

Tarea 06

Dalia Camacho

26 de septiembre de 2018

```
suppressMessages(library(tidyverse))
library(knitr)
library(bootstrap)
```

6-Cobertura de intervalos de confianza

En este problema realizarás un ejercicio de simulación para comparar la exactitud de distintos intervalos de confianza. Simularás muestras de una distribución Poisson con parámetro $\lambda = 2.5$ y el estadístico de interés es $\theta = \exp^{-2\lambda}$

Sigue el siguiente proceso:

- i) Genera una muestra aleatoria de tamaño $n = 60$ con distribución $Poisson(\lambda)$, parámetro $\lambda = 2.5$ (en R usa la función `rpois()`).

```
set.seed(4446889)
lambda <- 2.5
n <- 60
muestra <- rpois(n, lambda)
```

- ii) Genera 10,000 muestras bootstrap y calcula intervalos de confianza del 95% para $\hat{\theta}$ usando 1) el método normal, 2) percentiles y 3) BC_a .

```
metodos <- c("Normal", "Percentiles", "BCa")
alfa <- 0.025
nsim <- 10000

# Valor verdadero
theta <- exp(-2*lambda)

# Valor plug in
theta_plug <- exp(-2*mean(muestra))

# Muestra bootstrap
theta_boot <- rerun(nsim, exp(-2*mean(sample(muestra, replace = TRUE)))) %>% flatten_dbl()
#Intervalos normales
SE_lambdas <- sd(theta_boot)
IC_Norm <- c(LB=theta_plug+qnorm(alfa)*SE_lambdas, UB=theta_plug+qnorm(1-alfa)*SE_lambdas)

# Intervalos con percentiles
IC_perc <- c(LB=quantile(theta_boot,alfa), UB=quantile(theta_boot,1-alfa))

# Intervalos BCa
# BCA parameters
BCA_par <- bcanon(muestra, nboot = nsim, theta = function(x){exp(-2*mean(x))})
z0 <- BCA_par$z0
acc <- BCA_par$acc
alfa1 <- pnorm(z0+(z0+qnorm(alfa))/(1-acc*(z0+qnorm(alfa))))
```

```

alfa2 <- pnorm(z0+(z0+qnorm(1-alfa))/(1-acc*(z0+qnorm(1-alfa))))
# Intervalos
IC_BCA <- c(LB= quantile(theta_boot, alfa1), UB=quantile(theta_boot, alfa2))

# Hacer tabla
df <- data.frame(rbind(IC_Norm, IC_perc, IC_BCA))
rownames(df) <- metodos
kable(df)

```

	LB	UB
Normal	-0.0000354	0.0110685
Percentiles	0.0022429	0.0131237
BCa	0.0020982	0.0122773

Revisa si el intervalo de confianza contiene el verdadero valor del parámetro ($\theta = \exp^{(-2 \cdot 2.5)}$) en caso de que no lo contenga registra si falló por la izquierda (el límite inferior >1) o falló por la derecha (el límite superior <1).

```

Fallo <- function(IC, verdadero){
  in_IC <- (verdadero >= IC[1]) & (verdadero <= IC[2])
  if(in_IC){
    0
  }else if(verdadero < IC[1]){
    -1
  }else{
    1
  }
}

# El valor verdadero está en el intervalo normal
Fallo(IC_Norm, theta)

## [1] 0

# El valor verdadero está en el intervalo de percentiles
Fallo(IC_perc, theta)

## [1] 0

# El valor verdadero está en el intervalo de BCa
Fallo(IC_BCA, theta)

## [1] 0

```

a) Repite el proceso descrito 1000 veces y llena la siguiente tabla:

Método	% fallo izquierda	% fallo derecha	Cobertura	Longitud promedio
Normal				
Percentiles				
BC_a				

La columna cobertura es una estimación de la cobertura del intervalo basada en las simulaciones, para calcularla simplemente escribe el porcentaje de los intervalos que incluyeron el verdadero valor del parámetro. La longitud promedio es la longitud promedio de los intervalos de confianza bajo cada método.

```
Tabla_eval <- function(n){
  repet      <- 1000
  Fallo_Norm <- c()
  Fallo_Perc <- c()
  Fallo_BCA  <- c()

  Long_Norm  <- c()
  Long_Perc  <- c()
  Long_BCA   <- c()

  LB_Norm    <- c()
  UB_Norm    <- c()
  LB_Perc    <- c()
  UB_Perc    <- c()
  LB_BCA     <- c()
  UB_BCA     <- c()

  for( i in 1:repet){
    muestra <- rpois(n, lambda)

    # Valor plug in
    theta_plug <- exp(-2*mean(muestra))

    # Muestra bootstrap
    theta_boot <- rerun(nsim, exp(-2*mean(sample(muestra, replace = TRUE)))) %>% flatten_dbl()
    #Intervalos normales
    SE_lambdas <- sd(theta_boot)
    IC_Norm    <- c(LB=theta_plug+qnorm(alfa)*SE_lambdas,
                  UB=theta_plug+qnorm(1-alfa)*SE_lambdas)
    LB_Norm    <- c(LB_Norm, IC_Norm[1])
    UB_Norm    <- c(UB_Norm, IC_Norm[2])

    # Intervalos con percentiles
    IC_Perc    <- c(LB=quantile(theta_boot,alfa),
                  UB=quantile(theta_boot,1-alfa))
    LB_Perc    <- c(LB_Perc, IC_Perc[1])
    UB_Perc    <- c(UB_Perc, IC_Perc[2])

    # Intervalos BCa
    # BCA parameters
    BCA_par <- bcanon(muestra, nboot = nsim,
                     theta = function(x){exp(-2*mean(x))})
    z0      <- BCA_par$z0
    acc      <- BCA_par$acc
    alfa1    <- pnorm(z0+(z0+qnorm(alfa))/(1-acc*(z0+qnorm(alfa))))
    alfa2    <- pnorm(z0+(z0+qnorm(1-alfa))/(1-acc*(z0+qnorm(1-alfa))))

    # Intervalos
    IC_BCA <- c(LB= quantile(theta_boot, alfa1),
               UB=quantile(theta_boot, alfa2))
  }
}
```

```

LB_BCA      <- c(LB_BCA, IC_BCA[1])
UB_BCA      <- c(UB_BCA, IC_BCA[2])

# El valor verdadero está en el intervalo normal
Fallo_Norm <- c(Fallo_Norm, Fallo(IC_Norm, theta))
Long_Norm  <- c(Long_Norm, IC_Norm[2]-IC_Norm[1])

# El valor verdadero está en el intervalo de percentiles
Fallo_Perc <- c(Fallo_Perc, Fallo(IC_Perc, theta))
Long_Perc  <- c(Long_Perc, IC_Perc[2]-IC_Perc[1])

# El valor verdadero está en el intervalo de BCa
Fallo_BCA <- c(Fallo_BCA, Fallo(IC_BCA, theta))
Long_BCA  <- c(Long_BCA, IC_BCA[2]-IC_BCA[1])
}

Eval_Norm <- c(Fallo_Izquierda=length(which(Fallo_Norm==1))/repet,
              Fallo_derecha=length(which(Fallo_Norm==1))/repet,
              Longitud_promedio=mean(Long_Norm))

Eval_Perc <- c(Fallo_Izquierda=length(which(Fallo_Perc==1))/repet,
              Fallo_derecha=length(which(Fallo_Perc==1))/repet,
              Longitud_promedio=mean(Long_Perc))

Eval_BCA <- c(Fallo_Izquierda=length(which(Fallo_BCA==1))/repet,
              Fallo_derecha=length(which(Fallo_BCA==1))/repet,
              Longitud_promedio=mean(Long_BCA))

# Hacer tabla
df      <- data.frame(rbind(Eval_Norm, Eval_Perc, Eval_BCA))
rownames(df) <- metodos

# Hacer gráfica
df2 <- suppressWarnings(data.frame(cbind(metodo=c(rep(metodos[1],repet),
                                                rep(metodos[2],repet),
                                                rep(metodos[3],repet)),
                                      LB=c(LB_Norm, LB_Perc, LB_BCA),
                                      UB=c(UB_Norm, UB_Perc, UB_BCA))))

df2$UB <- as.numeric(as.character(df2$UB))
df2$LB <- as.numeric(as.character(df2$LB))

g1 <- ggplot(df2)+theme_bw()+
  geom_line(aes(rep(1:repet,3), LB, col=metodo))+
  geom_line(aes(rep(1:repet,3), UB, col=metodo))+
  geom_hline(yintercept = theta, col="orange")+
  xlab("")+
  theme(legend.title = element_blank())+
  facet_wrap(~metodo)

```

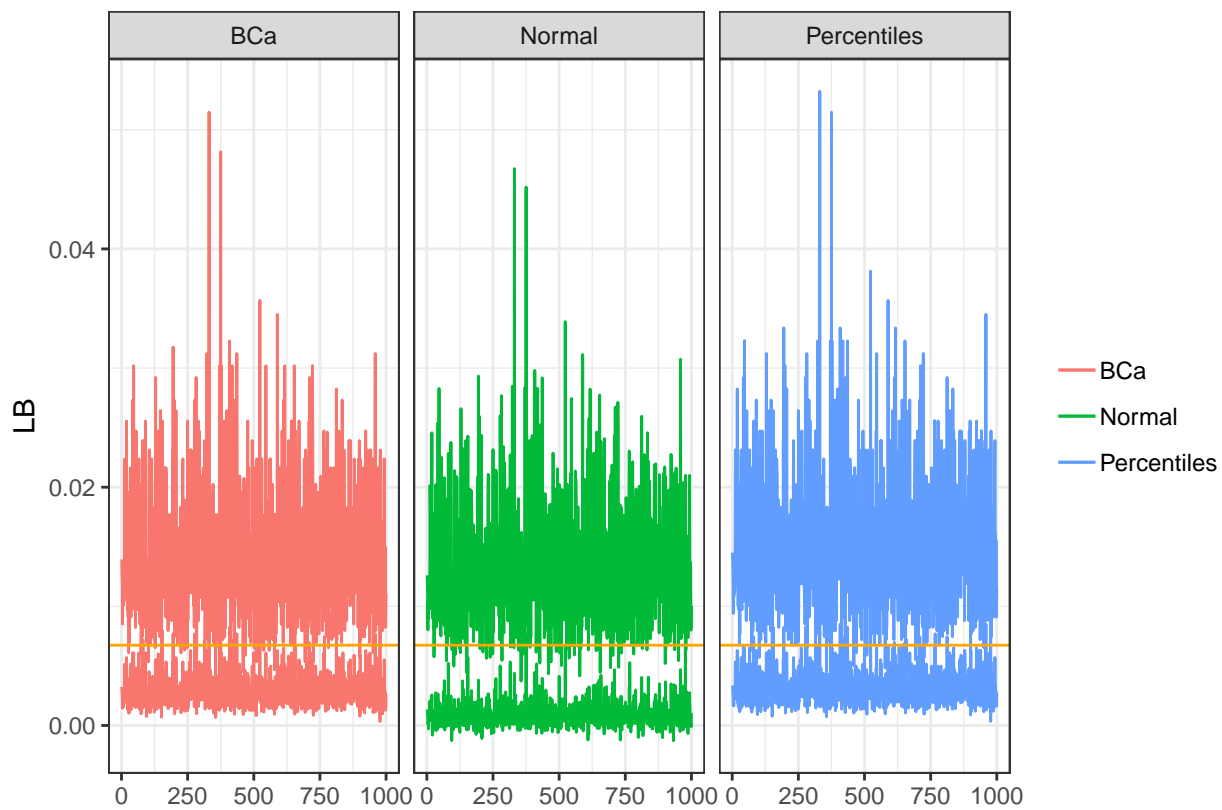
```
return(list(df, g1))
}
```

```
x <- Tabla_eval(60)
```

	Fallo_Izquierda	Fallo_derecha	Longitud_promedio
Normal	0.001	0.060	0.0123766
Percentiles	0.021	0.017	0.0121459
BCa	0.015	0.029	0.0114587

b) Gráfica

Realiza una gráfica de paneles, en cada panel mostrarás los resultados de uno de los métodos (normal, percentiles y BCa), el eje x corresponderá al número de intervalo de confianza ($1, \dots, 1000$) y en el vertical graficarás los límites de los intervalos, es decir graficarás 2 líneas (usa `geom_line()`) una corresponderá a los límites inferiores de los intervalos, y otra a los superiores.



Repite los incisos a) y b) seleccionando muestras de tamaño 300

```
x <- Tabla_eval(300)
```

	Fallo_Izquierda	Fallo_derecha	Longitud_promedio
Normal	0.008	0.034	0.0050053
Percentiles	0.030	0.021	0.0049858
BCa	0.025	0.024	0.0049264

