José Antonio Álvarez Ocete
77553417Q
joseantonio.alvarezo@estudiante.uam.es

Entrega de problemas Boostrap
Métodos Avanzados en Estadística

2021-10-25

Ejercicio 2.

Enunciado. Sea $X_1, \dots, X_n$ una muestra de n observaciones *iid* de una distribución F con esperanza μ y varianza σ^2, y sea $X_1^*, \dots, X_n^*$ una muestra de n observaciones de la distribución empírica de la muestra original F_n. Calcula las siguientes cantidades:

a) $\mathbb{E}_{F_n}(\bar{X}_n^*) := \mathbb{E}(\bar{X}_n^* | X_1, \dots, X_n)$
Esta esperanza nos supone un cálculo directo utilizando la linealidad de la esperanza:

$$\mathbb{E}_{F_n}(\bar{X}_n^*) = \mathbb{E}_{F_n}\left(\frac{1}{n}\sum_{i=1}^n X_i^*\right) = \frac{1}{n}\sum_{i=1}^n \mathbb{E}_{F_n}(X_i^*) \overset{\text{iid}}{=} \frac{1}{n} \cdot n\ \mathbb{E}_{F_n}(X_1^*) = \bar{x}$$

b) $\mathbb{E}_F(\bar{X}_n^*)$
Por un proceso análogo al anterior podemos obtener que:

$$\mathbb{E}_F(\bar{X}_n^*) = \mathbb{E}_F\left(\frac{1}{n}\sum_{i=1}^n X_i^*\right) = \frac{1}{n}\sum_{i=1}^n \mathbb{E}_F(X_i^*) \overset{\text{iid}}{=} \frac{1}{n} \cdot n\ \mathbb{E}_F(X_1^*) = \mathbb{E}_F(X_1^*)$$

Sin embargo, $\mathbb{E}_F(X_1^*)$ no se puede calcular directamente pues depende de la muestra tomada $X_1, \dots, X_n$. Podemos utilizar la **ley de la esperanza iterada**$\mathbb{E}[Y] = \mathbb{E}(\mathbb{E}(Y|X))$ y el valor calculado en el apartado anterior:

$$\mathbb{E}_F(X_1^*) = \mathbb{E}_F(\mathbb{E}(X_1^*|X_1, \dots, X_n)) = \mathbb{E}_F(\mathbb{E}_{F_n}(X_1^*)) = \mathbb{E}_F(\bar{x}) = \mu$$

c) $\text{Var}_{F_n}(\bar{X}_n^*) := Var(\bar{X}_n^* | X_1, \dots, X_n)$
Expresaremos esta varianza en función de la varianza muestra insesgada:

$$s^2 = \frac{1}{n-1}\left(\sum_i x_i^2 - n\bar{x}^2\right) = \frac{n}{n-1}\left(\frac{1}{n}\sum_i x_i^2 - \bar{x}^2\right)$$

Al ser insesgada, sabemos que $\mathbb{E}[s^2] = \sigma^2$. Esto nos será útil en el último apartado. Procedamos con el cálculo de la varianza:

$$\begin{aligned}
\text{Var}_{F_n}(\bar{X}_n^*) &\overset{\text{iid}}{=} \frac{1}{n^2}\sum_{i=1}^n \text{Var}_{F_n}(X_i^*) \\
&\overset{\text{iid}}{=} \frac{1}{n^2} \cdot n\ \text{Var}_{F_n}(X_1^*) \\
&= \frac{1}{n}\left(\mathbb{E}_{F_n}({X_1^*}^2) - \mathbb{E}_{F_n}(X_1^*)^2\right) \\
&= \frac{1}{n}\left(\frac{1}{n}\sum_{i=1}^n X_i^2 - \bar{x}^2\right) \\
&= \frac{n-1}{n^2}\underbrace{\frac{n}{n-1}\left(\frac{1}{n}\sum_{i=1}^n X_i^2 - \bar{x}^2\right)}_{s^2} \\
&= \frac{n-1}{n^2}s^2
\end{aligned}$$

d) $Var_F(\bar{X}_n^*)$

Para este último apartado haremos uso de la **ley de la varianza total**:

$$\text{Var}(Y) = \mathbb{E}(\text{Var}(Y|X)) + \text{Var}(\mathbb{E}(Y|X))$$

Procedemos con el cálculo de la varianza:

$$\begin{aligned}
\text{Var}_F(\bar{X}_n^*) &= \mathbb{E}_F(\underbrace{\text{Var}_{F_n}(\bar{X}_n^*)}_{\frac{n-1}{n^2}s^2}) + \text{Var}_F(\underbrace{\mathbb{E}_{F_n}(\bar{X}_n^*)}_{\bar{x}}) \\
&= \frac{n-1}{n^2}\mathbb{E}_F(s^2) + \text{Var}_F(\bar{x}) \\
&\overset{\text{iid}}{=} \frac{n-1}{n^2}\sigma^2 + \frac{1}{n}\text{Var}_F(X_1) \\
&= \frac{n-1}{n^2}\sigma^2 + \frac{1}{n}\sigma^2 \\
&= \frac{2n-1}{n^2}\sigma^2
\end{aligned}$$

Ejercicio 7.

Enunciado. Sea F una distribución con media μ, varianza σ^2 y coeficiente de asimetía:

$$\gamma = \frac{\mathbb{E}[(X-\mu)^3]}{\sigma^3}$$

Genera $R = 1000$ muestras de observaciones iid $X_1, \ldots, X_n$ con $X_i \equiv N(0,1)$ para $n = 100$. Para cada una de ellas, calcula tres intervalos de confianza bootstrap de nivel 95% para γ usando el método híbrido, el método normal y el método percentil. Determina el porcentaje de intervalos que contienen al parámetro en cada caso. Repite el ejercicio con muestras procedentes de una distribución exponencial de parámetro $\lambda = 1$.

Resolveremos este ejercicio utilizando R. Se adjunta el cuaderno de Rmarkdown, en caso de que el resto del ejercicio sea más cómodo de visualizar en dicho formato puede utilizar Rstudio para ello. Importamos los paquetes necesarios:

```
library(tidyverse)
library(gapminder)
library(comprehenr)
library(ggplot2)
library(dplyr)
theme_set(theme_bw())
```

Implementamos una función que calcula el coeficiente de asimetría muestral de una muestra dada. Utilizaremos el estimador natural: sustituir la esperanza por el promedio de los valores, la media por la media muestral y la varianza por la varianza muestral, siendo conscientes de que esta varianza muestra es s^2 y no σ^2.

```
asymmetry_coefficient <- function(muestra) {
  mean <- mean(muestra)
  sigma <- sd(muestra)
mean((muestra - mean)^3 / sigma)
}
```

Implementamos una función que pinta los intervalos alrededor del valor real θ, sirviéndonos del código proporcionado en las diapositivas.

```r
plot_intervals <- function(intervals, hits, theta, method_name, distribution_name) {
  m <- nrow(intervals)
  df <- data.frame(ic_min <- intervals[,1],
                   ic_max <- intervals[,2],
ind = 1:m,
hits = hits)

  gg <- ggplot(df) +
    geom_linerange(aes(xmin=ic_min, xmax=ic_max, y=ind, col=hits)) +
    scale_color_hue(labels = c("NO", "SÍ")) +
    geom_vline(aes(xintercept = theta), linetype = 2) +
    theme_bw() +
    labs(y = 'Muestras', x = 'Intervalos (nivel 0.95)',
title = paste('IC -', method_name, 'para', distribution_name, sep=' '))
print(gg)
}
```

Implementamos dos funciones auxiliares adicionales. La primera muestra una distribución (normal o exponencial) `n` veces. La segunda calcula el intervalo de confianza del coeficiente de asimetría utilizando el método proporcionado por parámetro.

```r
sample_dist <- function(distribution, n) {
if (distribution == 'normal') {
rnorm(n, 0, 1)
  } else if (distribution == 'exponential') {
rexp(n, 1)
  } else {
print('Distribution not supported')
  }
}

compute_interval <- function(method, gammas_bootstrap, gamma_original, alfa) {
if (method == 'Método híbrido') {
# Metodo hibrido
    n = length(gammas_bootstrap)
    T_bootstrap <- sqrt(n) * (gammas_bootstrap - gamma_original)
    ic_min <- gamma_original -  quantile(T_bootstrap, 1-alfa/2)/sqrt(n)
    ic_max  <- gamma_original -  quantile(T_bootstrap, alfa/2)/sqrt(n)
c(ic_min, ic_max)
  } else if (method == 'Método normal') {
# Metodo normal
    et_boostrap <- sd(gammas_bootstrap)
    ic_min <- gamma_original + qnorm(alfa/2, 0, 1)*et_boostrap
    ic_max  <- gamma_original - qnorm(alfa/2, 0, 1)*et_boostrap
c(ic_min, ic_max)
  } else if (method == 'Método percentil') {
# Metodo percentil
    ic_min <- quantile(gammas_bootstrap, alfa/2)
    ic_max  <- quantile(gammas_bootstrap, 1-alfa/2)
c(ic_min, ic_max)
  } else {
print('Method not supported')
  }
}
```

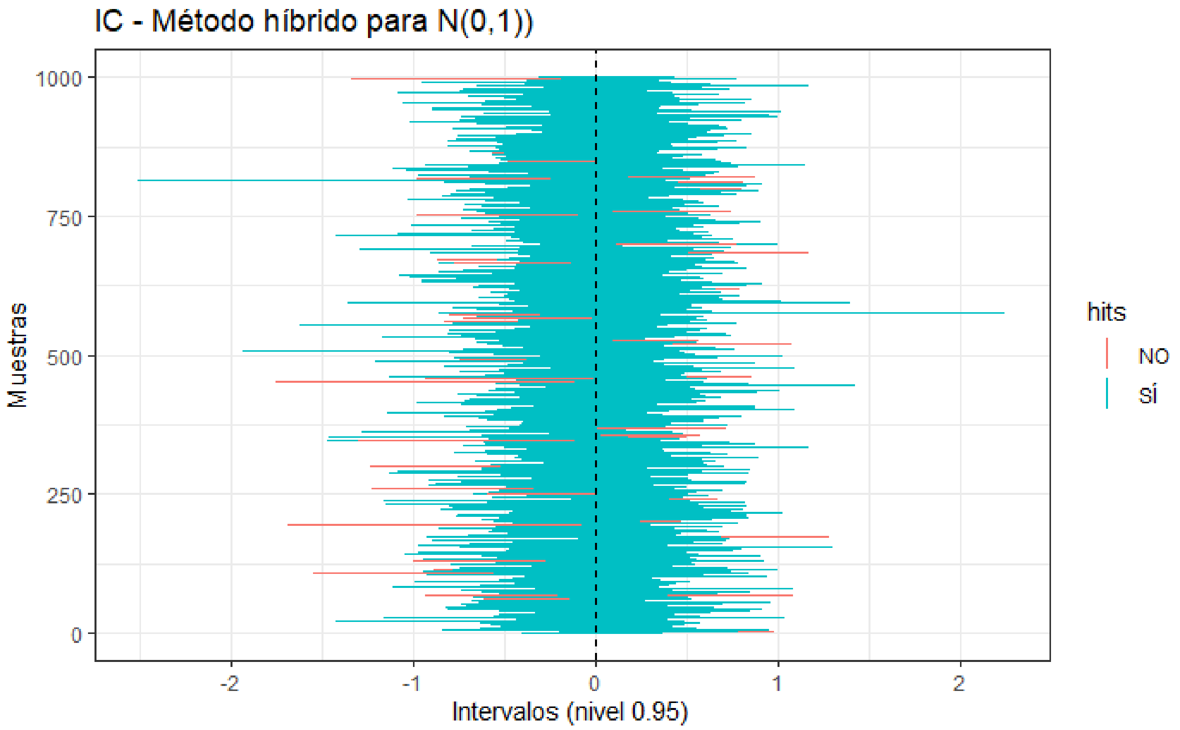
Utilizando las funciones anteriores implementamos una función final que, dada una distibución y un método, calcula `n=100` muestras originales, remuestrea `R=1000` veces y calcula el intervalor de confianza para el coeficiente de asimetría. Repetimos este proceso `m=1000` veces y dibujamos los distintos intervalos de confianza. Adicionalmente se mostrará la precisión del método.

```r
exercise_2 <- function (distribution, method) {
  R <- 1000
  n <- 100
  m <- 1000

  alfa <- 0.05
  theta = if (distribution == 'normal') 0 else 2

  intervalos <- NULL
  aciertos <- NULL

for (i in 1:m) {
# Usar la distribución para muestrear
    muestra_original <- sample_dist(distribution, n)
    gamma_original <- asymmetry_coefficient(muestra_original)

# Muestreo boostrap y computar los estimadores boostrap
    muestras_bootstrap <- sample(muestra_original, n*R, rep = TRUE)
    muestras_bootstrap <- matrix(muestras_bootstrap, nrow = n)
    gammas_bootstrap <- apply(muestras_bootstrap, 2, asymmetry_coefficient)

# Computar el intervalo y el acierto
    interv <- compute_interval(method, gammas_bootstrap, gamma_original, alfa)
    intervalos <- rbind(intervalos, interv)
    aciertos <- c(aciertos, interv[1] < theta & interv[2] > theta)
  }

# Cálculo del accuracy
  acc = 100 * length(aciertos[aciertos %in% TRUE]) / m
       print(paste(method, ' - accuracy: ', acc, '%', separator=''))

# Gráfico
  distribution_name = if (distribution == 'normal') 'N(0,1))' else 'exp(1)'
plot_intervals(intervalos, aciertos, theta,
                 method, distribution_name)
}
```

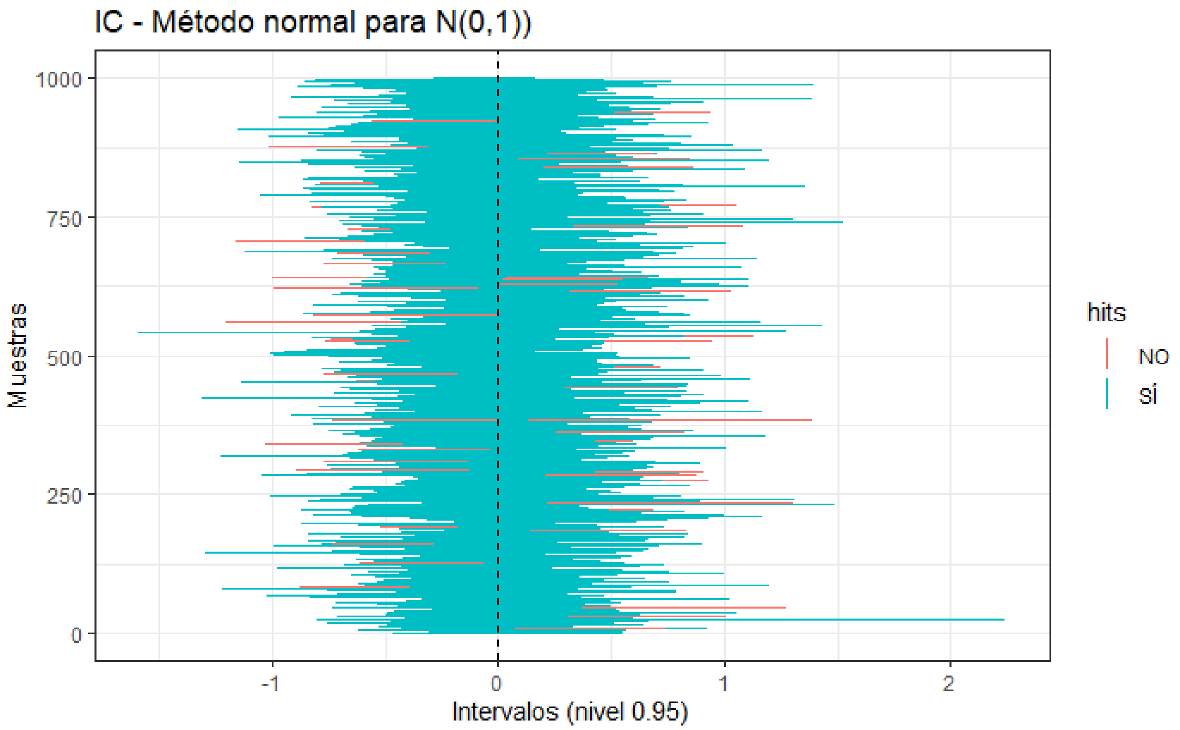
Utilizamos la función anterior para mostrar los distintos intervalos de confianza para la normal, así como su precisión.

```r
set.seed(123)

methods <- c('Método híbrido', 'Método normal', 'Método percentil')
for (method in methods) {
exercise_2('normal', method)
}
```

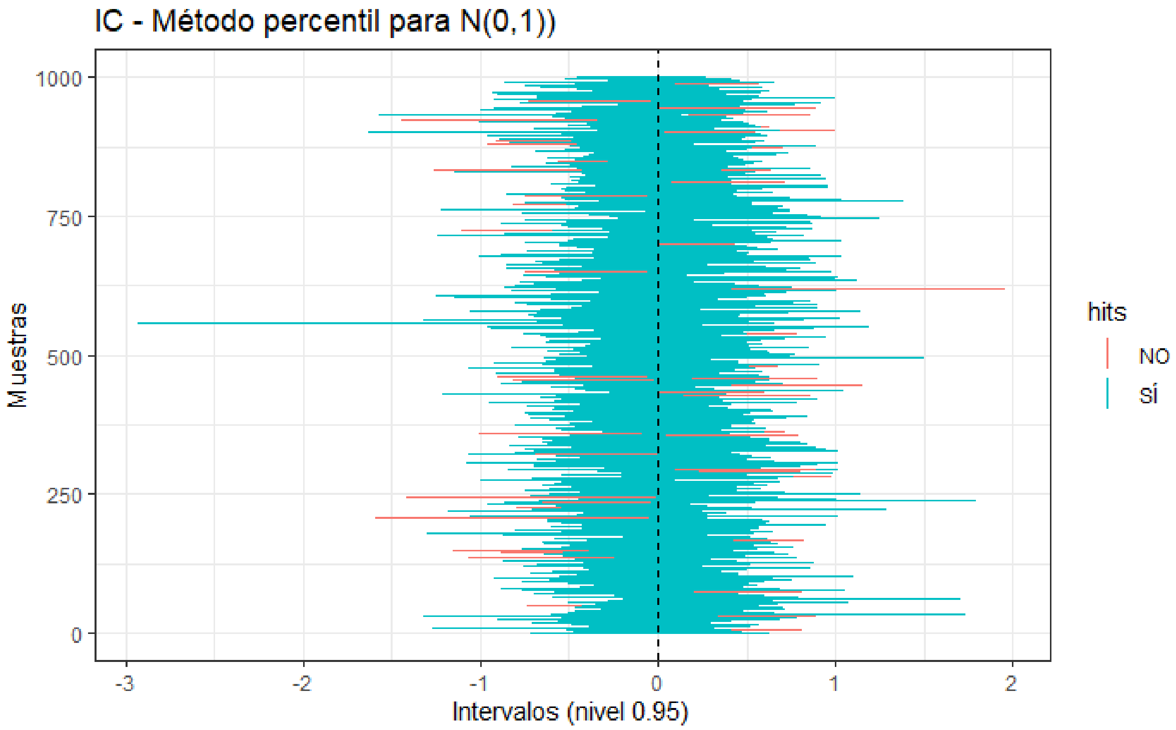
```
## [1] "Método híbrido  - accuracy:  95.2 % "
```



[1] "Método normal - accuracy: 94.8 % "



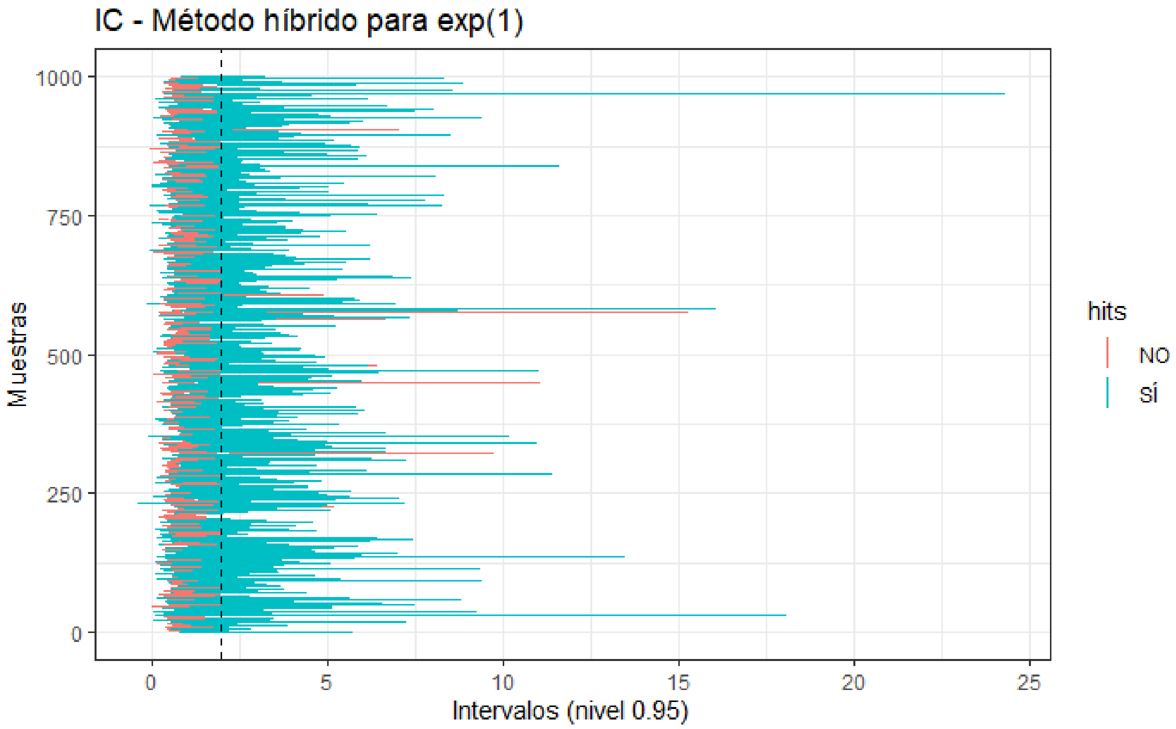
[1] "Método percentil - accuracy: 94.6 % "



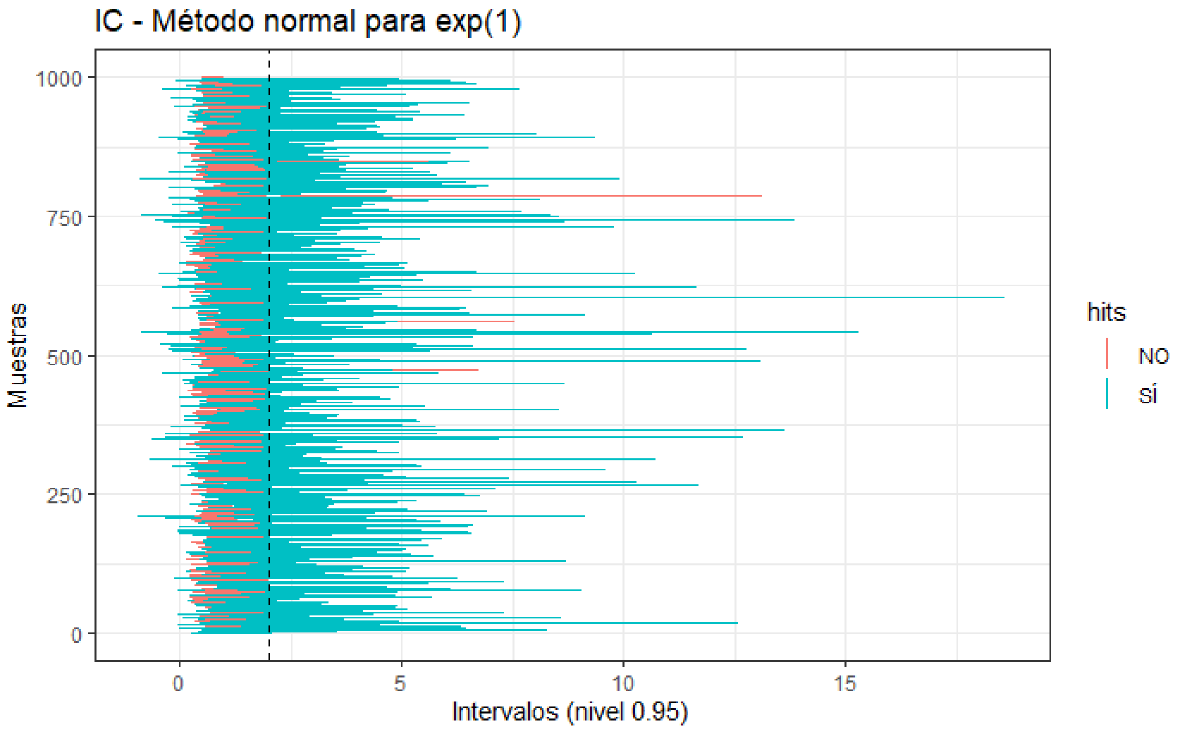
Obtenemos así precisiones cercanas al 95%. Este resultado era de esperar pues tomamos `alfa=0.05`. Repetimos el experimento para la distribución exponencial con parámetro $\lambda = 1$.

```
methods <- c('Método híbrido', 'Método normal', 'Método percentil')
for (method in methods) {
exercise_2('exponential', method)
}
```

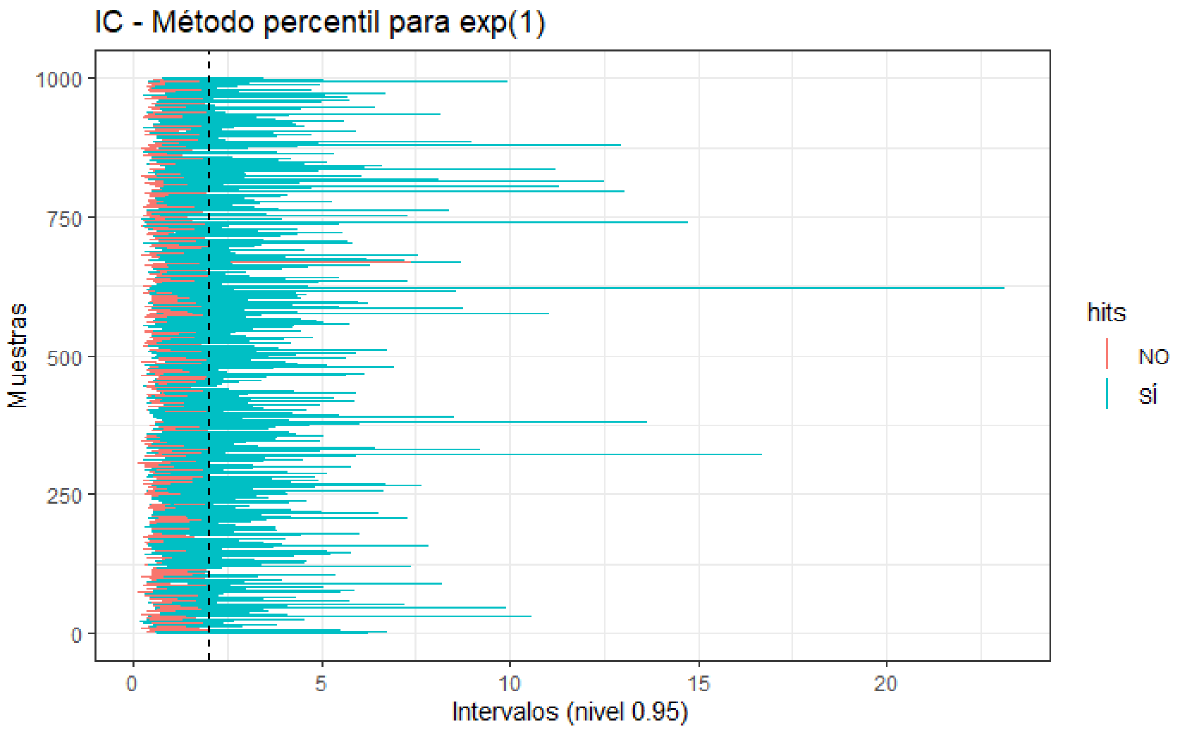
```
## [1] "Método híbrido  - accuracy:  65.3 % "
```




```
## [1] "Método normal  - accuracy:  68 % "
```




```
## [1] "Método percentil  - accuracy:  63.6 % "
```



En este caso, el coeficiente de asimetría real es 2. Sin embargo, vemos como obtenemos una precisión mucho menor: entorno al 65% en los diferentes métodos. Esto puede deberse a que el coeficiente de asimetría no se estime correctamente utilizando boostrap.