

Covariable dependiente del tiempo Un ejemplo data burn

El modelo de Cox PH puede ser utilizado para modelar la dependencia de la función de riesgo de las variables explicativas. Estas variables explicativas se miden sólo una vez.

Sin embargo, en algunas situaciones, se puede desear variables explicativas cuyos valores cambia con el tiempo. En muchos estudios de supervivencia se recolecta mediciones de las variables explicativas a lo largo del estudio y el mejor modelo para la función riesgo sobre el tiempo puede ser obtenido al permitir que la función de riesgo en un momento determinado dependa de la medición más reciente de la variable explicativa. La evaluación informal de la asunción de riesgos proporcionales se puede hacer gráficamente, por ejemplo mediante el examen de las curvas de supervivencia Kaplan-Meier (si los riesgos son proporcionales entonces las curvas de supervivencia no se debe cruzar) o las curvas de los log de las funciones riesgos acumuladas (como hemos visto, estos deben ser paralelas). Una prueba más formal de la asunción de riesgos proporcionales se puede hacer mediante la introducción de una variable dependiente del tiempo en el modelo de regresión.

Vamos a mostrar otro ejemplo de modelo de riesgos proporcionales de Cox con ajuste covariables dependientes del tiempo. Los datos utilizados aquí es el conjunto de datos dados en el archivo burn.txt. Para simplificar el problema, sólo nos centramos en dos o tres covariables fijas más importantes, además de las dos covariables dependientes del tiempo (tiempo de la escisión o el tiempo de estudio y al tratamiento con antibióticos profilácticos o sobre el tiempo de estudio). Así al reorganizar los datos en un nuevo conjunto de datos [burnnew.txt](#) y las covariables son: Z1, el tratamiento (0-rutina de baño 1-Cuerpo de limpieza), Z2, el sexo (0-hombre, 1 mujer); Z3, la raza (0 no- blancos, 1 blanco), T1, el tiempo de la escisión o en el tiempo de estudio y el indicador asociado, IT1, T2, el tiempo de tratamiento profiláctico con antibióticos o en el tiempo de estudio y el indicador asociado, IT2. Estamos interesados en el tiempo la infección, que se registra en el T3. El estado de la censura del evento infección es IT3.

1. Estocástico covariables dependientes del tiempo

En Splunk, para el ajuste del modelo de Cox con coeficientes dependiente del tiempo, se tiene que construir un Nuevo conjunto de datos, por ejemplo, utilizando el proceso de formulación de contar "(start, stop, status)". Por ejemplo, en los datos originales, tenemos una línea de la siguiente manera:

```
z1 z2 z3 t1 it1 t2 it2 t3 it3
148 1 0 1 15 1 39 1 76 0
```

Considere sólo un ensayo en el modelo, por ejemplo, las covariables T1, el tiempo de la escisión. Los pacientes que recibieron la escisión se tratan como dos de los casos, antes y después de la escisión. Después de reformular este dato en particular se tiene:

```

start stop event z1 z2 z3 it1 id
0 15 0 1 0 1 0 148
15 76 0 1 0 1 1 148

```

Recodificar las dos variables de TD como la siguiente:

$Z_1(t) = 0$ si $t \leq$ tiempo en que se produce la escisión

$= 1$ Si $t >$ tiempo en que se produce la escisión

$Z_2(t) = 0$ si $t \leq$ tiempo en que se produce el tratamiento profiláctico antibiótico

$= 1$ si $t >$ tiempo en que se produce el tratamiento profiláctico antibiótico

que puede ser hecha por la rutina **tdc(time, censor, tdc, data)** ([tdc.ssc](#)). En esta función, el argumento **time** y **censor** se suministran con el tiempo de supervivencia y la condición de censurando por separado, argumento **data** con covariables fijas. La **tdc** toma una matriz de variables dependientes del tiempo, con cada covariable TD ocupando dos columnas (primera columna es el tiempo y la segunda es el indicador de si se produce el evento).

Lo primero se ajustan al modelo con sólo las covariables fijas y luego con variables fijas + dependiente del tiempo. Desde la salida de la computadora, tenemos los siguientes resultados. Ni el tiempo de escisión y el tiempo de tratamiento de anticuerpos tiene efectos significativos sobre el tiempo de la infección.

	Modelo con sólo covariables fijas			Modelo con tratamiento y ambas variables dependiente del tiempo			Modelo con tratamiento y TDC T1			Modelo con tratamiento y TDC T2		
	B	se (B)	exp (B)	B	se (B)	exp (B)	B	se (B)	exp (B)	B	se (B)	exp (B)
TR	-.625 (0.034)	.294	.535	-.494 (0.10)	.300	.610	-.486 (0.10)	0.298	0.615	-.565 (0.057)	0.296	0.568
Género	-.640 (0.100)	.390	.527									
raza	2.153 (0.033)	1.010	8.613									
T1				-.591 (0.19)	.449	.554	-.582 (0.19)	0.448	0.559			
T2				0.080 (0.83)	.367	1.084				0.0309 (0.930)	0.365	1.031

```

fit.cox=coxph(Surv(t3, it3) ~ z1 + z2 + z3, data=burnnew)
summary(fit.cox)

time=burnnew$t3
censor=burnnew$it3
tdcs=burnnew[,4:7]
data=burnnew[,1:3]
burn2=tdc(time, censor, tdcs, data)> tdc.fit1=coxph(Surv(start, stop,
event) ~ z1 + it1 + it2, data=burn2)
summary(tdc.fit1)

tdc1 = tdcs[, 1:2]
burn3=tdc(time, censor, tdc1, data)
names(burn3)[4]<-"it1"
tdc.fit2=coxph(Surv(start, stop, event) ~ z1 + it1, data=burn3)
summary(tdc.fit2)

burn4 = tdc(time, censor, tdcs[3:4], data)
names(burn4)[4]="it2"
tdc.fit3=coxph(Surv(start, stop, event) ~ z1 + it2, data=burn4)
summary(tdc.fit3)

```

Para ilustrar cómo el factor de riesgo, $\exp(Xb + \gamma Z(t))$, cambios sobre el tiempo, asumimos que el tiempo de la escisión (T1) y tiempo de tratamiento con anticuerpos (T2) son las variables importantes y se incluyen en el modelo con todas las otras covariables fijas. Echemos un vistazo a los individuos con el tratamiento (id = 3, 4, 10, 22) y aquellos sin tratamiento (id = 62, 66, 65, 67).

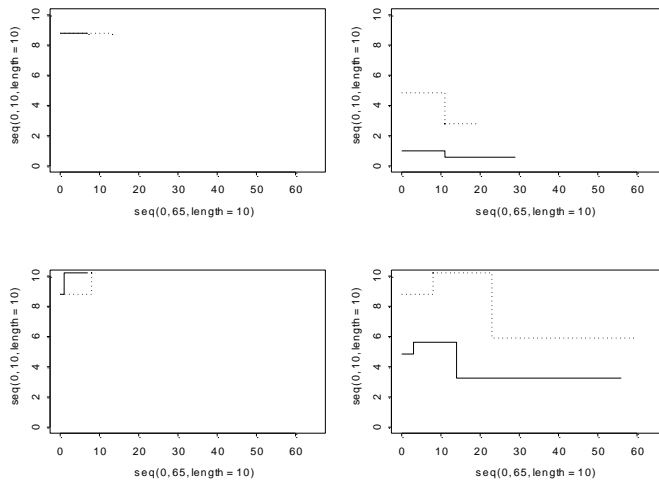
```

tdc.fit4=coxph(Surv(start, stop, event) ~ z1 + z2 + z3 + it1 + it2, data
= burn2)
coef=as.vector(tdc.fit4$coeff)
coef=coef[c(4:5, 1:3)]
X=burn2[,4:8]
datay=exp(as.matrix(X)%*%coef)
dataxy=cbind(burn2$stop, datay, burn2$id)
id=matrix(c(3, 4, 10, 22, 62, 66, 65, 67), byrow=T, nrow=2)

par(mfrow=c(2,2))
for (i in 1:4)
{
  plot(seq(0, 65, length=10), seq(0, 10, length=10), type="n")
  ii=0
  for (j in id[, i])
  {
    ii=ii+1
    x=(dataxy[,1])[dataxy[,3]==j]
    y=(dataxy[,2])[dataxy[,3]==j]
    lines(stepfun(c(0,x), y), lty=ii)
  }
}

```

```
}
```



También podemos considerar el modelo con las interacciones entre las covariables fijas y TDC. El procedimiento de selección paso a paso del modelo resulta en el modelo final con todas las covariables fijas y dos términos de interacción, entre el género y el tiempo de la escisión y entre la raza y el tiempo de la escisión.

```
> tdc.fit<-coxph(Surv(start, stop, event) ~ (z1 + z2 + z3 ) * it1, data =
burn3)
> tdc.fit
Call:
coxph(formula = Surv(start, stop, event) ~ (z1 + z2 + z3) * it1, data =
burn3)
```

	coef	exp(coef)	se(coef)	z	p
z1	-0.547	0.57855	0.354	-1.547	0.120
z2	-1.539	0.21457	0.730	-2.110	0.035
z3	6.441	627.32206	10.028	0.642	0.520
it1	4.448	85.42339	10.090	0.441	0.660
z1:it1	0.117	1.12356	0.658	0.177	0.860
z2:it1	1.867	6.47202	0.915	2.041	0.041
z3:it1	-5.515	0.00403	10.082	-0.547	0.580

Likelihood ratio test=25.6 on 7 df, p=0.000605 n= 236

```
> library(MASS)
```

```
> stepAIC(tdc.fit)
```

Start: AIC= 427.02

```
Surv(start, stop, event) ~ (z1 + z2 + z3) * it1
```

	Df	AIC
- z1:it1	1	425.0524

```

<none> NA 427.0211
- z3:it1  1 427.4337
- z2:it1  1 429.8527

```

Step: AIC= 425.05

```
Surv(start, stop, event) ~ z1 + z2 + z3 + it1 + z2:it1 + z3:it1
```

```

      Df      AIC
<none> NA 425.0524
- z3:it1  1 425.4590
  - z1    1 426.0701
- z2:it1  1 427.8591

```

Call:

```
coxph(formula = Surv(start, stop, event) ~ z1 + z2 + z3 + it1 + z2:it1 +
z3:it1, data =
      burn3)
```

	coef	exp(coef)	se(coef)	z	p
z1	-0.514	0.59814	0.299	-1.721	0.085
z2	-1.539	0.21470	0.730	-2.109	0.035
z3	6.445	629.41523	10.051	0.641	0.520
it1	4.514	91.31598	10.106	0.447	0.660
z2:it1	1.851	6.36501	0.910	2.033	0.042
z3:it1	-5.515	0.00403	10.105	-0.546	0.590

Likelihood ratio test=25.5 on 6 df, p=0.000273 n= 236

2. Modelo determinístico con covariables dependiente del tiempo para chequear la suposición de riesgos proporcionales.

Los datos se reformula por función splitSuv (time, event, data) (splitSurv.ssc), los argumentos de que son el tiempo de supervivencia, el estado de la censura y las covariables fijas en orden. Un efecto lineal varía con el tiempo para la variable fijada X; es decir, pueden ser incluidos por "X: stop" término en el modelo.

```

> reburn=splitSuv(t3, it3, burnnew[, 1:3])
> fit1=coxph(Surv(start, stop, event) ~ z1 + z2 + z3 + stop:z1, data =
reburn)
> summary(fit1)
Call:
coxph(formula = Surv(start, stop, event) ~ z1 + z2 + z3 + stop:z1, data =
reburn)

```

n= 3109

	coef	exp(coef)	se(coef)	z	p
--	------	-----------	----------	---	---

z1	-0.4395	0.644	0.422	-1.042	0.300
z2	-0.6468	0.524	0.390	-1.657	0.098
z3	2.1459	8.550	1.010	2.124	0.034
stop:z1	-0.0161	0.984	0.027	-0.598	0.550

	exp(coef)	exp(-coef)	lower .95	upper .95
z1	0.644	1.552	0.282	1.47
z2	0.524	1.909	0.244	1.13
z3	8.550	0.117	1.180	61.94
stop:z1	0.984	1.016	0.933	1.04

Rsquare= 0.005 (max possible= 0.132)
 Likelihood ratio test= 17.1 on 4 df, p=0.00182
 Wald test = 11.5 on 4 df, p=0.0212
 Score (logrank) test = 13.9 on 4 df, p=0.00762

```
> fit2=coxph(Surv(start, stop, event) ~ z1 + z2 + z3 + stop:z2, data =
reburn)
> summary(fit2)
Call:
coxph(formula = Surv(start, stop, event) ~ z1 + z2 + z3 + stop:z2, data =
reburn)
```

n= 3109

	coef	exp(coef)	se(coef)	z	p
z1	-0.6223	0.537	0.294	-2.113	0.035
z2	-0.9585	0.383	0.590	-1.625	0.100
z3	2.1552	8.630	1.010	2.133	0.033
stop:z2	0.0218	1.022	0.028	0.777	0.440

	exp(coef)	exp(-coef)	lower .95	upper .95
z1	0.537	1.863	0.301	0.956
z2	0.383	2.608	0.121	1.219
z3	8.630	0.116	1.192	62.499
stop:z2	1.022	0.978	0.967	1.080

Rsquare= 0.006 (max possible= 0.132)
 Likelihood ratio test= 17.3 on 4 df, p=0.00166
 Wald test = 11.6 on 4 df, p=0.0203
 Score (logrank) test = 13.9 on 4 df, p=0.00764

```
> fit3=coxph(Surv(start, stop, event) ~ z1 + z2 + z3 + stop:z3, data =
reburn)
> summary(fit3)
Call:
coxph(formula = Surv(start, stop, event) ~ z1 + z2 + z3 + stop:z3, data =
reburn)
```

n= 3109

	coef	exp(coef)	se(coef)	z	p
z1	-0.649	0.522	0.2955	-2.20	0.028
z2	-0.645	0.525	0.3905	-1.65	0.098
z3	5.897	363.980	3.9401	1.50	0.130
stop:z3	-0.139	0.870	0.0925	-1.50	0.130

	exp(coef)	exp(-coef)	lower .95	upper .95
z1	0.522	1.91436	0.293	9.32e-001
z2	0.525	1.90648	0.244	1.13e+000
z3	363.980	0.00275	0.161	8.22e+005
stop:z3	0.870	1.14910	0.726	1.04e+000

Rsquare= 0.007 (max possible= 0.132)
Likelihood ratio test= 20.8 on 4 df, p=0.000355
Wald test = 9.18 on 4 df, p=0.0568
Score (logrank) test = 14.5 on 4 df, p=0.00592

Try a different time function $g(t) = \log(t) - \text{mean}(t)$

```
> z4=log(reburn$stop)-mean(reburn$stop)
> fit11=coxph(Surv(start, stop, event) ~ z1 + z2 + z3 + I(z4*z1), data =
reburn)
> summary(fit11)
Call:
coxph(formula = Surv(start, stop, event) ~ z1 + z2 + z3 + I(z4 * z1),
data = reburn)
```

n= 3109

	coef	exp(coef)	se(coef)	z	p
z1	-3.115	0.0444	4.479	-0.695	0.490
z2	-0.649	0.5227	0.390	-1.662	0.097
z3	2.152	8.6048	1.010	2.130	0.033
I(z4 * z1)	-0.188	0.8286	0.337	-0.558	0.580

	exp(coef)	exp(-coef)	lower .95	upper .95
z1	0.0444	22.533	6.83e-006	288.17
z2	0.5227	1.913	2.43e-001	1.12
z3	8.6048	0.116	1.19e+000	62.32
I(z4 * z1)	0.8286	1.207	4.28e-001	1.60

Rsquare= 0.005 (max possible= 0.132)
Likelihood ratio test= 17.1 on 4 df, p=0.00187
Wald test = 11.6 on 4 df, p=0.021
Score (logrank) test = 13.9 on 4 df, p=0.00776

A continuación, se hará el análisis del punto de cambio. Seleccione el punto de máxima verosimilitud asociada a ella. De género pueden tener un efecto variable en el tiempo, que

se puede hacer por la función `changePt` (`data`, `indic`) ([changePt.ssc](#)), donde el argumento `data` toma los datos reformulado por la función `splitSurv` y `indic` muestra la covariable fijada de interés.

```
> changePt(reburn, 4)
      event.time  logliks
[1,]          1 -217.2525
[2,]          2 -216.7316
[3,]          3 -217.6008
[4,]          4 -217.7581
[5,]          5 -216.9209
[6,]          6 -217.4549
[7,]          7 -217.7749
[8,]          8 -217.7686
[9,]          9 -217.6935
[10,]         10 -217.7094
[11,]         11 -217.5719
[12,]         13 -217.3211
[13,]         14 -217.5854
[14,]         16 -217.7456
[15,]         17 -217.6147
[16,]         18 -216.3561
[17,]         19 -216.7689
[18,]         21 -217.1448
[19,]         23 -217.4374
[20,]         32 -217.6488
[21,]         42 -216.3298
[22,]         44 -216.7694
[23,]         47 -217.2353
[24,]         51 -217.7764
[1] "best="
[1] 21
Call:
coxph(formula = Surv(data$start, data$stop, data$event) ~ x1 + x2,
      method = "breslow")

n= 3109

      coef exp(coef) se(coef)      z      p
x1 -0.459   0.63210      0.3 -1.53 0.13
x2 -5.785   0.00307     12.9 -0.45 0.65

      exp(coef) exp(-coef) lower .95 upper .95
x1   0.63210      1.58 3.51e-001 1.14e+000
x2   0.00307     325.53 3.42e-014 2.76e+008

Rsquare= 0.002   (max possible= 0.132 )
Likelihood ratio test= 6.6  on 2 df,   p=0.0368
```


Wald test = 2.57 on 2 df, p=0.277
Score (logrank) test = 5.48 on 2 df, p=0.0646

> changePt(reburn, 5)

	event.time	logliks
[1,]	1	-218.2118
[2,]	2	-218.1533
[3,]	3	-218.3450
[4,]	4	-218.0548
[5,]	5	-217.2861
[6,]	6	-216.9251
[7,]	7	-217.6808
[8,]	8	-217.3681
[9,]	9	-216.9712
[10,]	10	-216.4737
[11,]	11	-215.5089
[12,]	13	-215.0880
[13,]	14	-214.6195
[14,]	16	-214.0815
[15,]	17	-215.1789
[16,]	18	-216.0502
[17,]	19	-217.2415
[18,]	21	-218.0480
[19,]	23	-218.3470
[20,]	32	-217.2850
[21,]	42	-217.4875
[22,]	44	-217.7596
[23,]	47	-218.1056
[24,]	51	-218.3503

[1] "best="

[1] 14

Call:

coxph(formula = Surv(data\$start, data\$stop, data\$event) ~ x1 + x2,
method = "breslow")

n= 3109

	coef	exp(coef)	se(coef)	z	p
x1	-1.71	0.18	0.728	-2.35	0.0190
x2	2.49	12.05	0.930	2.68	0.0075

	exp(coef)	exp(-coef)	lower .95	upper .95
x1	0.18	5.550	0.0433	0.75
x2	12.05	0.083	1.9463	74.61

Rsquare= 0.004 (max possible= 0.132)

Likelihood ratio test= 11.1 on 2 df, p=0.00389

Wald test = 7.34 on 2 df, p=0.0255

Score (logrank) test = 8.92 on 2 df, p=0.0116

```

> changePt(reburn, 6)
      event.time    logliks
[1,]          1 -214.8830
[2,]          2 -214.8368
[3,]          3 -214.7584
[4,]          4 -214.6545
[5,]          5 -214.4963
[6,]          6 -214.4364
[7,]          7 -214.3373
[8,]          8 -214.2610
[9,]          9 -214.1741
[10,]         10 -214.0746
[11,]         11 -213.9058
[12,]         13 -213.8442
[13,]         14 -213.7787
[14,]         16 -213.7009
[15,]         17 -213.5070
[16,]         18 -213.2553
[17,]         19 -213.0905
[18,]         21 -212.9121
[19,]         23 -212.6449
[20,]         32 -212.3236
[21,]         42 -214.8464
[22,]         44 -214.8632
[23,]         47 -214.8847
[24,]         51 -214.8977
[1] "best="
[1] 20
Call:
coxph(formula = Surv(data$start, data$stop, data$event) ~ x1 + x2,
      method = "breslow")

n= 3109

      coef exp(coef) se(coef)      z    p
x1  7.27 1.44e+003      13.5  0.537 0.59
x2 -7.87 3.82e-004      13.6 -0.579 0.56

      exp(coef) exp(-coef) lower .95 upper .95
x1 1.44e+003  6.97e-004 4.30e-009 4.79e+014
x2 3.82e-004  2.62e+003 1.04e-015 1.40e+008

Rsquare= 0.005    (max possible= 0.132 )
Likelihood ratio test= 14.6  on 2 df,   p=0.00067
Wald test            = 0.56  on 2 df,   p=0.757
Score (logrank) test = 8.1   on 2 df,   p=0.0174

```

```

z1<-burn[,2]
z2<-burn[,3]
z3<-burn[,4]
t1<-burn[,13]
it1<-burn[,14]
t2<-burn[,15]
it2<-burn[,16]
t3<-burn[,17]
it3<-burn[,18]
burnnew<-data.frame(z1,z2,z3,t1,it1,t2,it2,t3,it3)
fit.cox=coxph(Surv(t3, it3) ~ z1 + z2 + z3, data=burnnew)

summary(fit.cox)

Call:
coxph(formula = Surv(t3, it3) ~ z1 + z2 + z3, data = burnnew)

n= 154, number of events= 48

      coef exp(coef) se(coef)      z Pr(>|z|)
z1 -0.6254    0.5350   0.2944 -2.125  0.0336 *
z2 -0.6399    0.5273   0.3899 -1.641  0.1008
z3  2.1555    8.6325   1.0112  2.132  0.0330 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

      exp(coef) exp(-coef) lower .95 upper .95
z1    0.5350      1.8690    0.3005    0.9527
z2    0.5273      1.8964    0.2456    1.1323
z3    8.6325      0.1158    1.1895   62.6457

Concordance= 0.674 (se = 0.044 )
Rsquare= 0.103 (max possible= 0.942 )
Likelihood ratio test= 16.76 on 3 df,  p=0.0007925
Wald test               = 11.3 on 3 df,  p=0.01019
Score (logrank) test = 13.55 on 3 df,  p=0.003585

tdc<-function(time, censor, tdc, data)
{
  n <- length(time)
  id<-1:n;
  ntdc <- ncol(tdc)/2;

  tdc1 <- matrix(0, 1, ntdc);

  data1 <- matrix(0, 1, ncol(data));
  dimnames(data1)[[2]] <- names(data);

  ii <- 0;
  start <- stop <- event <-numeric(1);
  tdc.t <- tdc[,seq(1, ncol(tdc), 2)];
  tdc.e <-tdc[,seq(2, ncol(tdc), 2)];

```

```

if(is.vector(tdc.t)==T)
{
    tdc.t=as.matrix(tdc.t)
    tdc.e=as.matrix(tdc.e)
}
sort.time <- t(apply(cbind(t0=rep(0,n),tdc.t,t4=time), 1, sort));
temp <- rep(0, ntdcs);

for (i in 1:n)
{
    tt <- unique(sort.time[i,])
    for(j in 1:length(tt))
    {
        if(time[i] > tt[j])
        {
            ii <- ii+1;
            start[ii] <- tt[j]
            stop[ii] <- min(time[i], tt[j+1])
            id[ii] <- i
            if(time[i] > tt[j+1]) event[ii] <- 0
            else event[ii] <- censor[i]

            data1 <- rbind(data1, data[i,])
            temp <-tdc.e[i,];
            if(time[i]>stop[ii])

temp[tdc.t[i,]>=stop[ii]]=0

                else temp[tdc.t[i,]>stop[ii]]=0
                tdcsl <- rbind(tdcsl, temp)
            }
        }
    }
    tdcsl <- tdcsl[-1, ]
    data1 <- data1[-1, ]
    X=cbind(start=start, stop=stop, event=event, tdcsl, data1, id=id)
    row.names(X)=1:dim(X)[1]
    X
}

splitSuv<-function(time, event, data) #data contains the other covariates
{
    #sort in ascending order
    sort.ord = order(time);
    n=length(time);
    time1=sort(time);
    id=(1:n)[sort.ord];
    event1 = event[sort.ord];
    data1=data[sort.ord,];

    #variables required for time-dependent covariate analysis
    stop.int = unique(time1);
    start.int = c(0, stop.int[-length(stop.int)]);

    nrep = rep(0, n);
    for (i in 1:n)
    {
        nrep[i] = sum(stop.int<=time1[i]);
    }
}

```

```

start.int1 = start.int[1:nrep[1]];
stop.int1 = stop.int[1:nrep[1]];
for (i in 2:n)
{
    start.int1 = c(start.int1, start.int[1:nrep[i]]);
    stop.int1 = c(stop.int1, stop.int[1:nrep[i]]);
}

id1 = rep(id, nrep);
data1 = data1[rep(1:n, nrep),];
event2 = rep(0, length(stop.int1));
event2[cumsum(nrep)] = event1;

X = cbind(start = start.int1, stop = stop.int1, event = event2, data1,
id = id1);
row.names(X) = 1:nrow(X);
X
}

changePt <- function(data, indic) #the output data from splitSurv
{
    stop = data$stop;
    event.time = unique(stop[data$event == 1]);
    x1 = data[, indic]
    n = length(event.time);
    logliks = rep(0, n);
    for (i in 1:n)
    {
        x2 = x1;
        x2[stop <= event.time[i]] = 0;
        ph.cp = coxph(Surv(data$start, data$stop, data$event) ~ x1 +
x2, method = "breslow");
        logliks[i] = ph.cp$loglik[2];
    }

    print(cbind(event.time, logliks))

    best = (1:n)[logliks == max(logliks)];
    print("best="); print(best);
    x2 = x1;
    x2[stop <= (event.time[best])] = 0;
    ph.cp = coxph(Surv(data$start, data$stop, data$event) ~ x1 + x2,
method = "breslow");
    summary(ph.cp)
}

```