Sesión 9: Uso de STAN y JAGS en Análisis de Sobrevivencia

Aplicaciones en Computación Estadística

Natalie Julian - www.nataliejulian.com

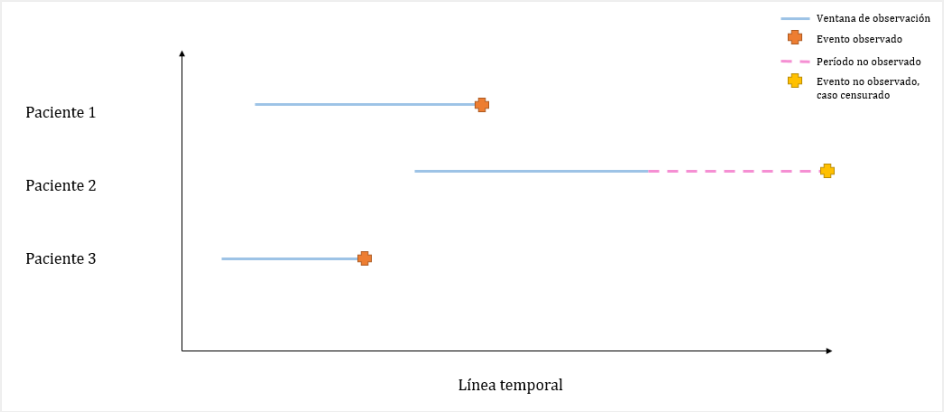
Estadística UC y Data Scientist en Zippedi Inc.

Modelo de Riesgos Proporcionales caso Exponencial en STAN en R

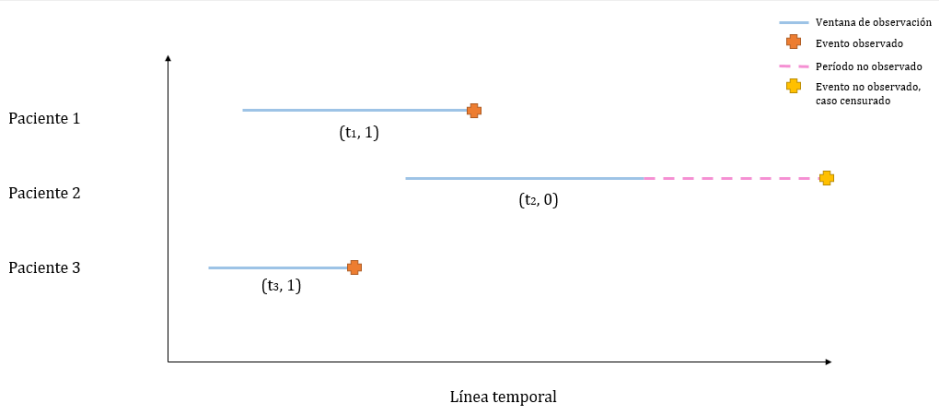
Sobrevivencia de pacientes con cáncer de mama posterior a extirpación de mama

La base de datos `mastectomy` contiene información de sobrevivencia de pacientes con cáncer de mama posterior a mastectomía. Se tienen los tiempos de sobrevivencia en meses. La variable *event* indica si el evento (muerte) ocurrió en el período de observación o no, en caso de que no, se considera una observación censurada por la derecha, es decir, aún no ocurre el fallo pero eventualmente ocurrirá. Se posee una cota inferior para el tiempo de sobrevivencia para los casos censurados. La variable *metastized* indica si existe metástasis (extensión del cáncer en el cuerpo).

Concepto de censura por la derecha



Concepto de censura por la derecha



Efectos en la verosimilitud de casos censurados

- Los fallos o eventos observados corresponden a registros completamente observados, es decir, tenemos toda su información, estos casos son sencillos de añadir al modelo dependiendo de la distribución que se asuman para los tiempos t, se utiliza $f(t)$.
- Los fallos o eventos no observados, corresponden a registros parcialmente observados (casos censurados $\delta = 0$) se tiene una cota inferior (para el caso de la censura por la derecha) para estos casos se añade a la verosimilitud la función de distribución complementaria acumulada $1 - F(t)$.

En un caso de análisis de sobrevivencia, la función de sobrevivencia se define como $S(t) = 1 - F(t)$. La verosimilitud para cada observación queda escrita de la siguiente forma: $f(t_i|x_i, \beta)^{\delta_i} S(t_i|x_i, \beta)^{1-\delta_i}$

Sobrevivencia de pacientes según metástasis

La variable *metastized* indica si existe metástasis (extensión del cáncer en el cuerpo).
Interesa evaluar la sobrevivencia de pacientes dependiendo de si existe metástasis o no.

Datos

```
library("HSAUR")

data("mastectomy", package = "HSAUR")

print(mastectomy)
   time event metastized
1    23  TRUE         no
2    47  TRUE         no
3    69  TRUE         no
4    70 FALSE         no
5   100 FALSE         no
6   101 FALSE         no
7   148  TRUE         no
8   181  TRUE         no
9   198 FALSE         no
```

Modelo de Riesgos Proporcionales

Considere el siguiente modelo de riesgos:

$$h(t; x, \beta) = h_0(t) exp(x\beta)$$

Si se calcula el cuociente enter dos hazard $h_i(t; x, \beta)$ siempre se cancela el término $h_0(t)$, por ende es constante respecto al tiempo.

Un modelo con estas características se denomina de Riesgos proporcionales pues para dos individuos el hazard ratio es constante respecto al tiempo.

La función $h_0(t)$ se denomina hazard base o basal, corresponde a la función de riesgo cuando todas las covariables están centradas en el promedio.

Riesgo Relativo en Modelo de Riesgos Proporcionales

Es fácil ver que el riesgo relativo al incrementar en una unidad la variable x es:

$$HR(t; x+1, x) = \frac{h(t; x+1, \beta)}{h(t; x, \beta)} = \frac{h_0(t) exp((x+1)\beta)}{h_0(t) exp(x\beta)} = exp(\beta)$$

En el caso nuestro, nuestra posible variable explicativa es categórica (en este caso sería presentar metástasis) se interpreta:

Si $\beta > 0$ el riesgo de morir es mayor para el grupo Metástasis sí
Si $\beta < 0$ el riesgo de morir es mayor para el grupo Metástasis no
Si $\beta = 0$ el riesgo de morir es el mismo para ambos grupos

Notar que $h_(t; x, \beta)$ al tener una componente que dependa del tiempo y otra de los parámetros implica un modelo semiparámetrico.

Modelo de Riesgos Proporcionales caso Exponencial

Suponga que $h_0(t) = h_0 = exp(\gamma)$ (es decir, la función hazard basal es constante respecto al tiempo) entonces:

$$h(t; x, \beta) = h_0(t)exp(x\beta) = exp(\gamma)exp(x\beta) = exp(\gamma + x\beta)$$

Este modelo pertenece a la familia de modelos de Riesgos proporcionales y también de tiempos acelerados (esto lo veremos más tarde). Este modelo es una regresión de riesgos proporcionales exponencial.

Es interesante que en este caso se modela con una distribución exponencial de parámetros $exp(\gamma + x\beta)$ donde γ se entiende como el intercepto y $x\beta$ es la combinación lineal de los regresores. Ver fuente: `data.princeton.edu/wws509/notes/c7.pdf`

.

Elección de prioris

Tenemos dos parámetros en nuestro modelo:

- β
- El intercepto (γ)

Elección de priori para β

- β

β se asocia al riesgo relativo de presentar metástasis, donde el riesgo relativo es $exp(\beta)$. Se esperaría que $exp(\beta) > 1$ (pero esto está por verse). Elegiremos una distribución a priori para β centrada en 0 y con una varianza no tan grande. Utilizaremos como priori para β una distribución Normal(0,2), permitiéndole flexibilizar entre valores mayores, menores o iguales a cero.

Elección de priori para el intercepto

- El intercepto (γ)

Para el intercepto, notar que e^{γ} se asocia a la hazard basal, y en modelos exponenciales el riesgo es (exactamente) inversamente proporcional al tiempo de sobrevida medio, por ende:

```
mean(mastectomy$time)
[1] 96.61364
```

Luego, $\frac{1}{exp(\gamma)} \approx 96.6$

```
log(1/96.61364) #Estimación para el intercepto (gamma)
[1] -4.57072
```

Por lo anterior, se utilizará una priori centrada en torno a -5 aproximadamente, una priori Normal(-5,2).

Ver fuente: https://stackoverflow.com/questions/35542953/opposite-directions-of-exponential-hazard-model-coefficients-with-survreg-and

Modelo en STAN

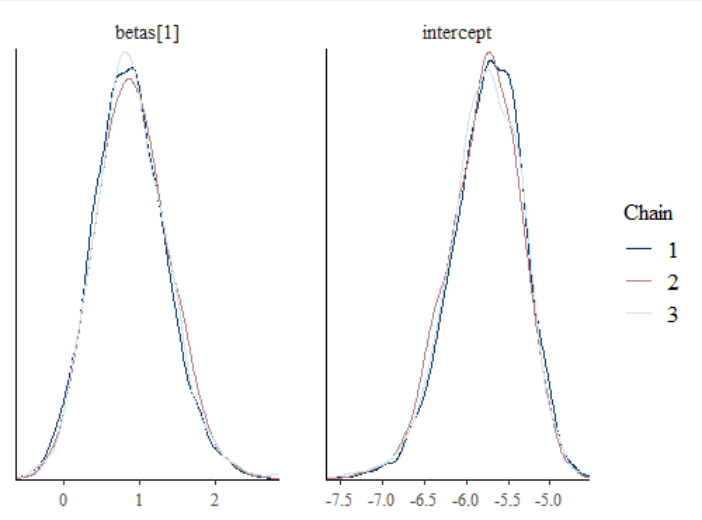
```
parameters {
    vector[NC] betas;
    real intercept;
}

model {
    betas ~ normal(0,2);
    intercept ~ normal(-5,2);

    target += exponential_lpdf(times_uncensored | exp(intercept+X_uncensored*betas));
    //Función de densidad para casos no censurados f(t)

    target += exponential_lccdf(times_censored | exp(intercept+X_censored*betas));
    //Función de densidad acumulada complementaria 1-F(t)
}

generated quantities {
     vector[N] log_lik;

     for(i in 1:N_uncensored){
           log_lik[i] = exponential_lpdf(times_uncensored[i] | exp(intercept+X_uncensored[i]*betas));
      }
      for(i in 1:N_censored){
           log_lik[i+N_uncensored] = exponential_lccdf(times_censored[i] | exp(intercept+X_censored[i]*betas));
      }
}
```

Se extrae la log-verosimilitud para cada caso con `generated quantities`.

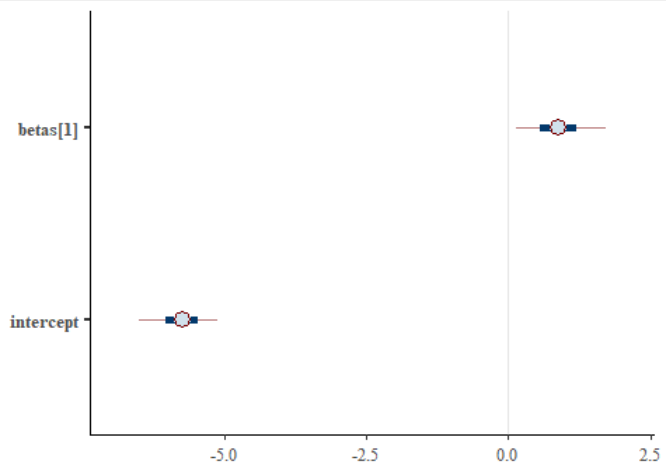
Resultados

```
fit1 <- stan(model_code=scode,
             data=stan_data,
             warmup=150, #quema
             iter=4000, #largo de la cadena
             chains=3) #cadenas

summary(fit1)

                    mean      se_mean         sd          2.5%          25%          50%
betas[1]       0.8915184 0.0094492855 0.47817036    0.01299189    0.5649573    0.8711698
intercept     -5.7728833 0.0084517034 0.42827788   -6.68706213   -6.0424940   -5.7438620
```

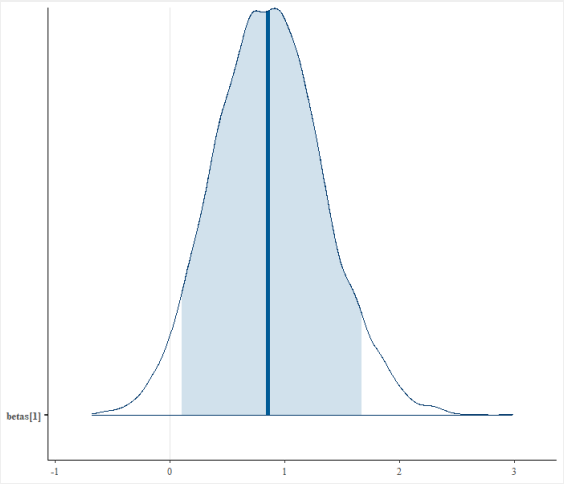
Resultados



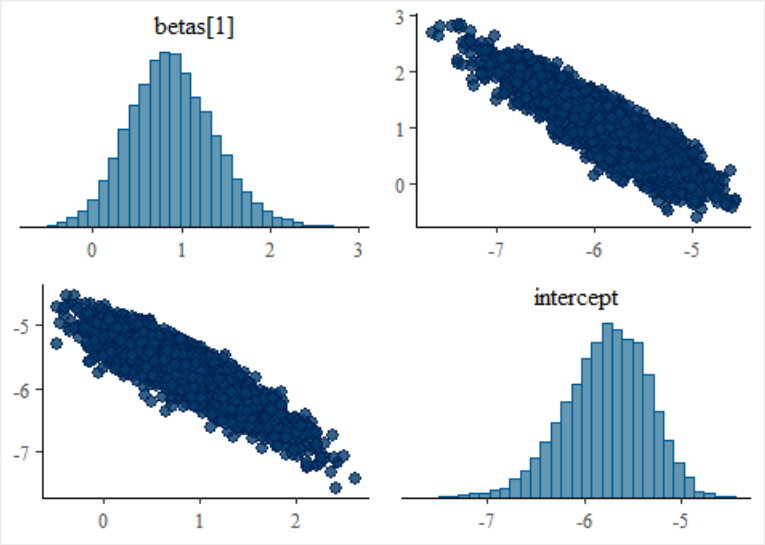
Resultados



Resultados



Resultados



Librería survival

```
library(survival)

modelphe<-survreg(Surv(time, event)~metastized, data=mastectomy, dist="exponential")

modelphe$coefficients #Dan en direcciones opuestas
(Intercept)  metastized
  5.7563749  -0.9110636
```

Con survreg modelamos los tiempos de sobrevivencia medios. Con el modelo en STAN modelamos el riesgo de fallo (muerte). Recordar que el riesgo es (exactamente) inversamente proporcional a los tiempos de sobrevivencia medios en modelos exponenciales.
Sabemos que $h(t; x, \beta) = exp(\gamma + x\beta)$, luego, los tiempos de sobrevivencia medios se modelan a través de:

$$\frac{1}{h(t; x, \beta)} = \frac{1}{exp(\gamma + x\beta)} = \frac{exp(0)}{exp(\gamma + x\beta)} = exp(-\gamma - x\beta)$$

Comparando modelos

Planteemos un modelo donde la priori del intercepto no se especifique (STAN asume una distribución normal centrada en cero y de varianza muy grande) y para β una priori Normal(0,10), comparemos estos modelos.

Modelo 2

```
parameters {
    vector[NC] betas;
    real intercept;
}

model {
    betas ~ normal(0,10);
    target += exponential_lpdf(times_uncensored | exp(intercept+X_uncensored*betas));
    //Función de densidad para casos no censurados f(t)

    target += exponential_lccdf(times_censored | exp(intercept+X_censored*betas));
    //Función de densidad acumulada complementaria 1-F(t)
}

generated quantities {
     vector[N] log_lik;

     for(i in 1:N_uncensored){
           log_lik[i] = exponential_lpdf(times_uncensored[i] | exp(intercept+X_uncensored[i]*betas));
      }
      for(i in 1:N_censored){
           log_lik[i+N_uncensored] = exponential_lccdf(times_censored[i] | exp(intercept+X_censored[i]*betas));
      }
}
```

Criterios de comparación de modelos en STAN

```
library(loo)
loglike1 <- extract_log_lik(fit1) #Extraemos la log-verosimilitud
loglike2 <- extract_log_lik(fit2)

(waic1<-waic(loglike1)) #Menor waic es mejor

Computed from 11550 by 44 log-likelihood matrix

          Estimate   SE
elpd_waic   -158.6 15.1
p_waic         2.2  0.3
waic         317.3 30.1

(waic2<-waic(loglike2))

Computed from 11550 by 44 log-likelihood matrix

          Estimate   SE
elpd_waic   -158.8 15.1
p_waic         2.3  0.3
waic         317.6 30.3
```

Criterios de comparación de modelos en STAN

```
(looc1<-loo(loglike1)) #menor looic mejor
         Estimate   SE
elpd_loo   -158.7 15.1
p_loo         2.3  0.3
looic       317.4 30.1
------
Monte Carlo SE of elpd_loo is 0.0.

All Pareto k estimates are good (k < 0.5).
See help('pareto-k-diagnostic') for details

(looc2<-loo(loglike2))
         Estimate   SE
elpd_loo   -158.8 15.1
p_loo         2.4  0.3
looic       317.6 30.3
------
Monte Carlo SE of elpd_loo is 0.0.

All Pareto k estimates are good (k < 0.5).

       elpd_diff se_diff
model1  0.0       0.0
model2 -0.1       0.2

# elpd: Expected log predictive density (ELPD)
#Modelo 1 es mejor, pero las diferencias no son tan considerables
```

Modelo AFT de Valores Extremos con JAGS en R

Modelo AFT (Tiempo de fallo acelerado)

Los modelos de tiempo de fallo acelerado (AFT) consisten en modelar los tiempos de sobrevivencia T en escala logarítmica a través de una combinación lineal de las covariables x. Es decir, sea $Y = log(T)$ entonces un modelo AFT se escribe de la siguiente forma:

$$Y_i = x_i^T \beta + \epsilon_i$$

Donde ϵ corresponde a los errores.

¿Por qué utilizar un modelo AFT trae consigo una ventaja al momento de plantear modelos de regresión para los tiempos de sobrevivencia?

Modelo AFT

¿Por qué utilizar un modelo AFT trae consigo una ventaja al momento de plantear modelos de regresión para los tiempos de sobrevivencia?

Al querer modelar tiempos, esta variable recorre valores positivos, lo que puede generar conflicto al utilizar un modelo de regresión (pues se puede escapar de este intervalo o soporte). Al aplicarle logaritmo a esta variable de interés, ampliamos el soporte y nos liberamos de este conflicto.

Modelo AFT de Valores Extremos

Si asumimos una distribución de valores extremos para el $log(T)$ esto es equivalente a que T tenga una distribución Weibull.

Ver fuente: https://www4.stat.ncsu.edu/~dzhang2/st745/chap5.pdf

Modelo Weibull

$$S(t) = exp(-\lambda t^{p)}$$

Despejando tenemos que:

$$t = (-log(S(t)))^{\frac{1}{p}} \frac{1}{\lambda^{\frac{1}{p}}}$$

Reparametrizando se tiene que:

$$\frac{1}{\lambda^{\frac{1}{p}}} = exp(x\beta)$$

Lambda queda en función de la combinación lineal de los predictores. Esta estructura puede cambiar dependiendo de la parametrización.

Ver fuente `https://stat.ethz.ch/education/semesters/ss2011/seminar/contents/handout_9.pdf`

Modelo en JAGS

```
model{
for(i in 1:n){
is.censored[i]~ dinterval(time[i], cens[i])
time[i] ~ dweib(alpha,lambda[i])
lambda[i] <- exp(-mu[i]*alpha)
mu[i]<-inprod(beta[],X[i,])
}

#Priordistributions
for(l in 1:Nbetas){beta[l]~dnorm(0,0.001)}
sigma~dunif(0,100)
alpha<- 1/sigma}
```

JAGS utiliza una parametrización distinta a la de R para el caso Weibull. Ver fuente `https://stats.stackexchange.com/questions/18550/how-do-i-parameterize-a-weibull-distribution-in-jags-bugs`

JAGS

```
library(rjags)

d.jags <- list(n=nrow(mastectomy), time=mastectomy$time, cens=cens, X=X,
                  is.censored=is.censored, Nbetas=ncol(X))

i.jags <- function(){ list(beta=rnorm(ncol(X)), sigma=runif(1)) }
p.jags <- c("beta", "alpha")

m1 <- jags.model(data=d.jags, file=model, inits=i.jags, n.chains=3)

update(m1, 1000) #Quema

res <- coda.samples(m1, variable.names=p.jags, n.iter=10000, thin=10)

summary(res[1])
1. Empirical mean and standard deviation for each variable,
   plus standard error of the mean:

           Mean     SD Naive SE Time-series SE
alpha    0.8481 0.1526 0.004826       0.005431
beta[1]  6.0768 0.6371 0.020145       0.037808
beta[2] -1.1595 0.6827 0.021590       0.039611
```