos relacionados con censura y truncamiento, seguido de una serie de definiciones básicas para luego presentar una versión del estimador de Kaplan y Meier, continuando con los métodos de comparación de funciones de riesgos, posteriormente se presentan una serie de secciones relacionadas con el modelo de regresión de Cox.

### 1.1. Censura y truncamiento.

Existen dos mecanismos que no hacen posible la observación completa de los tiempos de seguimiento, como lo son la censura y el truncamiento. En cuanto a la censura existen dos tipos: “censura tipo I” en la cual los individuos son observados hasta un tiempo determinado y, la “censura tipo II” en la cual los individuos son observados hasta que ocurran un número determinado de fallas o eventos de interés. Los mecanismos de censura (tipo I) y truncamiento más frecuentes son presentados a continuación:

- i) **Censura por la derecha:** Se presenta cuando hasta la última observación que se le hace al individuo, aún no se ha ocurrido el evento que se desea observar. Existen varias razones para que se presente este tipo de censura:
  - Que hasta el momento de la finalización del estudio no haya ocurrido el evento, esto ocurriría en el caso de que el período de seguimiento sea finito.
  - Que el individuo haya abandonado el estudio.
  - Que haya ocurrido en el individuo otro evento que imposibilite la ocurrencia del evento que se desea observar.

- ii) **Censura por la izquierda:** Es poco común en análisis de supervivencia, se presenta cuando para la primera observación que se realiza sobre el individuo ya ha ocurrido el evento que se desea observar. Este tipo de censura suele confundirse con el truncamiento por la izquierda o la entrada tardía.
- iii) **Censura por intervalos:** Se presenta cuando solo se sabe que al individuo le ocurre el evento de interés entre un instante  $t_i$  y un tiempo  $t_j$ .
- iv) **Entrada tardía al estudio (truncamiento por la izquierda):** Se presenta cuando el individuo comienza a observarse posteriormente al verdadero evento inicial.
- v) **Truncamiento por la derecha:** Se presenta cuando sólo se incluyen los individuos que presentan el evento o falla de interés.

Para obtener un panorama general de los distintos tipos de censura puede verse el libro de Andersen y colaboradores (Andersen et al., 1993) o el de Klein y Moeschberger (1997).

## 1.2. Definiciones básicas.

### 1.2.1. Función de supervivencia.

La función de supervivencia se define como la probabilidad de que una persona sobreviva (no le ocurra el evento de interés) al menos hasta el tiempo  $t$ . Una definición más formal puede darse de la siguiente manera: sea  $T$  una variable aleatoria positiva (o no negativa) con función de distribución  $F(t)$  y función de densidad de probabilidad  $f(t)$ . La función de supervivencia  $S(t)$  es:

$$S(t) = 1 - F(t) = P[T > t]$$

### 1.2.2. Funciones de riesgos (hazard).

La función de razón de riesgos o tasa instantánea de fallas  $\lambda(t)$  se define como el cociente entre la función de densidad y la función de supervivencia:

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

Se interpreta como la probabilidad de que a un individuo le ocurra el evento de interés en la siguiente unidad de tiempo  $\Delta t$  dado que ha sobrevivido hasta el tiempo  $t$ .

Dicha función proviene de la tasa media de fallas, como sigue:

Dada la Probabilidad condicional de fallas en el período  $(t; t + \Delta t)$ , dado que la persona sobrevive en el período  $(0; t)$ , la tasa media de fallas (TMF) se define como:

$$\text{TMF} = \frac{F(t + \Delta t) - F(t)}{\Delta t} \frac{1}{S(t)}$$

Tomando límites para  $\Delta t \rightarrow 0$ , queda:

$$\lambda(t) = \lim_{\Delta t \rightarrow 0} \text{TMF} = \frac{F'(t)}{S(t)} = \frac{f(t)}{S(t)}$$

La función de riesgo acumulada  $\Lambda(t)$  se define como:

$$\Lambda(t) = \int_0^t \lambda(u) du = -\log S(t)$$

Como habíamos planteado anteriormente, lo que distingue al análisis de supervivencia es la presencia de la censura. El caso más común de censura es la censura por la derecha, que se caracteriza porque sólo se sabe que el tiempo de ocurrencia del evento de interés es mayor que el último tiempo de observación.

Los datos de supervivencia suelen presentarse en la forma  $(t_i, \delta_i)$  donde  $t_i$  es el tiempo de observación y,  $\delta_i = 0$  si la observación es censurada y  $\delta_i = 1$  cuando se observa la ocurrencia del evento de interés.

### 1.3. Estimador de Kaplan y Meier.

La presencia de datos censurados o truncados hace que la función de supervivencia no pueda ser obtenida directamente a través de argumentos probabilísticos haciéndose necesario el uso de algunos estimadores. Existen varias formas de estimar la función de supervivencia, entre los más conocidos son los basados en tablas de vida, entre el que se incluye el estimador actuarial y el estimador de Kaplan y Meier, que es más práctico, porque no es necesario trabajar con períodos de tiempos, sino que los mismos tiempos de observación van contribuyendo a la estimación de la función de supervivencia.

El estimador de Kaplan y Meier (1958) es el estimador de la función de supervivencia más utilizado y se define para el caso en que los datos puedan presentar censura por la derecha como:

$$\hat{S}_{KM}(t) = \prod_{t_i \leq t} \frac{r(t_i) - d(t_i)}{r(t_i)}$$

donde  $r(t_i)$  y  $d(t_i)$  son el número de individuos en riesgo y el número de muertes (o de ocurrencia del evento de interés) en el momento  $t_i$ .

La varianza del estimador de Kaplan y Meier se obtiene a través de la fórmula de Greenwood (1926):

$$V(\hat{S}_{KM}(t)) = \hat{S}_{KM}^2(t) \sum_{t_i \leq t} \frac{d(t_i)}{r(t_i)[r(t_i) - d(t_i)]}$$

El intervalo de confianza del 95% para escala plana (o de identidad), llamado así porque es obtenido de manera estándar al que se obtiene cualquiera de los intervalos de confianza, sin utilizar ninguna transformación, se obtiene mediante:

$$\hat{S}_{KM}(t) \pm 1.96ee(\hat{S}_{KM}(t))$$

donde  $ee(\hat{S}_{KM}(t))$  es el error estándar de estimación del estimador de Kaplan y Meier.

#### 1.4. Comparación de las funciones de supervivencia.

La comparación de dos curvas de supervivencias se efectúa a través de contrastes basados en tablas de contingencia como la siguiente:

**Tabla No. 1.** Tabla usada para el contraste de igualdad de funciones de supervivencia en dos grupos en el tiempo de observación  $t_i$

Evento	Grupo		Total
	1	0	
Muerte	$d_1(t_i)$	$d_0(t_i)$	$d(t_i)$
No muerte	$r_1(t_i) - d_1(t_i)$	$r_0(t_i) - d_0(t_i)$	$r(t_i) - d(t_i)$
En riesgo	$r_1(t_i)$	$r_0(t_i)$	$r(t_i)$

Donde por comodidad se han definido los grupos, como 1 y 0, correspondiendo estos grupos a cada una de las dos curvas de supervivencia.

Para construir el estadístico de contraste basta con calcular el número esperado de muertes y la varianza estimada del número de muertes para uno de los grupos; por ejemplo, para el grupo 1 el número esperado de muertes se calcula de la siguiente manera:



$$\hat{e}_1(t_i) = \frac{r_1(t_i)d(t_i)}{r(t_i)}$$

La varianza estimada de  $d_i(t_i)$  está basada en la distribución hipergeométrica y para el grupo 1 está definida como:

$$\hat{V}(d_1(t_i)) = \frac{r_1(t_i)r_0(t_i)(r(t_i)-d(t_i))}{r^2(t_i)(r(t_i)-1)}$$

Finalmente, el estadístico de contraste se define de la siguiente manera:

$$Q = \frac{\left[ \sum_{i=1}^m w_i (d_1(t_i) - \hat{e}_1(t_i)) \right]^2}{\sum_{i=1}^m w_i^2 \hat{V}(d_1(t_i))}$$

Puede demostrarse que el estadístico anterior se puede aproximar mediante una Chi cuadrado de un grado de libertad si el número de ocurrencias de eventos es grande.

Bajo la hipótesis nula que asume que las dos funciones de supervivencia son iguales. En esta fórmula  $m$  es el número de tiempos de ocurrencia de eventos en ambos grupos y  $w_i$  denota los pesos, que toman valores distintos dependiendo del test utilizado. En nuestro caso sólo utilizaremos dos de los casos: el test de Mantel y Haenzel, mas conocido como el test de los rangos de logaritmos (log-rank test) y el test de Peto y Peto. Para una enumeración muy completa de los distintos test, basados en procesos de conteo (Andersen et al, 1993, Fleming y Harrington, 1991).

a) Como establecimos anteriormente, el más común de los test es de Mantel y Haenzel (o log-rank). Este test está diseñado para verificar igualdad o diferencia en la función de supervivencia en todos los tiempos. En este test los pesos son iguales a 1, es decir,  $w_i = 1$ . (Mantel, 1966).

b) Otro de los test comúnmente utilizados es el de Peto y Peto (1972). Este test permite verificar igualdad o diferencia de las funciones de supervivencia en los tiempos iniciales. En este test los pesos toman la forma:

$$w_i = \tilde{S}(t_{i-1}) \frac{r(t_i)}{r(t_i)-1}$$

donde  $\tilde{S}(t)$  es el estimador de la función de supervivencia definida por:

$$\tilde{S}(t) = \prod_{t_i \leq t} \left( \frac{r(t_i) + 1 - d(t_i)}{r(t_i) + 1} \right)$$

c) Otra forma de estudiar los test anteriores fue propuesta por Harrington y Fleming (Harrington y Fleming, 1982, Fleming y Harrington, 1991). Esos dos autores sugieren pesos de la forma:

$$w_i = \left[ \hat{S}_{KM}(t_{i-1}) \right]^\rho$$

y haciendo  $\rho = 0$  se tiene que  $w_i = 1$  (test log-rank) y, si  $\rho = 1$ , se obtiene el test de Peto y Peto. Esta manera de definir los pesos es la forma como trabaja el lenguaje R.

## 1.5. Sobrevida media y mediana.

### 1.5.1. Sobrevida media.

La sobrevida media o media de la supervivencia puede ser estimada mediante la siguiente expresión:

$$\hat{\mu} = \int_0^T \hat{S}_{KM}(t) dt$$

donde T es tiempo máximo de seguimiento observado durante el estudio.

La varianza de la media es:

$$\text{var}(\hat{\mu}) = \int_0^T \left( \int_0^T \hat{S}_{KM}(u) du \right)^2 \frac{dN(t)}{r(t)(r(t) - N(t))}$$

donde  $N(t) = \sum N_i(t) = d(t)$  es el número total de muertes (o de ocurrencia del evento de interés hasta el tiempo  $t$ ) y  $r(t)$  es el número de individuos en riesgo en el tiempo  $t$ .

### 1.5.2. Sobrevida mediana.

La sobrevida mediana o mediana de la supervivencia se define como el primer tiempo  $t$  que satisface la siguiente condición:

$$\hat{S}_{KM}(t) \leq 0.5$$

## 1.6. El modelo de regresión de Cox.

El modelo de regresión de Cox (1972) es el modelo de regresión más utilizado para datos de supervivencia en el área médica.

En el modelo de regresión de Cox, el riesgo para el  $i$ -ésimo individuo se define mediante la siguiente expresión:

$$\lambda(t; Z_i(t)) = \lambda_0(t) e^{\beta' Z_i(t)}$$

donde  $Z_i(t)$  es el vector de covariables para el  $i$ -ésimo individuo en el tiempo  $t$ .

El modelo de Cox establecido anteriormente se dice que es un modelo semiparamétrico debido a que incluye una parte paramétrica y otra parte no paramétrica.

i) La parte paramétrica es  $r_i(t) = e^{\beta' Z_i(t)}$ , llamada puntaje de riesgo (risk score), y  $\beta$  es el vector de parámetros de la regresión.

ii) La parte no paramétrica es  $\lambda_0(t)$  que es llamada función de riesgo base, es una función arbitraria y no especificada.

El modelo de regresión de Cox se llama también modelo de riesgos proporcionales debido a que el cociente entre el riesgo para dos sujetos con el mismo vector de covariables es constante en el tiempo, es decir:

$$\frac{\lambda(t; Z_i(t))}{\lambda(t; Z_j(t))} = \frac{\lambda_0(t) e^{\beta' Z_i(t)}}{\lambda_0(t) e^{\beta' Z_j(t)}} = \frac{e^{\beta' Z_i(t)}}{e^{\beta' Z_j(t)}}$$

Suponiendo que una muerte ha ocurrido en el tiempo  $t^*$ , entonces la verosimilitud de que la muerte le ocurra al individuo  $i$ -ésimo y no a otro individuo es:

$$L_i(\beta) = \frac{\lambda_0(t^*) r_i(t^*)}{\sum_j Y_j(t^*) \lambda_0(t^*) r_j(t^*)} = \frac{r_i(t^*)}{\sum_j Y_j(t^*) r_j(t^*)}$$

El producto de los términos de la expresión anterior  $L(\beta) = \prod L_i(\beta)$  es llamada la verosimilitud parcial y fue introducida por Cox (1972).

La maximización de  $\log(L(\beta))$  da una estimación para  $\beta$  sin necesidad de estimar el parámetro de ruido o función de riesgo base  $\lambda_0(t)$

## 1.7. Contrastes de hipótesis para el modelo de Cox.

Una vez que se ha ajustado un modelo de Cox, existen tres contrastes de hipótesis para verificar la significación del modelo, estos tests son asintóticamente equivalentes, pero no siempre sucede lo mismo en la práctica.

### 1.7.1. Test de razón de verosimilitud.

El primero de los contrastes es el denominado test de razón de verosimilitud y es el que presenta una mayor confiabilidad. Este test se define como:

$$2\left\{\log(L(\beta_0)) - \log(L(\hat{\beta}))\right\}$$

donde  $\beta_0$  son los valores iniciales de los coeficientes y  $\hat{\beta}$  es la solución luego de ajustar el modelo.

### 1.7.2. Test de Wald.

El segundo de los contrastes es conocido como el test de Wald y es quizás el más natural debido a que proporciona un contraste por variables en vez de una medida de significación global. El estadístico de contraste se define mediante:

$$(\hat{\beta} - \beta_0)' \hat{\Sigma}_{\hat{\beta}}^{-1} (\hat{\beta} - \beta_0)$$

donde  $\hat{\Sigma}_{\hat{\beta}}$  es la matriz de varianzas y covarianzas estimada.

### 1.7.3. Test de puntajes (score test).

El tercer contraste es el conocido como test de los puntajes, definido como  $U'IU$ , donde  $U$  es el vector de derivadas del  $\log(L(\beta))$  dado por:

$$U(\beta) = \sum_{i=1}^n \int_0^{\infty} [Z_i(t) - \bar{Z}(\beta, t)] dN_i(t)$$

$I$  es la matriz de información dada por:

$$I(\beta) = \sum_{i=1}^n \int_0^{\infty} \frac{\sum_j Y_j(t) r_j(t) [Z_i(t) - \bar{Z}(\beta, t)] [Z_i(t) - \bar{Z}(\beta, t)]'}{\sum_j Y_j(t) r_j(t)} dN_i(t)$$

y  $\bar{Z}(\beta, t)$  es la media de las covariables para aquellos todavía en riesgo en el tiempo  $t$ , dada por:

$$\bar{Z}(\beta, t) = \frac{\sum_j Y_j(t) r_j(t) Z_i(t)}{\sum_i Y_i(t) r_i(t)}$$

## 1.8. Modelos de Cox estratificados.

Una extensión del modelo de Cox permite obtener la estimación de los modelos para distintos grupos disjuntos o estratos. El modelo obtenido se conoce como modelo de Cox estratificado y está definido para el estrato  $j$ -ésimo como:

$$\lambda(t; Z_i(t)) = \lambda_j(t) e^{\beta' Z_i(t)}$$

Este modelo permite obtener la estimación del modelo en presencia de una variable de estratificación sobre la cual se desean obtener funciones de supervivencia por cada uno de los distintos grupos y probablemente poder estudiar la existencia o no de las funciones de supervivencia entre los grupos.

El modelo de Cox estratificado también constituye una de las maneras de corregir el modelo de Cox cuando no se cumple el supuesto de riesgos proporcionales para alguna de las covariables. En este caso suele correrse el modelo estratificando por la covariable que no cumple con el supuesto de riesgo proporcional. Este procedimiento permite corregir el sesgo en la estimación del parámetro que puede presentarse cuando se viola el supuesto de riesgo proporcional. Sin embargo, presenta una desventaja y es que no existe ningún  $\beta$  que permita estimar el efecto de la covariable de estratificación.

## 1.9. Estudio de residuos en el análisis de supervivencia.

Una de las ventajas que han surgido del enfoque del análisis de supervivencia es la posibilidad de efectuar análisis de residuos (Andersen et al., 1993, Fleming y Harrington, 1991, Therneau y Grambsch, 2000, Therneau et al., 1990).

Los residuos se pueden utilizar para:

1. Descubrir la forma funcional correcta de un predictor continuo.
2. Identificar los sujetos que están pobremente predichos por el modelo.
3. Identificar los puntos o individuos de influencia.
4. Verificar el supuesto de riesgo proporcional.

Existen cuatro tipos de residuos de interés en el modelo de Cox: los residuos de martingala, los de desvíos (deviances), los de puntaje (score) y los de Schoenfeld. De estos cuatro residuos pueden derivarse otros dos: los dfbetas y los residuos escalados de Schoenfeld. A continuación explicaremos brevemente cada uno de estos residuos.

### 1.9.1. Residuos de martingala.

Los residuos de martingala se definen como:

$$\hat{M}_i(t) = N_i(t) - \hat{E}_i(t) = N_i(t) - \int_0^t Y_i(s) e^{\beta'Z_i(s)} d\hat{\Lambda}_0(\beta, s)$$

donde  $\hat{\Lambda}_0(\beta, s)$  es el estimador del riesgo base de Breslow (o de Tsiatis o de Nelson y Aalen) definido como:

$$\hat{\Lambda}_0(\beta, s) = \int_0^s \frac{\sum_{i=1}^n dN_i(s)}{\sum_{i=1}^n Y_i(s) e^{\beta'Z_i(s)}}$$

y están basados en la martingala de un proceso de conteo para el i-ésimo individuo,  $M_i(t) = N_i(t) - E_i(t)$ , definida mediante:

$$M_i(t) = N_i(t) - \int_0^t Y_i(s) e^{\beta'Z_i(s)} \lambda_0(s) ds$$

Los residuos de martingala son muy asimétricos y con una cola muy larga hacia la derecha, particularmente para datos de supervivencia para un solo evento.

Los residuos de martingala se usan para estudiar la forma funcional de una covariable.

### 1.9.2. Residuos de desvíos (deviances).

Los residuos de desvíos se obtienen mediante una transformación de normalización de los desvíos de martingala y son similares en forma a los residuos de desvíos (deviances) en la regresión de Poisson.

Los residuos de desvíos se definen de la manera siguiente: si todas las covariables son fijas en el tiempo, los residuos toman la forma:

$$d_i = \text{signo}(\hat{M}_i) * \sqrt{-\hat{M}_i - N_i \log\left(\frac{N_i - \hat{M}_i}{N_i}\right)}$$

Una expansión de Taylor de un término muestra que:

$$d_i \approx \frac{N_i - \hat{E}_i}{\sqrt{\hat{E}_i}}$$

que es formalmente equivalente a los residuos de Pearson de los modelos lineales generalizados.

Los residuos de desvíos se utilizan para la detección de valores atípicos (outliers).

### 1.9.3. Residuos de puntajes (scores).

Los residuos de puntajes se definen como:

$$U_{ij} = U_{ij}(\hat{\beta}, \infty)$$

donde  $U_{ij}(\beta, t)$ ,  $j = 1, \dots, p$  son las componentes del vector fila de longitud  $p$  obtenido a través del proceso de puntaje para el  $i$ -ésimo individuo:

$$U_i(\beta) = \int_0^t [Z_i(t) - \bar{Z}(\beta, t)] dN_i(t)$$

Los residuos de puntajes se utilizan para verificar la influencia individual y para la estimación robusta de la varianza.

### 1.9.4. Residuos de Schoenfeld.

Los residuos de Schoenfeld (1982) se definen como la matriz:

$$s_{ij}(\beta) = Z_{ij}(t_i) - \bar{Z}_j(\beta, t_i)$$

con una fila por muerte y una columna por covariable, donde  $i$  y  $t_i$  son los individuos y el tiempo de ocurrencia del evento respectivamente.

Los residuos de Schoenfeld son útiles para la verificación del supuesto de riesgo proporcional en el modelo de Cox.

## 1.10. Interpretación del modelo de Cox.

La interpretación del modelo de Cox no se hace directamente a través de su coeficiente estimado sino del exponencial de la estimación del coeficiente estimado,  $\exp(\hat{\beta})$ .

Para variables dicotómicas  $\exp(\hat{\beta})$  es un estimador de la razón de riesgos (hazard ratio) y se interpreta como la cantidad de riesgo que se tiene con la presencia de cada covariable en relación a la ausencia del resto de las la covariables.

Los intervalos de confianza del 95% para  $\exp(\hat{\beta})$  se obtienen mediante:

$$\exp(\hat{\beta} \pm 1.96 ee(\hat{\beta}))$$

donde  $ee(\hat{\beta})$  es el error estándar de  $\hat{\beta}$ .

Para el caso de covariables continuas,  $\exp(\hat{\beta})$  representa la razón de riesgos (hazard ratio) al incrementar en una unidad la covariable.

Resulta más interesante estimar la razón de riesgos al incrementar la covariable en  $c$  unidades y esto se hace mediante  $\exp(c\hat{\beta})$ , siendo su intervalo de confianza del 95% de la forma:

$$\exp(c\hat{\beta} \pm 1.96|c| ee(\hat{\beta}))$$

Para una explicación más detallada puede verse Hosmer y Lemeshow (1999).



## Parte 2. Aspectos relacionados con el lenguaje R.

En esta sección se presenta una introducción al lenguaje R para luego continuar con el estudio de las funciones de R que permiten llevar a cabo un análisis de supervivencia, posteriormente se presenta un ejemplo completo de un análisis de supervivencia utilizando el lenguaje R..

### 2.1. Introducción al lenguaje R.

El lenguaje R es un lenguaje de computación formal diseñado para ser utilizado en la manipulación y análisis de datos que posee una serie de facilidades gráficas. Es además un lenguaje de licencia gratuita.

El lenguaje R puede ser considerado como un programa orientado a objetos debido a que la filosofía del lenguaje es que el resultado de la evaluación de cada función genera un objeto, que posee a su vez una serie de atributos.

El lenguaje R puede ser utilizado de manera interactiva, pudiéndose obtener resultados con cada línea de comandos, esta característica la hace diferente de otros paquetes importantes como los son el sistema SAS y el SPSS en su versión de programación.

Es además, un paquete con mucho potencial para el desarrollo de dispositivos gráficos y posee además un buen nivel de manipulación de datos pero quizás en este aspecto no es tan poderoso como por ejemplo el SAS, lo cual no constituye un problema ya que la importación y exportación de datos desde y hacia otros sistemas está completamente resuelta, también existe la posibilidad de comunicarse con otros lenguajes poderosos como lo son PERL y PYTHON, con los cuales las posibilidades de manejos con cualquier clase de estructuras de datos son prácticamente ilimitadas.

El Lenguaje R es una iniciativa de desarrollo escrito inicialmente en 1996 por Robert Gentleman y Ross Ihaka de la Universidad de Auckland de Nueva Zelandia que se basó en el ambiente del lenguaje S. El lenguaje S es un lenguaje ideado a finales de los ochenta por personas ligadas a los Laboratorios Bell (Chambers, Becker y otros) y que fue comercializado con el nombre de S-PLUS por la compañía de software de Seattle STATSCI. Posteriormente fue comercializado por MATHSOFT, por LUCENT TECHNOLOGY y más recientemente por INSIGHTFUL CORPORATION.

Recientemente (04 de Octubre de 2004), fue liberada la versión 2.0.0 del lenguaje R (R Development Core Team, 2004), esta versión tiene incorporado el sistema básico y 25 paquetes considerados como estándares y recomendados. El acceso a la base del lenguaje puede hacerse a través de la página principal del proyecto R (<http://www.r-project.org>) o a través de los servidores espejos (mirror sites) de la red Comprehensive R Archive Network (<http://cran.r-project.org>).

Otra ventaja del lenguaje R es la gran cantidad de paquetes contribuidos disponibles en la página de CRAN, actualmente hay disponibles más de 400. Un aspecto importante de resaltar es que cada paquete viene acompañado de su manual en formato pdf.

### **2.1.1. Instalación del lenguaje R.**

El lenguaje R bajo Windows, se instala ejecutando el archivo `rw2000.exe` y siguiendo las instrucciones de instalación, este archivo se puede bajar de cualquiera de los servidores de CRAN, este archivo ejecutable tiene un tamaño de 23 Mb y ocupa un espacio en disco máximo de 49 Mb, en su instalación completa (full). Los requerimientos de equipo no son muy exigentes, puede ser instalado en equipos de la familia x86 o superiores y funcionan con los sistemas operativos Microsoft Windows superiores a las versiones 3.11.

### **2.1.2. Instalación de los paquetes adicionales.**

Los paquetes contribuidos se instalan directamente desde el ambiente de trabajo del lenguaje R, utilizando el menú Packages. Esto puede hacerse de dos maneras:

- i) Directamente de las páginas de CRAN, para lo cual hay que estar conectado a la internet.
- ii) A través de los archivos ejecutables previamente bajados en formato zip, en esta opción no es necesario estar conectado a internet.

Un aspecto interesante es que los paquetes que han sido instalados previamente pueden ser actualizados directamente de la página de CRAN, esta facilidad solo se puede utilizar si se está conectado a internet.

### **2.1.3. Ayudas y documentación del R.**

R es un lenguaje que ofrece varios niveles de ayuda y estas 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("tópico")` para obtener ayuda acerca del tópico específico, el inconveniente de esta instrucción es que hay que colocar exactamente el nombre del tópico.

La instrucción `help.search("tópico")` es más flexible porque se realiza una búsqueda del tópico en todas los paquetes que han sido bajados a al R de la máquina.

Las dos opciones anteriores también están disponibles a través del menú Help (R functions (text)... y, Search help..., respectivamente.

En el menú Help también están disponibles una excelente ayuda en formato HTML, en el cual se van incorporando las ayudas de los paquetes que se van incorporando.

Otra forma de ayuda son los manuales en formato pdf, teniendo disponible a través del menú Help los relativos a la base del R. Los manuales del resto de los paquetes pueden accederse mediante el acrobat reader.

Existe también una serie de documentación no oficial de R que está disponible a través de la página de CRAN. Esta documentación es considerada como contribuida y hay material en diversos idiomas, incluyendo al castellano.

### **2.1.4. Acceso a datos internos disponibles.**

Una gran parte de los paquetes tiene disponible una serie de datos que pueden ser trabajados, sobre todo en la etapa de aprendizaje del lenguaje R. La verificación de los datos disponibles en todos los paquetes disponibles en el R que se está trabajando puede verse mediante la instrucción `data()` y para ver los datos de un paquete en específico debe escribirse la opción `data("paquete")`.

### **2.1.5. Acceso a datos externos disponibles.**

La forma más sencilla de acceder a datos externos es mediante la función `read.table` para lo cual se recomienda tener los datos en un archivo de texto delimitado por algún carácter, por defecto el espacio. Se recomienda además tener los nombres de las variables en la primera fila en cuyo caso hay que colocar la opción `header=TRUE`.

También existen facilidades para importar (y exportar) datos desde (y hacia) otros sistemas, una opción interesante es la del paquete *foreign*, que permite importar y exportar datos de Minitab, S, SAS, SPSS y Stata, entre otros.

### **2.1.6. La opción de asignación.**

Todo comando que se ejecuta en R produce un resultado y para poder tener disponible este resultado debe hacerse un proceso de asignación, esto se hace mediante los caracteres menor que (`<`) y el guión (`-`), generando el efecto visual `<-`, el proceso de asignación se hace asignando el valor o el objeto a la derecha de `<-` al objeto cuyo nombre es colocado a la izquierda de `<-`. El nombre del objeto al cual se le hace la asignación puede incluir cualquier carácter alfanumérico, incluyendo puntos y se recomienda que comience con un letra. El formato de la asignación es:

```
Nombre.del.objeto<-(función u objeto generado)
```

### **2.1.7. Verificación de objetos disponibles.**

Los objetos disponibles pueden ser verificados mediante la instrucción `objects()`.

### **2.1.8. Eliminación de objetos no deseados.**

Cada proceso de asignación va generando un nuevo objeto, en caso de no necesitar más un objeto este debe borrarse utilizando la instrucción `rm("Nombre objeto a eliminar")`. En el caso de que se quieran eliminar varios objetos, pueden colocarse todos ellos dentro del paréntesis separados por comas.

### **2.1.9. R diferencia las mayúsculas de las minúsculas.**

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.

### **2.1.10. Datos faltantes en R.**

Los datos faltantes para variables numéricas en R se suele especificar con el valor NA. No todas las funciones admiten la presencia de datos faltantes por lo que hay que revisar primero la documentación disponible para la función o efectuar pruebas correspondientes.

### **2.1.11. Comentarios en R.**

El lenguaje R admite comentarios. Un comentario comienza con el carácter numeral (`#`), considerándose como comentario a todo lo que aparezca en la línea de comando a la derecha de `#`.

### **2.1.12. Creación de datos en R.**

El lenguaje R permite la creación de diversas estructuras de datos para el caso de que se tengan pocas variables y pocos individuos el proceso de creación puede hacerse a través de una lista de combinaciones por columna, mediante la instrucción:

```
list(cbind(var1=c(valor1,...,valorn),...,vark=c(valor1,...,valorn)))
```

Otra opción es la creación de vectores por separado y unirlos por columnas mediante la función `cbind` y luego eliminar los objetos de las variables (vectores) pero este procedimiento es menos eficiente que anterior.

Otra forma de crear datos es crearlos con un software externo, por ejemplo Microsoft excel, guardarlos como archivos de texto delimitados, teniendo cuidado de que el separador de enteros y decimales debe ser un punto, luego copiarlos en el sudirectorio del R y leerlos mediante la función `read.table`.

### 2.1.13. Carga y descarga de objetos.

Algunas funciones de R, necesitan tener accesible el objeto que contiene las variables a analizar, la carga en el ambiente se hace mediante la instrucción `attach(objeto)` y la descarga mediante la instrucción `dettach(objeto)`.

### 2.1.14. Envío de gráficos a otros programas.

El envío de gráficos a otros programas se hace colocando el gráfico en el pisapapeles. Existen dos opciones para esto, colocándolo como `metafile` o como `bitmap`, el ícono de la cámara lo coloca como `metafile`.

### 2.1.15. Salida del lenguaje R.

Para salir del lenguaje R, se debe escribir la instrucción `q()`, seleccionar la opción Exit del menú File o hacer clic en la `x` colocada en el extremo superior derecho de la ventana. En este momento se pregunta si se desea guardar el espacio de trabajo, en caso de seleccionar si se graban de manera definitiva los objetos y los comandos que se han generado durante la sesión y si se selecciona no, se pierde la información de la sesión por lo que debe estar muy atento para tomar la decisión al momento de cerrar la sesión de R.

## 2.2. Análisis de supervivencia utilizando el lenguaje R.

Existen varios paquetes que permiten llevar a cabo análisis de supervivencia, el más utilizado, que también tiene el estatus de recomendado es el *survival*, cuya versión 2.13.2 está disponible desde el pasado 4 de Octubre para la versión de R 2.0.0. El paquete *survival* es una librería desarrollada por Thomas Lumley a partir del código para S desarrollado por Terry Therneau (S original by Terry Therneau and ported by Thomas Lumley, 2004), puede verse también el texto de Therneau y Grambsch (2000). Existen otros paquetes en CRAN que hacen análisis de supervivencia o proporcionan ejemplos usados en la literatura, entre ellos se encuentran el *eha* para análisis de historias de eventos, el *emplink* que permite calcular razones de verosimilitud para datos censurados y truncados, el *KMsurv* que contiene los datos del libro de Klein y Moeschberger (1997), el *msm* que trabaja con modelos de Markov de múltiple estados continuos en el tiempo y que son útiles para algunos modelos multivariados, el *muha* que hace estimaciones de la función de riesgos, el *smoothSurv* que trabaja con modelos de regresión con distribuciones de errores suavizadas y el *survrec* ideado para modelos de eventos recurrentes.

### 2.2.1. El paquete *survival*.

El paquete *survival* permite llevar a cabo análisis de supervivencia para datos que presentan diversos mecanismos de censura. Este es un paquete que tiene la característica de ser un paquete con el estatus de recomendado el cual ya viene incorporado en la versión 2.0.0 de R (versión 2.13.2), sin embargo, es recomendable estar atentos para actualizarlo de manera permanente. Algunas de las funciones de este serán descritas brevemente en la subsecciones siguientes. Para ejecutar cualquiera de las funciones de este paquete es necesario invocar la librería mediante la instrucción:

```
library(survival)
```

#### 2.2.1.1 La función *surv*.

La función *Surv* permite crear objetos tipo *survival*, la estructura para datos que presentan censura por la derecha es:

```
Surv(time, event)
```

En la cual *time* representa el tiempo y *event* representa el estatus de censura considerados como cero (0) para datos censurados y uno (1) cuando el evento es observado.

Una estructura más completa de la función *Surv*, que es útil para otros tipos de censuras es:

```
Surv(time, time2, event, type=, origin=0)
```

En donde *time* representa el tiempo de inicio de la observación, *time2* el tiempo de finalización, se asume que los intervalos de tiempos son abiertos en su extremo inferior y cerrados en su tiempo superior, es decir (*time, time2*], *event* es la condición de ocurrencia del evento que depende del tipo de censura (*type*), que por defecto es censura por la derecha, y *origin* es una utilidad que permite trabajar bajo el enfoque de los procesos de conteo.

#### 2.2.1.2. La función *survfit*.

La función *survfit* permite obtener estimación de la función de supervivencias utilizando el método de Kaplan y Meier (opción por defecto) o de Fleming y Harrington. También permite predecir la función de supervivencia para modelos de Cox. La estructura de la función *survfit* es:

```
survfit(formula, data, weights, subset, na.action,  
        newdata, individual=F, conf.int=.95, se.fit=T,  
        type=c("kaplan-meier", "fleming-harrington", "fh2"),  
        error=c("greenwood", "tsiatis"),  
        conf.type=c("log", "log-log", "plain", "none"),  
        conf.lower=c("usual", "peto", "modified"))
```

la cual posee una serie de opciones que pueden ser revisadas en el manual (Lumley, 2004) o en la ayuda (`help("survfit")`)

Con la función `survfit` puede obtenerse diversa información:

Con `print(survfit( . . . ))` o directamente con `survfit( . . . )` se obtienen las medidas resumen.

Con `summary(survfit( . . . ))` se obtiene la función de supervivencia estimada.

Con `plot(survfit( . . . ))` se obtiene el gráfico de la función de supervivencia estimada. En esta función pueden controlarse un serie de opciones gráficas, se recomienda ver la ayuda correspondiente para más detalles.

Con `names(survfit( . . . ))` se obtiene el nombre de cada uno de los atributos de la función `survfit`. Esta función es útil para seleccionar atributos por separados o para realizar cálculos posteriores cuando sea necesario.

### 2.2.1.3. La función `survdiff`.

La función `survdiff` permite efectuar contrastes de hipótesis para verificar la igualdad o diferencia de dos o más curvas de supervivencias, basados en las familias de pruebas G-rho propuestas por Fleming y Harrington (1982). La estructura de la función `survdiff` es:

```
survdiff(formula, data, subset, na.action, rho=0)
```

Para más detalles ver la ayuda correspondiente (`help("survdiff")`).

### 2.2.1.4. La función `coxph`.

La función `coxph` permite ajustar modelos de regresión de Cox. Permite también ajustar modelos con variables dependientes del tiempo, modelos estratificados, modelos de múltiples eventos por individuo y otras extensiones derivadas del enfoque basado en los procesos de conteo. La estructura de la función `coxph` es:

```
coxph(formula, data=parent.frame(), weights, subset,
      na.action, init, control, method=c("efron", "breslow", "exact"),
      singular.ok=TRUE, robust=FALSE,
      model=FALSE, x=FALSE, y=TRUE, ...
      )
```

Para mayores detalles puede consultar la ayuda correspondiente.

La función `coxph` puede combinarse con otras funciones que permiten obtener la siguiente información:

Con `print(coxph( . . . ))` o directamente con `coxph( . . . )` se obtienen los contrastes para verificar la adecuación del modelo de Cox ajustado. Con `summary(coxph( . . . ))` se obtiene un poco más de detalles de los contrastes.

Con `summary(survfit(coxph( . . . )))` se obtiene la función de supervivencia ajustada por el modelo de Cox.

Con `plot(survfit(coxph( . . . )))` se obtiene el gráfico de la función de supervivencia ajustada por el modelo de Cox. En esta función pueden controlarse un serie de opciones gráficas, se recomienda ver la ayuda correspondiente para más detalles.

Con `names(coxph( . . . ))` se obtiene el nombre de cada uno de los atributos de la función `coxph`. Esta función es útil para seleccionar atributos por separados o para realizar cálculos posteriores cuando sea necesario.

Otra funciones importantes que funcionan con la función `coxph` son lo son las funciones `cox.zph` y la función `residuals` (o `resid`), las cuales serán explicadas en la próximas dos secciones.

### 2.2.1.5. La función `cox.zph`.

La función `cox.zph` permite llevar a cabo el contraste de hipótesis de riesgos proporcionales, las salidas directas presentan el contraste global y de cada una de las covariables en el modelo. La hipótesis nula es el cumplimiento del supuesto de riesgos proporcionales, asociado a que los betas son ceros. La estructura de esta función es:

```
cox.zph(fit, transform="km", global=TRUE)
```

Esta función puede combinarse con la función `plot` para obtener la distribución de los betas, para lo cual se utiliza la siguiente estructura:

```
plot(x, resid=TRUE, se=TRUE, df=4, nsmo=40, var, ...)
```

donde `x` es un objeto de tipo `cox.zph` y `var` permite identificar la covariable que se va a representar de forma gráfica, en S-PLUS no es necesario hacer esta declaración porque se genera un gráfico compuesto donde se representan los gráficos para cada una de las covariables, consulte la ayuda para detalles adicionales. Para entender un poco más como funciona este comando puede ver el ejemplo del presente curso.

### 2.2.1.6. La función `residuals`.

Otra función importante asociada a los objetos del tipo `coxph` es la función `residuals`, o en su formato más corto `resid`. Esta función permite calcular los residuos de martingala, de puntajes (score), de tipo desvío (deviance) y de Schoenfeld. La estructura de esta función es:



```
residuals(object,  
  type=c("martingale", "deviance", "score", "schoenfeld",  
        "dfbeta", "dfbetas", "scaledsch", "partial"),  
  collapse=FALSE, weighted=FALSE, ...)
```

donde `object` es un objeto de tipo `coxph`.

Para mayores detalles consulte la ayuda correspondiente y para ver la implementación puede verse el ejemplo de este curso.

### 2.3. Ejemplo de análisis de supervivencia utilizando el lenguaje R.

En esta parte se presenta un análisis completo de un análisis de supervivencia utilizando el lenguaje R, para ello se ha trabajado con una versión reducida de los datos correspondientes a los pacientes de diálisis peritoneal analizados en la tesis de maestría de Borges (2002), que se encuentran en el archivo `dpa.txt`.

#### 2.3.1. Estructura del archivo de datos `dpa.txt`.

El archivo `dpa.txt` contiene la información de 246 pacientes que acudieron al Servicio de diálisis peritoneal del Hospital Clínico Universitario de Caracas entre los años 1980 y 2000. La variables seleccionadas para este archivo fueron:

orden:	Orden de los individuos en la base de datos.
sexofm:	Sexo (0 corresponde al sexo femenino y 1 al sexo masculino)
diabetes:	Diabetes mellitus (1 corresponde a un paciente diabético y 0 a uno no diabético)
meses:	Meses de seguimiento en diálisis peritoneal.
sensor2:	Condición de Censura (1 denota la muerte y 0 denota los datos censurados)
edad:	Edad del paciente al comienzo de la diálisis.
quetelet:	Índice de Quetelet.

#### 2.3.2. Lectura de los datos en R.

La lectura de los datos y su correspondiente asignación al objeto `dpa` se hace mediante la instrucción:

```
> dpa<-read.table("dpa.txt",header=TRUE)
```

Posteriormente y antes de comenzar a ejecutar las funciones del paquete *survival* debe ejecutarse el comando

```
> library(survival)
```

### 2.3.3. Estimación de la función de supervivencia a través del estimador de Kaplan y Meier.

La obtención del objeto que contiene la información de la estimación de la función de supervivencia a través del método de Kaplan y Meier se hace, luego de cargar el objeto `dpa` (`attach(dpa)`) mediante la instrucción:

```
> kml<-survfit(Surv(meses,censor2))
```

Para visualizar el resumen de la estimación debe escribirse la instrucción `print(kml)` (o simplemente `kml`), obteniéndose la siguiente salida:

```
Call: survfit(formula = Surv(meses, censor2))
```

n	events	median	0.95LCL	0.95UCL
246	64	61	55	Inf

Mediante el comando `names(kml)` se obtienen los nombres del objeto `kml`, teniendo la apariencia siguiente:

```
[1] "n"           "time"        "n.risk"      "n.event"    "surv"        "type"        "std.err"
[8] "upper"      "lower"       "conf.type"  "conf.int"   "call"
```

La estimación de la función de supervivencia se obtiene mediante la instrucción:

```
> summary(kml)
```

Obteniéndose la siguiente salida:

```
Call: survfit(formula = Surv(meses, censor2))
```

time	n.risk	n.event	survival	std.err	lower	95% CI	upper	95% CI
0	246	2	0.992	0.00573	0.9807	1.000		
1	240	1	0.988	0.00704	0.9740	1.000		
3	228	4	0.970	0.01102	0.9490	0.992		
4	221	1	0.966	0.01182	0.9431	0.989		
5	215	1	0.962	0.01259	0.9372	0.987		
6	209	1	0.957	0.01334	0.9311	0.983		
7	202	1	0.952	0.01409	0.9250	0.980		
8	197	1	0.947	0.01483	0.9187	0.977		
9	193	1	0.942	0.01554	0.9125	0.973		
10	188	1	0.937	0.01625	0.9061	0.970		
11	180	3	0.922	0.01831	0.8866	0.958		
12	171	1	0.916	0.01898	0.8800	0.954		
13	161	1	0.911	0.01970	0.8729	0.950		
14	151	4	0.887	0.02257	0.8435	0.932		
15	144	1	0.880	0.02324	0.8361	0.927		
17	135	3	0.861	0.02532	0.8127	0.912		
19	124	1	0.854	0.02605	0.8044	0.907		
20	119	2	0.840	0.02752	0.7873	0.895		
21	115	3	0.818	0.02956	0.7617	0.878		
22	110	3	0.795	0.03143	0.7361	0.859		
23	104	1	0.788	0.03205	0.7274	0.853		
25	94	1	0.779	0.03278	0.7177	0.846		
26	90	1	0.771	0.03354	0.7077	0.839		

## Análisis de supervivencia básico utilizando el lenguaje R

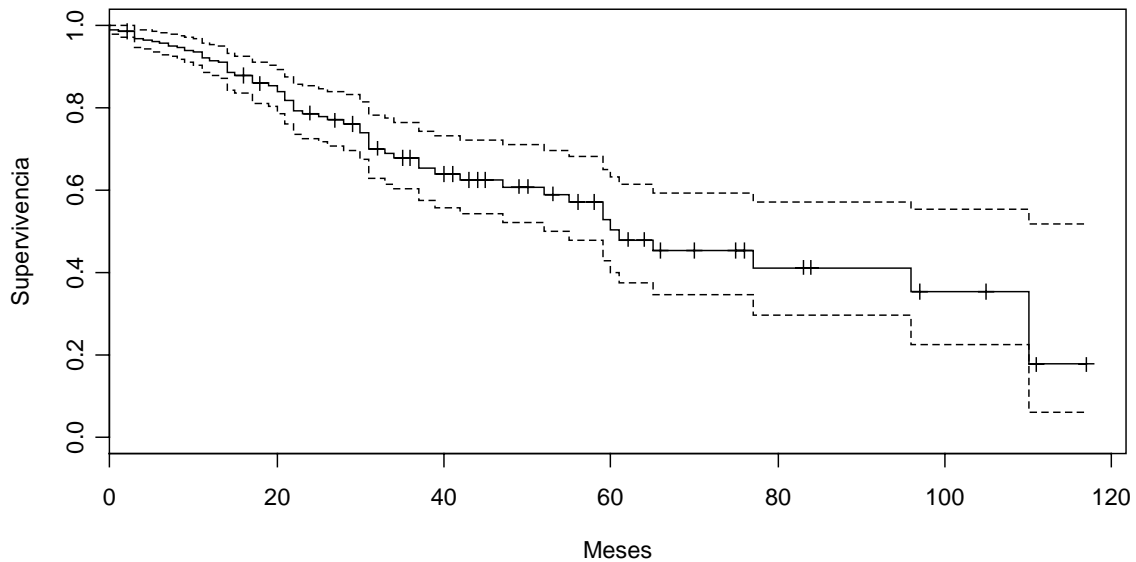
28	81	1	0.761	0.03445	0.6966	0.832
30	78	2	0.742	0.03623	0.6739	0.816
31	75	4	0.702	0.03933	0.6291	0.784
33	63	1	0.691	0.04025	0.6164	0.775
34	59	1	0.679	0.04124	0.6031	0.765
37	53	2	0.654	0.04348	0.5737	0.745
39	50	1	0.641	0.04453	0.5589	0.734
42	43	1	0.626	0.04592	0.5418	0.722
47	38	1	0.609	0.04757	0.5227	0.710
52	33	1	0.591	0.04958	0.5011	0.696
55	31	1	0.572	0.05152	0.4791	0.682
59	26	2	0.528	0.05616	0.4283	0.650
60	22	1	0.504	0.05850	0.4012	0.632
61	21	1	0.480	0.06044	0.3748	0.614
65	18	1	0.453	0.06268	0.3455	0.594
77	11	1	0.412	0.06920	0.2963	0.573
96	7	1	0.353	0.08054	0.2258	0.552
110	4	2	0.177	0.09701	0.0601	0.518

Finalmente, el gráfico de la función de supervivencia puede construirse mediante el comando:

```
> plot(km1,xlab="Meses",ylab="Supervivencia", main="Gráfico No. 1.  
Estimador de Kaplan y Meier")
```

Obteniéndose el siguiente gráfico:

**Gráfico No. 1. Estimador de Kaplan y Meier**

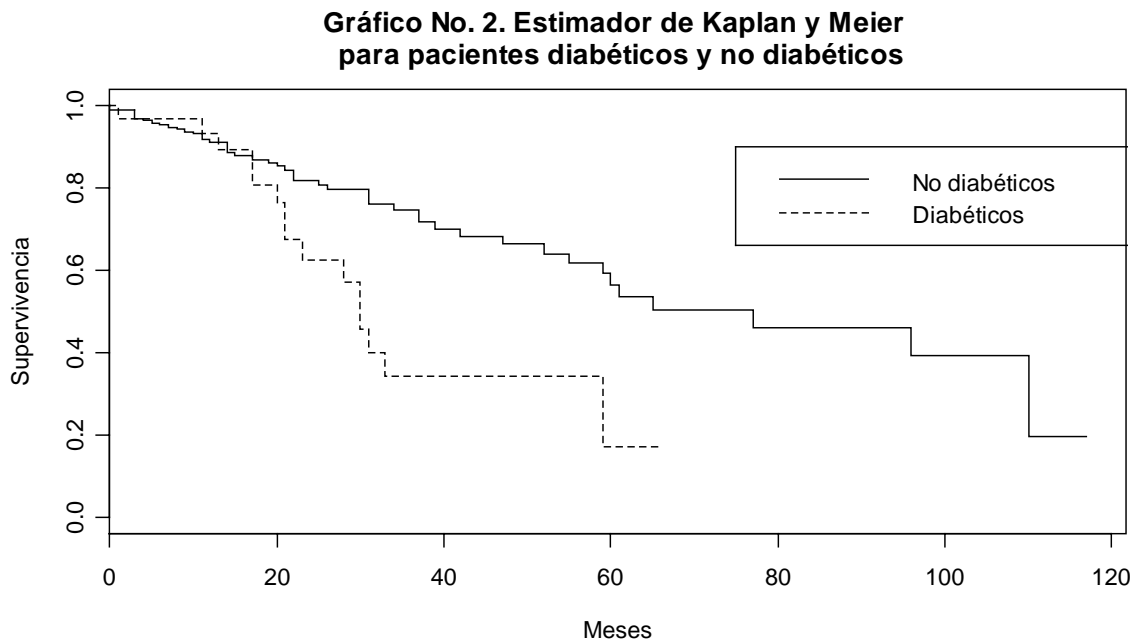


### 2.3.4. Comparación de funciones de supervivencia.

Suponga que queremos comparar las funciones de supervivencia de los pacientes diabéticos y los no diabéticos, para ello construyamos un gráfico donde se observe las estimaciones de Kaplan y Meier para los pacientes diabéticos y no diabéticos, esto lo haremos mediante las instrucciones:

```
> km2<-survfit(Surv(meses,censor2)~diabetes)
> plot(km2,xlab="Meses",ylab="Supervivencia", main="Gráfico No. 2.
Estimador de Kaplan y Meier \n para pacientes diabéticos y no
diabéticos",lty=c(1,2),mark.time=FALSE)
> legend(75,0.9,legend=c("No diabéticos","Diabéticos"),lty=c(1,2))
```

Obteniéndose el gráfico:



Donde se observa que aparentemente ambas funciones de supervivencia son distintas.

Para comparar ambas funciones de supervivencia se debe ejecutar el comando:

```
> survdiff(Surv(meses,censor2)~diabetes)
```

Obteniéndose el resultado:

```
Call:
survdiff(formula = Surv(meses, censor2) ~ diabetes)

      N Observed Expected (O-E)^2/E (O-E)^2/V
diabetes=0 212      49    56.46    0.986    8.6
diabetes=1  34      15     7.54    7.386    8.6

Chisq= 8.6 on 1 degrees of freedom, p= 0.00335
```

Y como  $p=0.00335 < 0.05$ , se rechaza la hipótesis nula de igualdad de funciones de supervivencia (para un nivel de significación del 5%).

### 2.3.5. Ajuste del modelo de Cox.

Ajustemos un modelo de Cox con diabetes, edad e índice de quetelet como covariables y asignemos el ajuste al objeto con nombre `cox1`, esto lo hacemos, mediante la instrucción:

```
> cox1<-coxph(Surv(meses,censor2)~diabetes + edad + quetelet, data=dpa,
na.action=na.exclude)
```

Y al ejecutar el comando `print(cox1)` (o simplemente `cox1`), con lo que obtenemos el siguiente resultado:

```
Call:
coxph(formula = Surv(meses, censor2) ~ diabetes + edad + quetelet,
      data = dpa, na.action = na.exclude)

            coef exp(coef) se(coef)      z      p
diabetes  0.5491     1.732   0.3208  1.71 0.0870
edad      0.0315     1.032   0.0097  3.25 0.0011
quetelet -0.0969     0.908   0.0389 -2.49 0.0130

Likelihood ratio test=18.8 on 3 df, p=0.000308 n=233 (13 observations deleted due to missing)
```

Que es una salida en donde la significación del modelos puede verificarse sólo a través del método de la razón de verosimilitud. Una salida más completa se presenta mediante la ejecución del comando `summary(cox1)`, la cual tiene la forma:

```
Call:
coxph(formula = Surv(meses, censor2) ~ diabetes + edad + quetelet,
      data = dpa, na.action = na.exclude)

n=233 (13 observations deleted due to missing)

            coef exp(coef) se(coef)      z      p
diabetes  0.5491     1.732   0.3208  1.71 0.0870
edad      0.0315     1.032   0.0097  3.25 0.0011
quetelet -0.0969     0.908   0.0389 -2.49 0.0130

            exp(coef) exp(-coef) lower .95 upper .95
diabetes     1.732      0.577    0.923     3.25
edad         1.032      0.969    1.013     1.05
quetelet     0.908      1.102    0.841     0.98

Rsquare= 0.077 (max possible= 0.892 )
Likelihood ratio test= 18.8 on 3 df, p=0.000308
Wald test              = 19.4 on 3 df, p=0.000229
Score (logrank) test = 19.8 on 3 df, p=0.000184
```

Con esta cual podemos concluir que el modelo es significativo cualquiera de los tres criterios (test de razón de verosimilitud, test de Wald y test de los puntajes (score o logrank))

La salida anterior también nos permite verificar la significación de cada uno de los coeficientes correspondientes a las covariables, observándose que el coeficiente correspondiente a diabetes es significativo al 10%, el de la edad lo es al 1% y el del índice de Quetelet lo es al 5%.

Otra información importante, obtenida directamente a través de salida anterior es la estimación de los riesgos relativos (a partir de los  $\exp(\text{coef})$ ), con los cuales podemos decir que la presencia de diabetes hace que la muerte tenga un riesgo de 1.732 veces el riesgo de muerte de los no diabéticos. En cuanto a la edad, una persona con una edad determinada tiene 1.032 veces el riesgo de morir en relación a una persona una año menor, para esta covariable, la interpretación pudiera también darse en relación a quinquenios obteniéndose un riesgo relativo de  $1.17 = \exp(5 * 0.0315)$ . Finalmente, al aumentar el índice de Quetelet en una unidad el riesgo se hace 0.908 veces que la del menor valor.

Mediante el comando:

```
> summary(survfit(cox1))
```

Puede obtenerse la función de supervivencia ajustada mediante el modelo de Cox, cuya salida es :

```
Call: survfit.coxph(object = cox1)
```

time	n.risk	n.event	survival	std.err	lower 95% CI	upper 95% CI
0	233	2	0.993	0.00516	0.983	1.000
3	220	2	0.985	0.00751	0.970	1.000
4	215	1	0.981	0.00847	0.965	0.998
5	209	1	0.977	0.00938	0.959	0.996
6	203	1	0.973	0.01024	0.953	0.993
7	196	1	0.969	0.01109	0.947	0.991
9	188	1	0.965	0.01195	0.941	0.988
10	183	1	0.960	0.01278	0.935	0.985
11	175	3	0.946	0.01520	0.917	0.976
12	166	1	0.941	0.01597	0.910	0.973
13	156	1	0.936	0.01678	0.904	0.970
14	148	4	0.915	0.01996	0.877	0.955
15	142	1	0.910	0.02069	0.870	0.951
17	133	3	0.893	0.02297	0.849	0.939
19	122	1	0.887	0.02378	0.841	0.935
20	117	2	0.874	0.02539	0.826	0.925
21	113	3	0.855	0.02768	0.802	0.911
22	108	3	0.835	0.02992	0.778	0.896
23	102	1	0.828	0.03068	0.770	0.890
25	92	1	0.820	0.03156	0.760	0.884
26	88	1	0.812	0.03249	0.751	0.878
28	79	1	0.803	0.03356	0.740	0.872
30	76	2	0.785	0.03566	0.718	0.858
31	73	4	0.747	0.03959	0.673	0.829
33	62	1	0.736	0.04074	0.660	0.820
34	58	1	0.725	0.04195	0.647	0.812
37	52	2	0.699	0.04474	0.616	0.792
39	49	1	0.685	0.04609	0.600	0.781
42	42	1	0.669	0.04781	0.582	0.770
47	37	1	0.652	0.04975	0.562	0.758
52	32	1	0.634	0.05207	0.539	0.744
55	30	1	0.614	0.05423	0.517	0.730
59	25	2	0.569	0.05965	0.463	0.699
60	21	1	0.544	0.06245	0.434	0.681
61	20	1	0.519	0.06474	0.406	0.663

## Análisis de supervivencia básico utilizando el lenguaje R

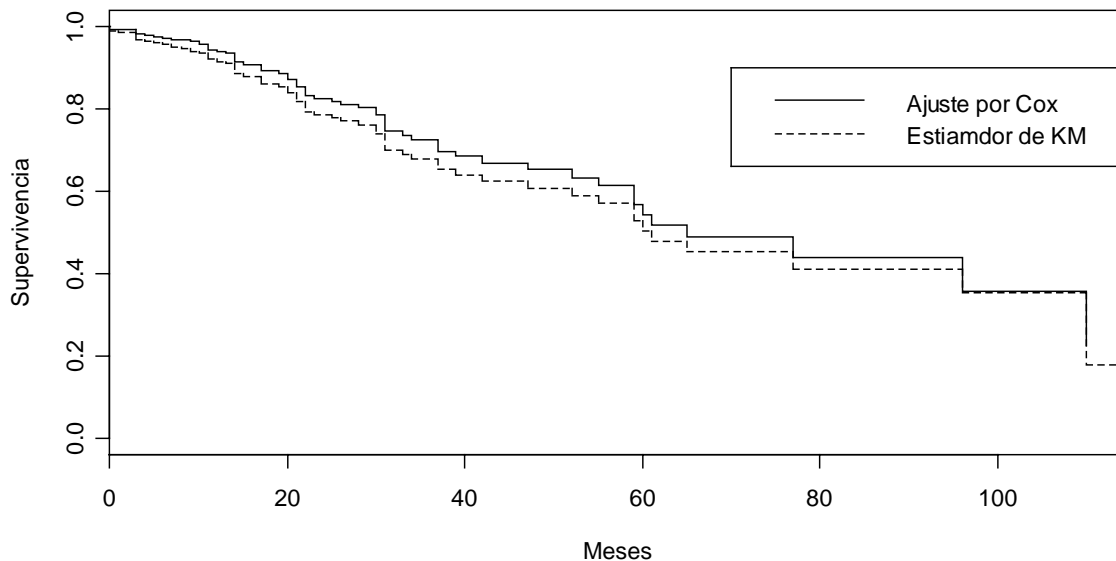
65	17	1	0.490	0.06759	0.374	0.642
77	10	1	0.440	0.07749	0.311	0.621
96	6	1	0.356	0.09776	0.208	0.610
110	3	1	0.224	0.12059	0.078	0.643

También puede obtenerse la gráfica correspondiente, pero en esta ocasión graficaremos la función de supervivencia obtenida mediante el estimado de Kaplan y Meier y la obtenida mediante el modelo de Cox, esto lo podemos hacer mediante los comandos:

```
> plot(survfit(cox1),conf.int=FALSE,main="Gráfico No. 3. Comparación del
ajuste del modelo de Cox \n y el estimador de
KM",xlab="Meses",ylab="Supervivencia")
> lines(km1,lty=2)
> legend(70,0.9,legend=c("Ajuste por Cox","Estiamdor de KM"),lty=c(1,2))
```

Obteniéndose la gráfica

**Gráfico No. 3. Comparación del ajuste del modelo de Cox y el estimador de KM**



Pudiéndose observar que el ajuste del modelo de Cox es sistemáticamente superior a la función de supervivencia de Kaplan y Meier.

### 2.3.6. Verificación de los supuestos del modelo de Cox.

#### 2.3.6.1. Supuesto de riesgos proporcionales.

El supuesto de riesgos proporcionales puede ser verificado mediante el contraste de hipótesis generado mediante el comando:

```
> cox.zph(cox1)
```

del cual se obtiene la salida:

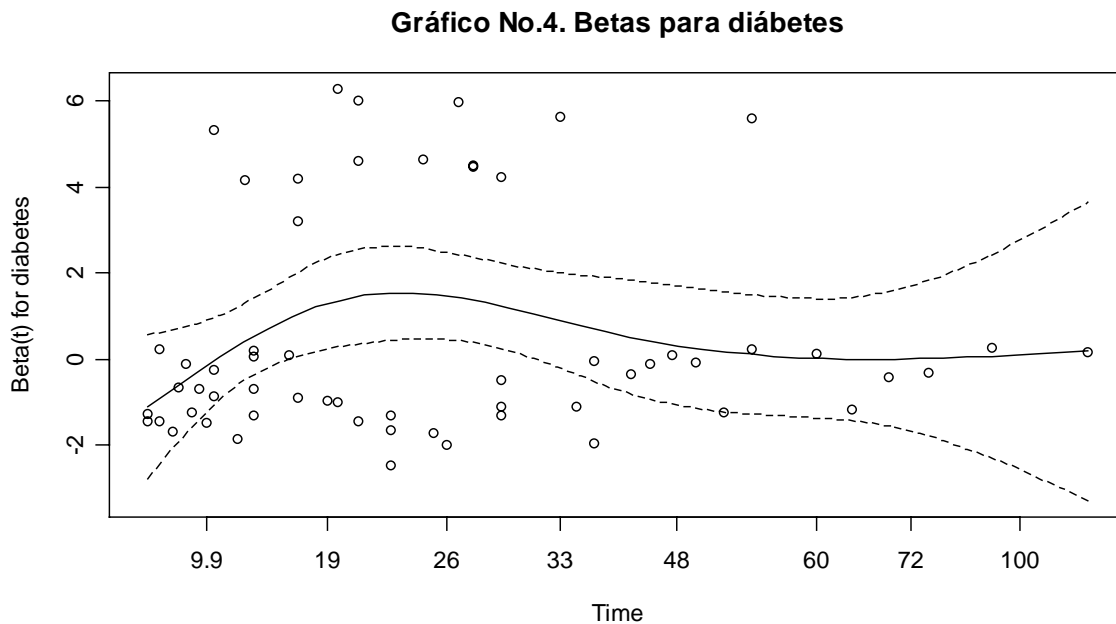
	rho	chisq	p
diabetes	0.0357	0.0808	0.776
edad	0.1165	1.0519	0.305
quetelet	-0.0540	0.2278	0.633
GLOBAL	NA	1.3791	0.710

De donde se concluye de que no existe evidencia significativa al 5% de que se viole el supuesto de riesgos proporcionales, ni desde el punto de vista global, ni para cada covariable.

Para cada una de las covariables también pueden obtenerse los gráficos para los betas. Para la Variable diabetes, el gráfico correspondiente se obtiene mediante el comando:

```
> plot(cox.zph(cox1),var=1,main="Gráfico No.4. Betas para diábetes")
```

Y toma la forma:

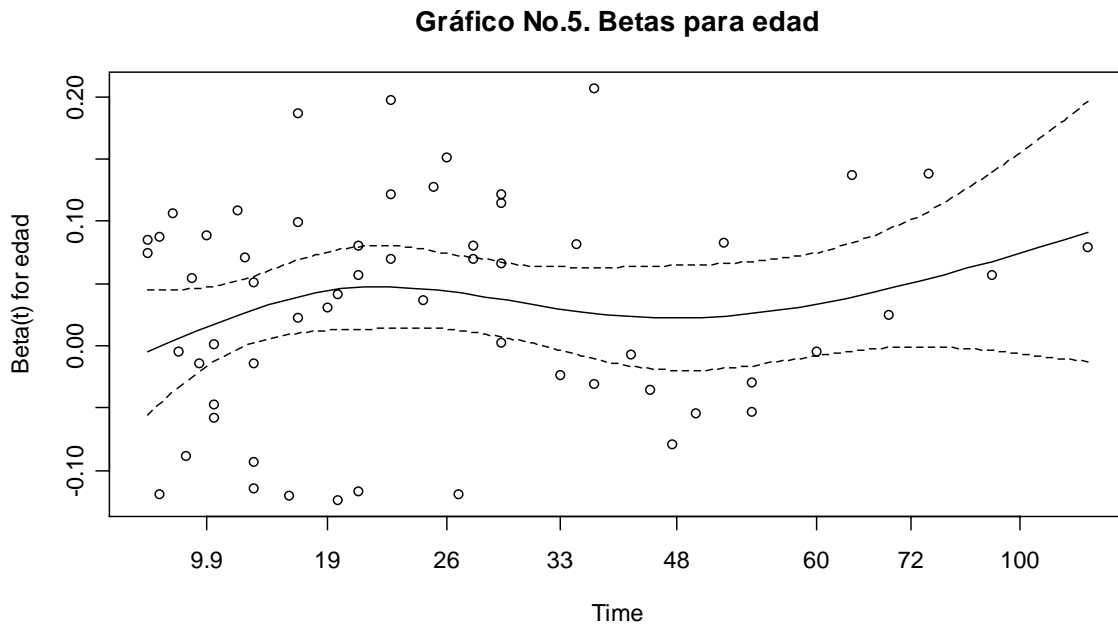


Para la variable edad, el código se escribe como:



```
> plot(cox.zph(cox1),var=2,main="Gráfico No.5. Betas para edad")
```

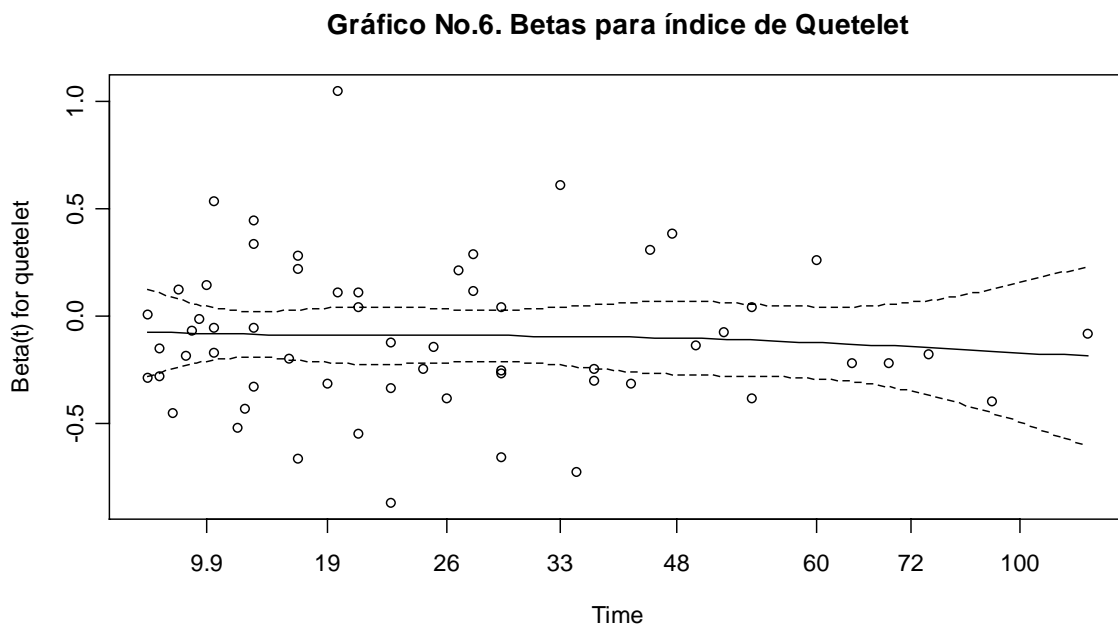
Y el gráfico es:



Y para la variable índice de Quetelet, el comando es.

```
> plot(cox.zph(cox1),var=3,main="Gráfico No.6. Betas para índice de Quetelet")
```

Y el gráfico es:

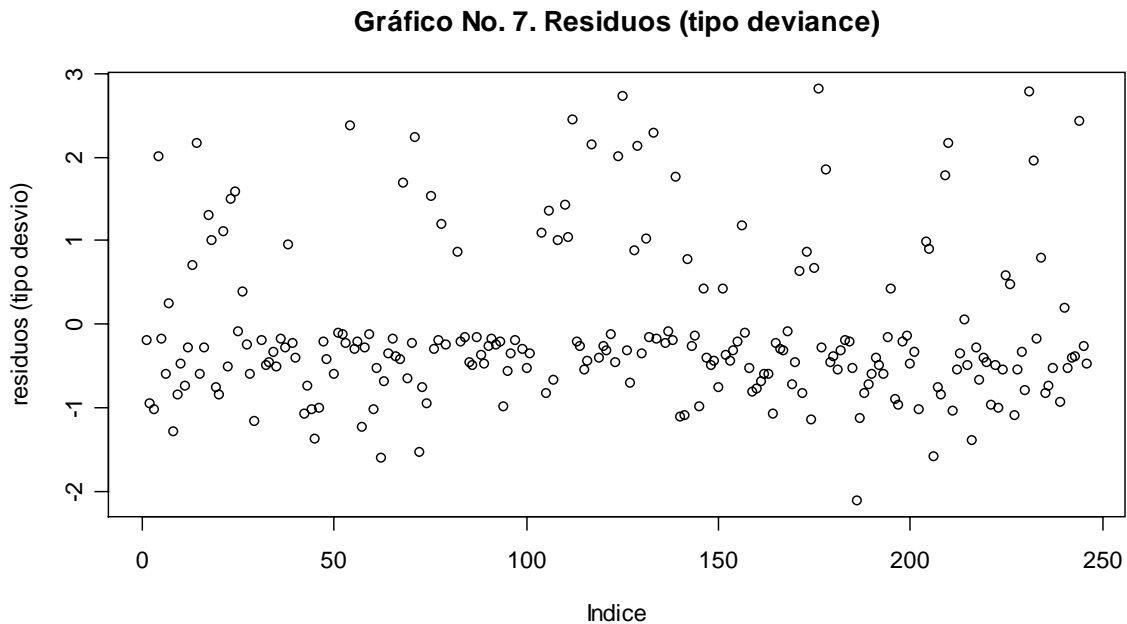


### 2.3.6.2. Residuos tipo deviance.

Los residuos tipo deviance pueden generarse a través del comando:

```
> plot(resid(cox1,type="deviance"),xlab="Indice",ylab="residuos (tipo desvio)", main="Gráfico No. 7. Residuos (tipo deviance)")
```

Los cuales generan el gráfico:



Y en cual se evidencia que no existe ningún individuo que esté influenciando en el ajuste del modelo.

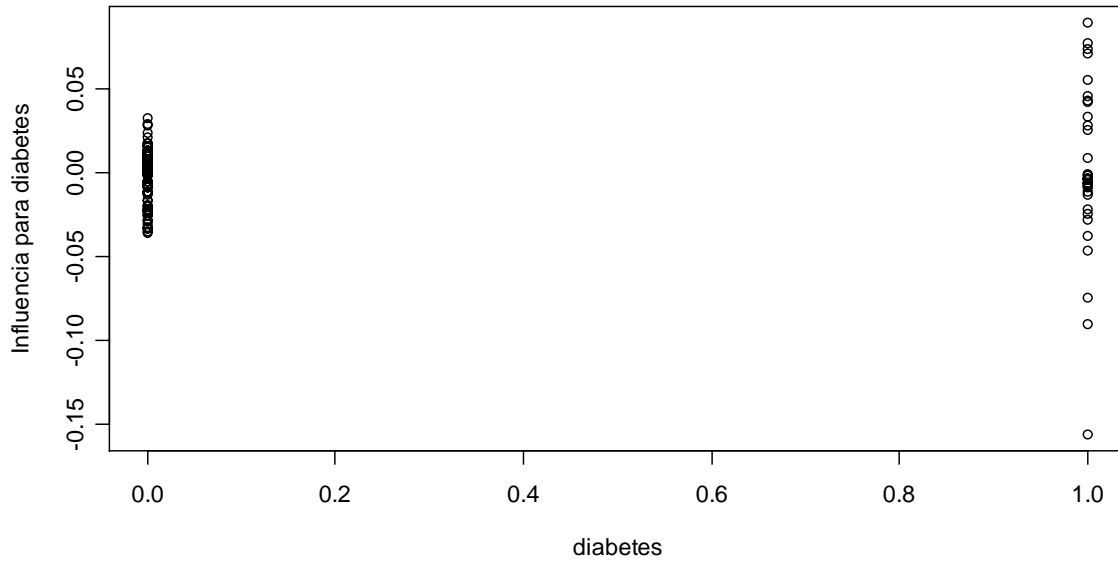
### 2.3.6.3. Gráficos de influencias sobre la estimación de cada coeficiente.

Estos gráficos se obtienen utilizando los residuos dfbetas. Para el caso de diabetes, el gráfico se genera a través de los comandos:

```
> rr<-resid(cox1,type="dfbeta")
> attach(dpa)
> plot(diabetes, rr[,1], xlab="diabetes", ylab="Influencia para diabetes",main="Gráfico No. 8. Gráfico de influencias para diabetes")
```

Obteniéndose el gráfico:

**Gráfico No. 8. Gráfico de influencias para diabetes**

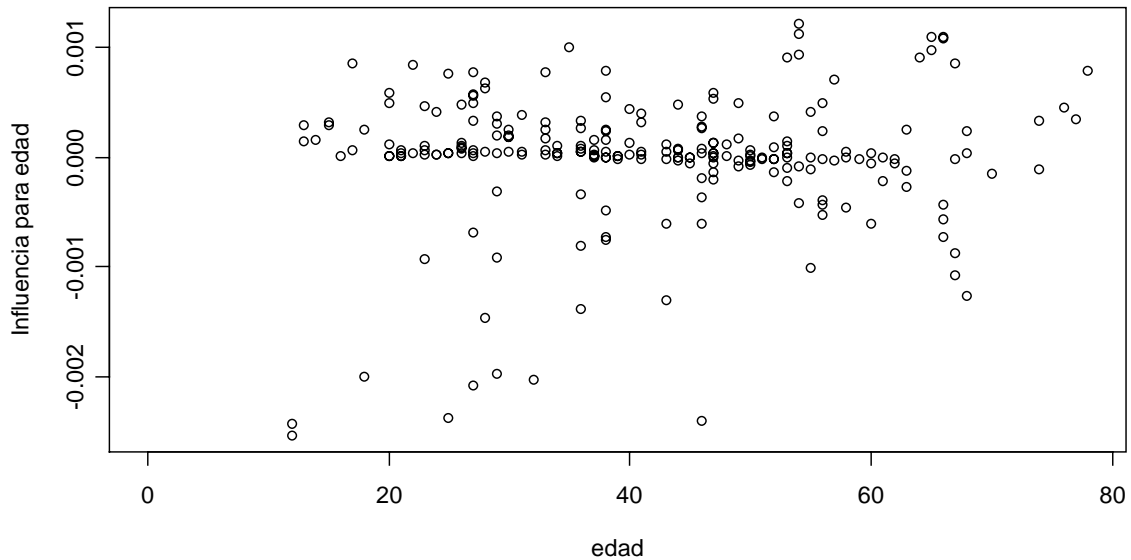


Para la variable edad, el gráfico se obtiene mediante el comando:

```
> plot(edad,rr[,2],xlab="edad",ylab="Influencia para edad",main="Gráfico No. 9. Gráfico de influencias para edad")
```

del cual se obtiene el siguiente gráfico:

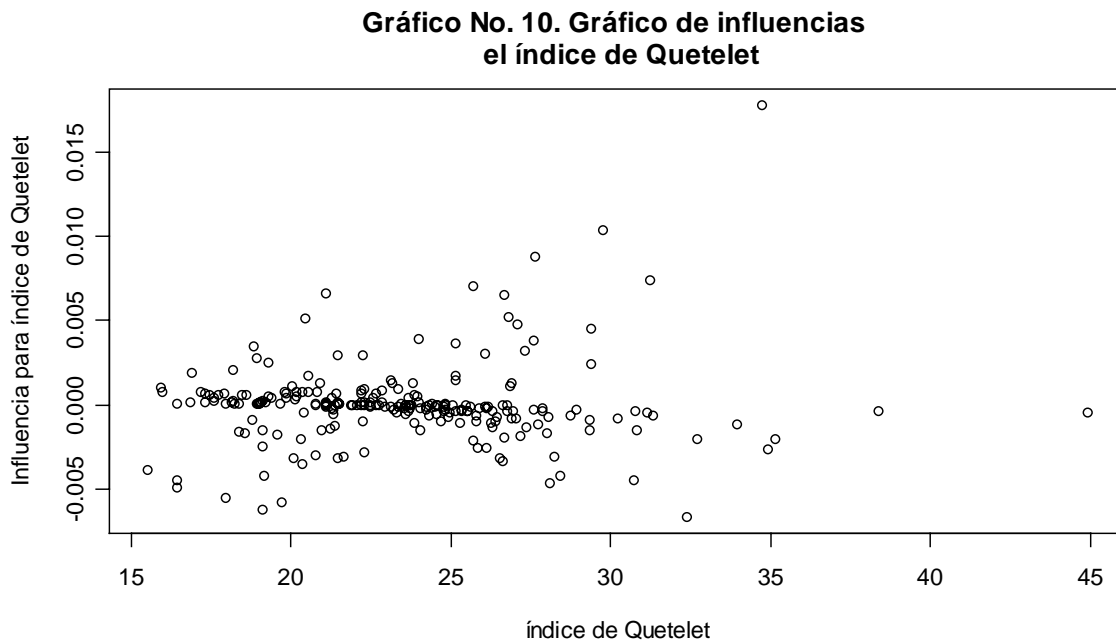
**Gráfico No. 9. Gráfico de influencias para edad**



Finalmente, para el índice de Qetelet, se obtiene mediante el comando:

```
> plot(quetelet,rr[,3],xlab="índice de Qetelet",ylab="Influencia para
índice de Qetelet",main="Gráfico No. 10. Gráfico de influencias \n el
índice de Qetelet")
```

Con el cual se genera el gráfico:



Pudiéndose observar que existe un individuo que esta influenciando sobre la estimación del coeficiente correspondiente al índice de Qetelet.

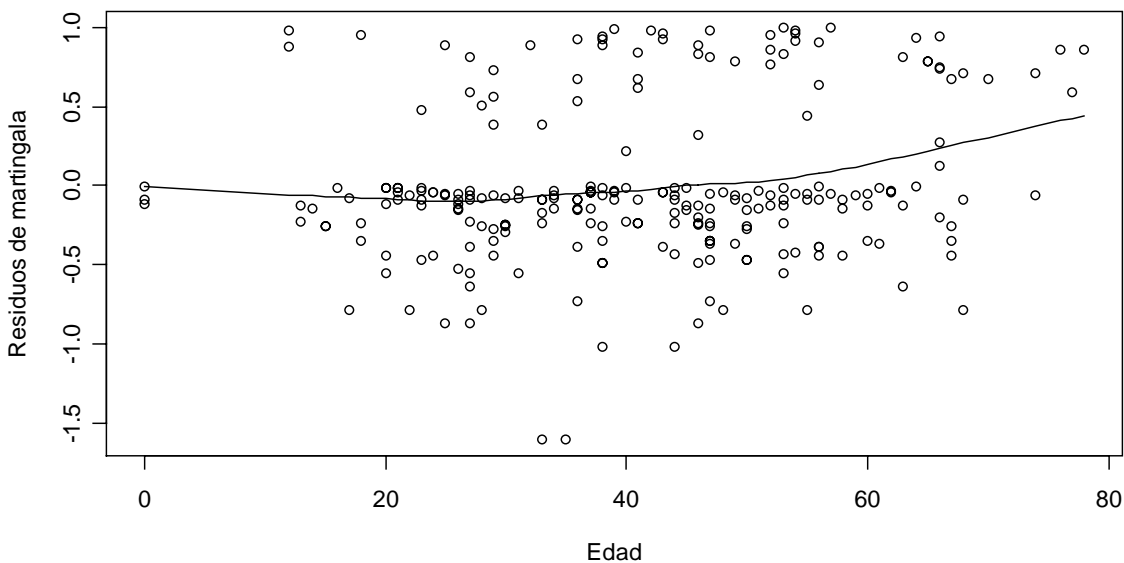
#### 2.3.6.4. Forma funcional de las variables continuas.

La adecuacidad de la forma funcional para la variable edad puede ser verificada a través de los residuos de martingala, mediante el siguiente código:

```
> cox1.0<-coxph(Surv(meses,censor2)~1,data=dpa, na.action=na.exclude)
> rr<-resid(cox1.0)
> plot(dpa$edad,rr,xlab="Edad",ylab="Residuos de martingala",
main="Gráfico No. 11. Verificacion de la forma funcional para edad")
> lines(lowess(dpa$edad,rr,iter=0))
```

Obteniéndose el gráfico:

**Gráfico No. 11. Verificación de la forma funcional para edad**



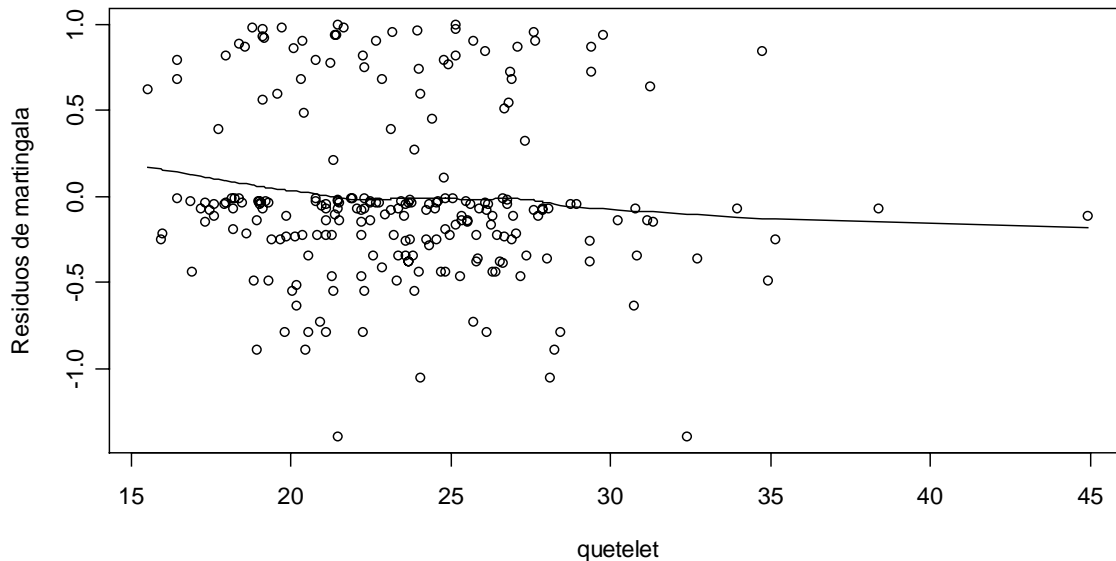
Observándose una forma funcional adecuada para la variable edad.

Para el caso de la variable índice de Quetelet, la adecuación de la forma funcional se puede verificar mediante los siguientes comandos:

```
> dpa.nq<- na.omit(dpa[, c("meses","censor2","diabetes", "edad",
"quetelet")])
> cox.nq.0<-coxph(Surv(meses,censor2)~1,data=dpa.nq)
> rr<-resid(cox.nq.0)
> plot(dpa.nq$quetelet,rr,xlab="quetelet",ylab="Residuos de martingala",
main="Gráfico No. 12. Verificación de la forma funcional \n el indice de
quetelet")
> lines(lowess(dpa.nq$quetelet,rr,iter=0))
```

el cual genera el siguiente gráfico:

**Gráfico No. 12. Verificación de la forma funcional  
el índice de quetelet**



En el que se observa que la forma funcional para el índice de Quetelet parece ser adecuada.

### Referencias.

- Andersen, P.K., Borgan, Ø., Gill, R.D. y Keiding, N. (1993). *Statistical Models Based on Counting Processes*. N.Y.: Springer-Verlag.
- Borges, R.E. (2002). *Análisis de Supervivencia Aplicado a un Caso de Diálisis Renal: Diálisis Peritoneal en el Hospital Clínico Universitario de Caracas y Hemodiálisis en el Hospital de Clínicas Caracas, 1980-2000*. Tesis de M.Sc. en Estadística Aplicada, Mérida, Venezuela: Instituto de Estadística Aplicada y Computación, Universidad de Los Andes. (Disponible en: <http://tesis.saber.ula.ve>)
- Cox, D.R. (1972). Regression models and life tables (with discussion). *Journal of the Royal Statistical Society: Series B*, **34**: 187-220.
- Fleming, T.R. y Harrington, D.P. (1991). *Counting Processes and Survival Analysis*. N.Y.: John Wiley & Sons, Inc.
- Greenwood, M. (1926). The natural duration of cancer. *Reports on Public Health and Medical Subjects*, **33**: 1-26, Londres: Her Majesty's Stationery Office.
- Harrington, D. P. and Fleming, T. R. (1982). A class of rank test procedures for censored survival data. *Biometrika*, **69**:553-566.

- Hosmer, D.W. y Lemeshow, S. (1999). *Applied Survival Analysis: Regression Modeling of Time to Event Data*. N.Y.: John Wiley & Sons, Inc.
- Kaplan, E.L. y Meier, P. (1958). Nonparametric estimation from incomplete observations. *Journal of the American Statistical Association*, **53**: 457-481.
- Klein, J.P. y Moeschberger, M.L. (1997). *Survival Analysis: Techniques for Censored and Truncated Data*. New York: Springer-Verlag.
- Mantel, N. (1966). Evaluation of survival data and two new rank order statistics arising in its consideration. *Cancer Chemotherapy Reports*, **50**: 163-170.
- Peto, R. y Peto, J. (1972). Asymptotically efficient rank invariant test procedures (with discussion). *Journal of the Royal Statistical Society: Series A*, **135**: 195-206.
- R Development Core Team (2004). *R: A language and environment for statistical computing*. Viena, Austria: R Foundation for Statistical Computing.
- S original by Terry Therneau and ported by Thomas Lumley (2004). *survival: Survival analysis, including penalised likelihood*. R package version 2.13-2.
- Schoenfeld, D. (1982). Partial residuals for the proportional hazard regression model. *Biometrika*, **69**: 239-241.
- Therneau, T.M. y Grambsch, P.M. (2000). *Modeling Survival Data: Extending the Cox Model*. N.Y.: Springer-Verlag.
- Therneau, T.M., Grambsch, P.M. y Fleming, T.R. (1990). Martingale-based residuals for survival models. *Biometrika*, **77**: 147-160.