

Trabalho de Inferência

Estimação dos Parâmetros de um Modelo de Regressão Exponencial com Censura Intervalar

Jayme Gomes dos Santos Junior

Luciana Helena Kowalski

16/novembro/2018

Introdução

Censura intervalar ocorre quando o objeto de estudo é inspecionado em períodos de tempo distintos U e V e ocorre um evento de interesse neste intervalo de tal maneira que o exato momento do evento é desconhecido, e.g., $U < t \leq V$.

Existem três principais tipos de censura:

1. $U = 0$, chamamos de censura à esquerda;
2. $V = \infty$, censura à direita;
3. $0 < U < V = a$, onde $a \in \mathbb{R}$, ou seja, sabe-se o início e fim do intervalo, a **censura intervalar**(CI) que será tratada neste trabalho.

Censura intervalar ocorre com mais frequência em análises longitudinais e experimentações clínicas. Acontecendo mais em virtude do crescimento de dados envolvendo estudos sobre o vírus HIV.

Este tipo de dados eram analisados com técnicas tradicionais de sobrevivência, onde se assumia que o evento de interesse ocorreu no fim ou na metade do intervalo, o que gerava erros de estimação e viés nos resultados LINDSEY; RYAN (1998). Após investigações a distribuição exponencial se mostrou mais eficaz quando se sabe o momento das inspeções AL-TAWARAH; MACKENZIE (2002).

Para estimar os parâmetros da Regressão Exponencial com Censura Intervalar será utilizado o método da **Máxima Verissimilhança**(MV), assim como PENG (2009), mas com a utilização do software **R**(R CORE TEAM (2015)) para todas as simulações e análises.

Os objetivos do presente trabalho são: *i*) simular uma base de dados e os resultados com parâmetros fixos; *ii*) estimar os parâmetros usando o método (MV); *iii*) comparar os resultados dos parâmetros estimados com os da simulação utilizando intervalo de confiança com $\alpha=0,05$.

Modelo

Regressão Exponencial com Censura Intervalar consiste em um conjunto de intervalos independentes \mathbf{Y}_i que seguem uma distribuição exponencial de parâmetros λ_i , com $i = \{1, \dots, n\}$, logo:

$$\underline{\mathbf{Y}} \sim \text{Exp}(\underline{\lambda})$$

Onde cada \mathbf{Y}_i é um intervalo que contém o evento de interesse $\mathbf{Y}_i = (\mathbf{Y}_{iu} ; \mathbf{Y}_{iv})$, sendo u o limite inferior e v o limite superior do intervalo, u e $v > 0$, pois se trata de intervalos de tempo, logo $\underline{\mathbf{Y}}$ é um vetor positivo.

O parâmetro $\lambda_i = \beta_0 + \beta_1 x_i$ com $\underline{\mathbf{X}} = \{\mathbf{x}_i = [0 ; p] | i = \{1, \dots, n\}, p \in \mathbb{R}\}$ sendo um vetor de coeficientes de regressão.

Logo:

$$f(\underline{\mathbf{y}}; \underline{\lambda}) = (\beta_0 + \beta_1 x_i) e^{-(\beta_0 + \beta_1 x_i) y_i}, (\beta_0 + \beta_1 x_i) > 0, y_i > 0$$

Simulação

Como o objetivo é estimar os parâmetros β_0 e β_1 , primeiro serão fixados seus valores e os valores do vetor dos coeficientes de regressão (\underline{X}) como uma sequência de cem valores entre 0 e 2 igualmente espaçados. Então criar $\underline{\lambda} = \exp(\beta_0 + \beta_1 x_i)$ para garantir que todos os valores sejam positivos e pequenos, como mostrado pela função `summary()`.

```
b0 = log(1)
b1 = -2
x <- seq(0, 2, l = 100)
lambda = exp(b0 + b1*x)
summary(lambda)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.01832 0.04979 0.13540 0.24810 0.36790 1.00000
```

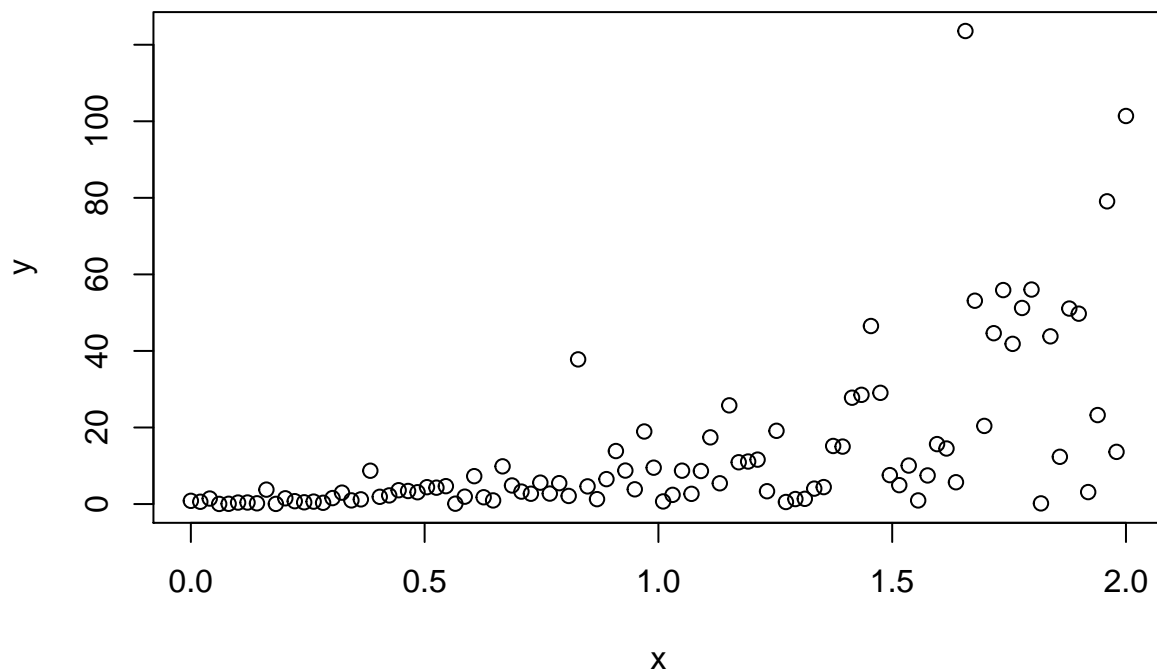
A maneira de simular os intervalos \underline{Y} foi primeiramente simular cem observações de variáveis aleatórias que sigam uma exponencial com os parâmetros calculados anteriormente utilizando a função `rexp()` que retorna valores que seguem a exponencial dado o parâmetro passado. Novamente foi usada a função `summary()` mostrando que todas as observações são positivas.

```
set.seed(123)
y = rexp(100, rate = lambda)
summary(y)
```

```
##      Min.   1st Qu.   Median     Mean  3rd Qu.     Max.
##  0.03565   1.54500   4.76500  14.01000  15.05000 123.60000
```

Para mostrar que as observações condizem com a distribuição exponencial, será usada a função `plot()` das observações dado os valores dos coeficientes de regressão(\underline{X}).

```
plot(y ~ x)
```



Simulando agora a (CI), criando \underline{U} (limites inferiores) e \underline{V} (limites superiores), usando as observações criadas anteriormente, subtraindo e somando 0.03 a cada uma delas. Este valor foi escolhido para garantir que \underline{Y}

> 0 como estipulado anteriormente. E criando uma base de dados com a função `data.frame()`, pois agora cada observação tem dois valores que são as inspeções dos objetos de estudo ao longo do tempo.

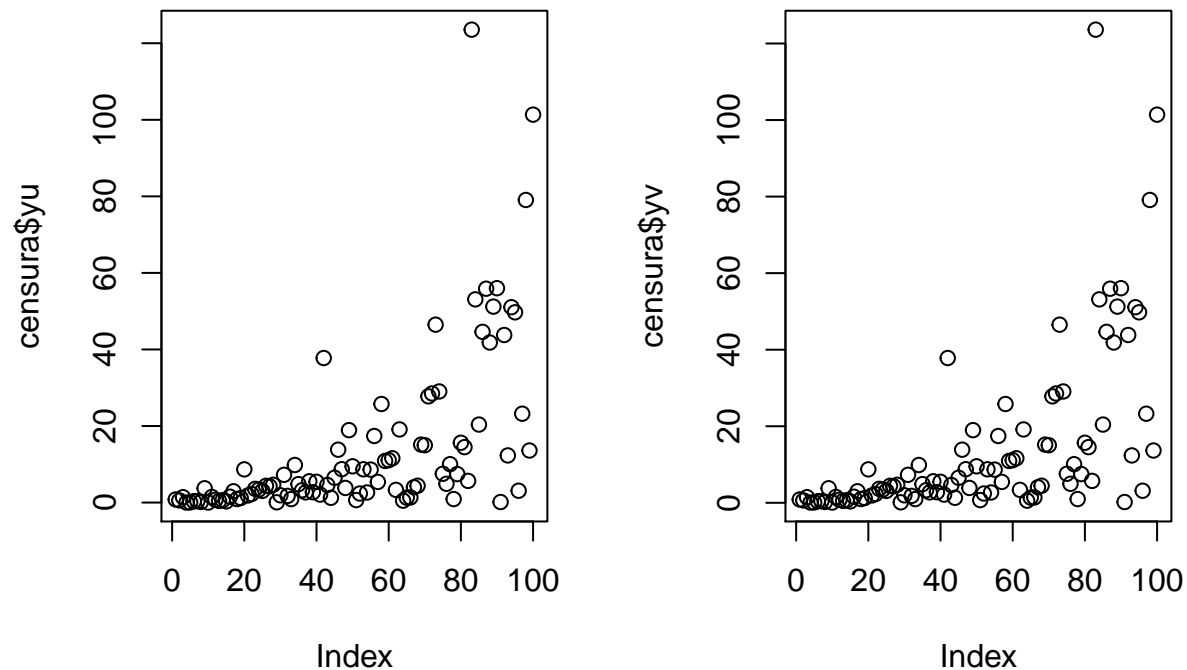
```
yu <- y - 0.03
yv <- y + 0.03
censura <- data.frame(yu, yv)
summary(censura)
```

```
##          yu          yv
## Min.   : 0.00565   Min.   : 0.06565
## 1st Qu.: 1.51464   1st Qu.: 1.57464
## Median : 4.73528   Median : 4.79528
## Mean   : 13.98039   Mean    : 14.04039
## 3rd Qu.: 15.02242   3rd Qu.: 15.08242
## Max.   :123.55311   Max.    :123.61311
```

Novamente a função `summary()` mostra que todos os intervalos são positivos como estipulado anteriormente.

Através dos gráficos é possível ver que os intervalos seguem uma distribuição exponencial.

```
par(mfrow = c(1,2))
plot(censura$yu)
plot(censura$yv)
```



Verossimilhança

Existe mais de uma maneira de chegar a função de verossimilhança deste modelo, portanto para este trabalho será utilizado um dos métodos utilizados na tese PENG (2009), onde:

$$L(\underline{y}; \underline{\lambda}) = \prod_{i=1}^n \int_{y_{iu}}^{y_{iv}} f(y_i; \lambda_i) dy,$$

$$= \prod_{i=1}^n \left[F(y_{iv}; \lambda_i) - F(y_{iu}; \lambda_i) \right]$$

Portanto a verossimilhança é subtrair da função acumulada da exponencial em y_v a função acumulada em y_u para cada y_i e depois multiplicar os resultados.

Log-Verossimilhança

$$l(\underline{y}; \underline{\lambda}) = \sum_{i=1}^n \left[\log \left(F(y_{iv}; \lambda_i) - F(y_{iu}; \lambda_i) \right) \right]$$

Estimador de Máxima Verossimilhança

Como as derivadas se tornam impossíveis analiticamente, a partir deste ponto os estimadores de máxima verossimilhança (EMV) para β_0 e β_1 serão feitos computacionalmente através de um maximizador numérico da função $l(\underline{y}; \underline{\lambda})$ usando a função `optim()` que é usada para minimizar funções com dois ou mais parâmetros, logo a função $l(\underline{y}; \underline{\lambda})$ será multiplicada por -1 para que a função funcione.

Função de Verossimilhança, Log-Verossimilhança e Cálculo da Verossimilhança da Simulação

```
# Verossimilhança
L_censura <- c()
for(i in 1:100){
  L_censura[i] <- pexp(censura$yv[i], rate = lambda[i]) - pexp(censura$yu[i], rate = lambda[i])
}
# Log-Verossimilhança
ll_censura <- log(L_censura)
# Cálculo da Verossimilhança
vero <- sum(-ll_censura)
```

Acima estão descritas computacionalmente as funções $L(\underline{y}; \underline{\lambda})$ e $l(\underline{y}; \underline{\lambda})$ respectivamente com o detalhe de que os produtórios e somatórios estarem omitidos, pois estão contemplados no cálculo da verossimilhança (estas funções executadas nessa ordem resultam exatamente na função de log-verossimilhança mostrada anteriormente), que é o número que representa a verossimilhança da simulação.

Como resultado temos que com $\beta_0 = 0$, $\beta_1 = -2$, temos uma verossimilhança = 585.911.

Agora deve ser criada uma função para calcular a verossimilhança com β_0 e β_1 desconhecidos, para testar o EMV.

```
#log-verossimilhança
ll <- function(theta, inferior, superior, x){
  lambda = exp(theta[1] + theta[2]*x)
  output <- -sum(log(pexp(superior, rate = lambda) - pexp(inferior, rate = lambda)))
  return(output)
}
# Avaliando nos betas da simulação e em outros para comparação
ll(theta = c(log(1), -2), inferior = censura$yu, superior = censura$yv, x = x)

## [1] 585.911
```

```
ll(theta = c(log(0.5), -0.5), inferior = censura$yu, superior = censura$yv, x = x)
```

```
## [1] 734.1352
```

Depois de pronta, a função foi avaliada usando os parâmetros β_0 e β_1 da simulação e depois com valores diferentes para mostrar que funciona corretamente.

Para calcular o EMV para os betas, será usada a função `optim()` para maximizar a $l(\underline{y}; \underline{\lambda})$

```
# Encontrando EMV Numericamente
```

```
oo <- optim(par = c(1, -3), fn = ll, inferior = censura$yu,
           superior = censura$yv, x = x, hessian = TRUE)
str(oo)
```

```
## List of 6
## $ par      : num [1:2] 0.0158 -2.0599
## $ value    : num 586
## $ counts   : Named int [1:2] 51 NA
## .. attr(*, "names")= chr [1:2] "function" "gradient"
## $ convergence: int 0
## $ message   : NULL
## $ hessian   : num [1:2, 1:2] 100 100 100 130
```

```
inv_Io <- solve(oo$hessian)
inv_Io
```

```
##           [,1]      [,2]
## [1,]  0.04343695 -0.03342970
## [2,] -0.03342970  0.03342323
```

Comparando os resultados do EMV com os da simulação temos que:

β_0 e β_1 simulados = 0 e -2, já os estimados = 0.0157913 e -2.0599495.

A verossimilhança simulada = 585.911 e a estimada = 585.755.

Para construir os intervalos de confiança para β_0 e β_1 , é preciso achar a matriz de informação de Fisher (matriz de informação esperada) $I_e(\beta_0, \beta_1) = -E[I_o(\beta_0, \beta_1)]$. Partindo da Matriz de informação observada $I_o(\beta_0, \beta_1)$.

```
# Matriz de Informação Observada
```

```
oo$hessian
```

```
##           [,1]      [,2]
## [1,]  99.99226 100.0116
## [2,] 100.01162 129.9503
```

```
# Utilizando a função 'solve()' chega-se na matriz de informação esperada
```

```
Ie <- solve(oo$hessian)
Ie
```

```
##           [,1]      [,2]
## [1,]  0.04343695 -0.03342970
## [2,] -0.03342970  0.03342323
```

Intervalo de Confiança

Construindo os intervalos de confiança para β_0 e β_1 com $\alpha=0,05$:

```
ic_Max <- oo$par + qnorm(0.975)*sqrt(diag(inv_Io))
ic_Min <- oo$par - qnorm(0.975)*sqrt(diag(inv_Io))
cbind(ic_Min, oo$par, ic_Max)
```

```
##           ic_Min           ic_Max
## [1,] -0.3926952  0.01579134  0.4242779
## [2,] -2.4182705 -2.05994948 -1.7016284
```

Com base nos intervalos de confiança é possível dizer que o EMV para β_0 e β_1 do modelo de regressão exponencial com censura intervalar funciona de forma satisfatória.

Referências

AL-TAWARAH, Y.; MACKENZIE, G. A logistic ph regression model for interval censored survival data., 2002.

LINDSEY, J. C.; RYAN, L. M. Methods for interval-censored data. **Statistics in Medicine**, v. 17, n. 2, p. 219–238, 1998. Wiley. Disponível em: <[https://doi.org/10.1002/\(sici\)1097-0258\(19980130\)17:2<219::aid-sim735>3.0.co;2-o](https://doi.org/10.1002/(sici)1097-0258(19980130)17:2<219::aid-sim735>3.0.co;2-o)>..

PENG, D. **Inferences in the Interval Censored Exponential Regression Model**. 2009.

R CORE TEAM. **R: A language and environment for statistical computing**. Vienna, Austria: R Foundation for Statistical Computing, 2015.