Universidad de Oviedo

Técnicas de remuestreo bootstrap

Raúl Blázquez Bullón

Supervisado por: Arís Fanjul Hevia y Raúl Pérez Fernández

Universidad de Oviedo

Facultad de Ciencias

Grado en Matemáticas

19 de junio de 2023

Índice general

Capítulo 1

Introducción

La estadística es una de las ramas de las matemáticas con más aplicaciones en el resto de ciencias, desde la física hasta la biología o medicina. Como ocurre normalmente en las matemáticas, la estadística está ligada a razonamientos lógicos y abstractos que no siempre nos resultan fáciles de comprender o de aplicar. En particular, cuando buscamos inferir propiedades de una población, podemos encontrarnos situaciones en las que no podemos determinar el comportamiento de cierto estimador de forma exacta usando métodos analíticos. Estas barreras pueden evitarse cuando empleamos el método bootstrap.

La palabra bootstrap (que en castellano significa trabilla de una bota) hace referencia a la expresión anglosajona "to pull oneself up by one's bootstrap", que podríamos traducir al castellano como salir de una situación por medios propios. Suele considerarse que esta expresión deriva del siguiente fragmento:

> "The Baron had fallen to the bottom of a deep lake. Just when it looked like all was lost, he thought to pick himself up by his own bootstraps".
>
> El Barón había caído al fondo de un profundo lago. Solo cuando parecía que todo estaba perdido, pensó en levantarse por sus propios medios.

del libro The Suprising Adventures of Baron Munchausen escrito por Rudolph Erich Raspe en el siglo XVIII [5].

Efron introdujo este método en el año 1979 [3] basándose en la técnica jackknife propuesta por Quenouille en 1949 [10], la cual se utilizaba para hallar el sesgo y varianza de un estimador. El método de Efron se basa en remuestrear a partir de una muestra dada usando

simulaciones con un ordenador (normalmente con el método de Monte Carlo), reemplazando de esta manera aproximaciones asintóticas complicadas o no adecuadas. Es por ello que el bootstrap se ha visto beneficiado de los avances computacionales de las últimas décadas, aumentando el número de simulaciones y disminuyendo los tiempos de ejecución.

A diferencia de los métodos tradicionales de estimación que se valen de hipótesis sobre la distribución de los datos para determinar el comportamiento de ciertos estadísticos, el método bootstrap no utiliza ninguna hipótesis sobre la distribución subyacente para obtener información sobre los mismos estadísticos. Esta técnica es especialmente útil cuando no se conoce la distribución de los datos o es difícil de aproximar con métodos asintóticos, haciendo que el bootstrap sea una herramienta muy versátil (y sobre todo fácil de utilizar).

Desde su introducción en 1979, se han desarrollado numerosas extensiones del método bootstrap, como el bootstrap bayesiano [13] o el método bias-corrected accelerated [4], aunque nosotros nos vamos a restringir a extensiones menos sofisticadas, como pueden ser el bootstrap paramétrico, el bootstrap simetrizado o el bootstrap suavizado (una extensión no paramétrica).

A lo largo de esta memoria estudiaremos el método bootstrap y cómo de bien funciona a la hora de aproximar la distribución de ciertos estimadores o en el cálculo de intervalos de confianza. Además, usaremos tres extensiones de la técnica bootstrap cuando conozcamos ciertas propiedades de la población de la que proviene la muestra, como puede ser la familia paramétrica a la que pertenece o la simetría. A la hora de obtener resultados de los que extraer conclusiones usaremos el software estadístico R [11].

1.1. Motivación

En el contexto de la inferencia estadística, es común encontrarnos con problemas de estimación de parámetros o de funciones de distribución. Normalmente, hacemos suposiciones sobre la distribución subyacente y obtenemos estimaciones de los parámetros de interés. Muchas veces es difícil encontrar una expresión de los estimadores y recurrimos a aproximaciones asintóticas, las cuales no funcionan bien si el tamaño de la muestra es pequeño. A diferencia de esta forma de proceder, el método bootstrap es una técnica de remuestreo que se basa en la distribución de la muestra y no asume hipótesis sobre la distribución de los datos. Es decir, dada una muestra sobre la que no conocemos nada, el método bootstrap nos permite inferir

de manera aproximada la distribución del estimador que estamos interesados en estudiar. Así, podemos hallar de manera aproximada el sesgo, varianza o cualquier momento de orden $k \in \mathbb{N}$ de dicho estimador.

1.2. Objetivos

La finalidad principal de este trabajo es entender cómo funciona el método de remuestreo bootstrap, las ventajas y desventajas que tiene sobre otras técnicas de inferencia. Asimismo, buscamos comparar los resultados que se obtienen usando bootstrap con los resultados de la inferencia clásica.

Un punto importante del trabajo es explorar las variantes del bootstrap y si presentan alguna mejora respecto del bootstrap original (conocido como bootstrap uniforme). En particular investigaremos en profundidad el bootstrap suavizado, viendo qué condiciones optimizan los resultados de esta extensión del bootstrap. Además, un objetivo fundamental es aplicar estas técnicas a ejemplos prácticos para evaluar su desempeño.

De forma más específica, estudiaremos cómo se hallan intervalos de confianza bootstrap, presentando tanto su cobertura como su longitud esperada, y si son buenas aproximaciones de los intervalos analíticos.

1.3. Estructura del trabajo

Las ideas que hemos seguido a la hora de redactar esta memoria han sido principalmente las de [2], y la estructura se ha basado en dar una introducción teórica del método a exponer seguido de un ejemplo práctico en el que aplicamos dicho método.

En el Capítulo §2 introducimos el método bootstrap uniforme y lo usamos para calcular los momentos de la cuasivarianza muestral.

En el Capítulo §3 presentamos tres modificaciones del bootstrap uniforme: paramétrico, simetrizado y suavizado. Para cada uno de ellos hallamos los momentos de la cuasivarianza tal y como hicimos en el capítulo anterior. Además, presentamos un estudio sobre la sensibilidad de los resultados al cambiar el parámetro de ventana y el núcleo usados en el bootstrap suavizado. Cerramos este capítulo con un análisis comparativo de los cuatro tipos de bootstrap.

En el Capítulo §4 presentamos tres formas de calcular intervalos de confianza bootstrap: percentil básico, t y t simetrizado. En cada uno de ellos calculamos un intervalo de confianza usando los cuatro métodos de los Capítulos §2 y §3, para luego estudiar la cobertura y longitud esperada de los intervalos bootstrap.

En el Capítulo §5 proporcionamos tres situaciones diferentes en las que podemos aplicar el método bootstrap para el cálculo de intervalos de confianza. Aquí utilizamos el bootstrap para tres distribuciones distintas (normal, exponencial y de Poisson).

En el Capítulo §6 resumimos las conclusiones del trabajo.

Finalmente, en el Apéndice A incluimos los códigos escritos en R necesarios para reproducir los resultados obtenidos en esta memoria. Es importante destacar que los códigos que hemos desarrollado están diseñados de manera legible y bien organizada, lo que facilita su comprensión y ejecución.

Capítulo 2

Motivación del bootstrap

Supongamos que tenemos una muestra aleatoria simple (m.a.s.) $\vec{X} = (X_1, \ldots, X_n)$, donde n es el tamaño muestral, que viene de una variable aleatoria con distribución F y que nos gustaría obtener información sobre un parámetro $\theta = \theta(F)$. Para ello, uno puede considerar el estadístico $T = T\left(\vec{X}, F\right) = \theta(F_n) - \theta(F) = \widehat{\theta} - \theta$, donde F_n es la función de distribución empírica. Una vez tenemos este estadístico, lo que se suele hacer es intentar calcular su distribución exacta o aproximarla cuando $n \to \infty$.

En el contexto bootstrap sustituimos la función de distribución F, que desconocemos, por una estimación $\widehat{F}$. En función de la estimación que consideremos tendremos un tipo distinto de bootstrap, por ejemplo si $\widehat{F} = F_n$ el método se denomina bootstrap uniforme, o si $\widehat{F} = F_{\widehat{\theta}}$, siendo $\widehat{\theta} = \theta(F_n)$ una estimación de θ, el método se llama bootstrap paramétrico. Con la estimación $\widehat{F}$ podemos generar remuestras (ligadas a la muestra original):

$$\vec{X}^* = (X_1^*, \ldots, X_n^*),$$

donde las X_i^* verifican que su función de distribución es $\widehat{F}$ para $i \in \{1, \ldots, n\}$. Entonces ahora podemos considerar el estadístico:

$$T^* = T\left(\vec{X}^*, \widehat{F}\right) = \widehat{\theta}^* - \widehat{\theta},$$

y la idea es que la distribución de $\widehat{\theta}^*$ en torno a $\widehat{\theta}$ aproxima la distribución de $\widehat{\theta}$ en torno a θ, y así aproximaremos la distribución de T por la de T^*. Normalmente, usaremos el método de Monte Carlo para hallar la distribución de T^*.

Con todo lo anterior, la ventaja del método bootstrap es que no necesita hipótesis sobre la forma de generar las remuestras. En el caso del bootstrap uniforme, el procedimiento

que seguiremos para crear dichas remuestras será, para cada $i \in \{1, \ldots, n\}$, simular X_i^*, asumiendo que $P(X_i^* = X_j) = 1/n$ con $j \in \{1, \ldots, n\}$. Para implementarlo podemos usar una variable uniforme $U_i \sim \mathcal{U}(0,1)$ y hacer $X_i^* = X_{\lfloor nU_i \rfloor + 1}$ para cada $i \in \{1, \ldots, n\}$, siendo $\lfloor x \rfloor$ la función suelo que devuelve la parte entera de $x \in \mathbb{R}$, esto es, el mayor número entero k tal que $k \leq x$.

Es importante destacar que en la práctica la función de distribución F suele ser desconocida, pero en los ejemplos que estudiaremos en adelante la conocemos porque dichos ejemplos son situaciones teóricas que construimos nosotros. Además, llamaremos "desconocido" a los parámetros o a la información que en un caso real no tendríamos.

Veremos a continuación dos ejemplos en los que utilizamos el método bootstrap para aproximar la función de distribución de un estimador y sus momentos a partir de datos que provienen de una distribución normal (para el primer ejemplo) y de una exponencial (para el segundo).

Ejemplo 2.1. Se han registrado 25 observaciones de una variable aleatoria que sigue una distribución normal de media 17.4 (desconocido) y desviación típica 2.1 (desconocido). Halla la distribución de la cuasivarianza muestral y los valores de sus momentos poblacionales, siendo la cuasivarianza muestral el siguiente estadístico:

$$\widehat{S}_X^2 = \frac{1}{n-1} \sum_{i=1}^{n} \left(X_i - \overline{X}\right)^2.$$

Resolveremos el ejemplo de dos formas:

- Clásica: ya que los datos provienen de una distribución normal, podemos aplicar el Teorema de Fisher sobre la cuasivarianza muestral:

$$\widehat{S}_X^2 = \frac{1}{n-1} \sum_{i=1}^{n} \left(X_i - \overline{X}\right)^2 \Rightarrow T := \frac{n-1}{\sigma^2} \widehat{S}_X^2 \sim \chi_{n-1}^2, \tag{2.1}$$

  siendo σ la desviación típica de la variable aleatoria normal.

- Bootstrap: en el contexto bootstrap generamos remuestras a partir de la muestra inicial, y a partir de estas podemos calcular la cuasivarianza muestral de la remuestra asociada a la b-ésima réplica, $\widehat{S}_X^{2(b)}$, donde $b \in \{1, \ldots, B\}$ y B es el número de réplicas, y el papel de σ^2 ahora lo juega $\widehat{S}_X^2$. Entonces vamos a ir generando distintas remuestras y

calculando el estadístico T de la Ecuación (2.1) obtendremos una distribución T^* que se asemejará a una χ^2_{n-1}.

Vamos a representar un histograma de la distribución bootstrap. Con el fin de visualizar la dependencia del método bootstrap con la muestra inicial, resolveremos el ejemplo para dos muestras diferentes pero de mismo tamaño muestral. Para ello, seguiremos los siguientes pasos (ver el Código A.1):

(1) Creamos dos muestras normales aleatorias con los parámetros del enunciado (en R usaremos la función `rnorm`).

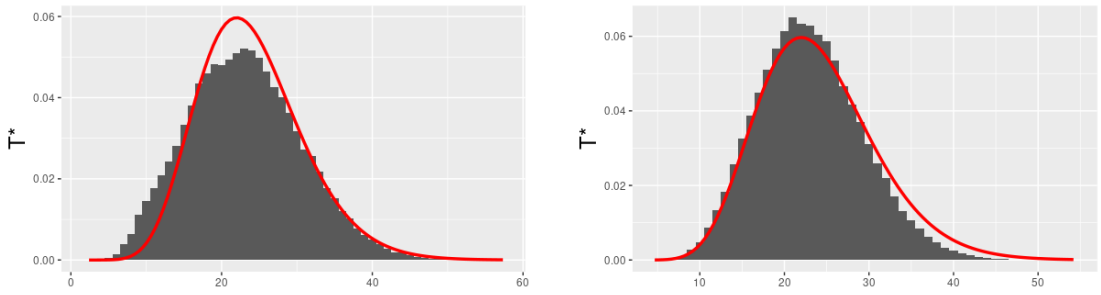
(2) Calculamos la cuasivarianza de cada muestra, $\widehat{S}^2_X$ (con el comando `var` en R).

(3) Creamos una remuestra para cada muestra (en R usamos la función `sample` con reemplazamiento), y de esta remuestra hallamos su cuasivarianza bootstrap, $\widehat{S}^{2(b)}_X$. Calculamos para cada remuestra el valor del estadístico T de la Ecuación (2.1) cambiando σ^2 por $\widehat{S}^2_X$, y $\widehat{S}^2_X$ por $\widehat{S}^{2(b)}_X$ y guardamos este valor.

(4) Repetimos el punto anterior para $b \in \{1, \ldots, B\}$ (en R tenemos la función `replicate`). En este caso, $B = 10^5$.

(5) Representamos los resultados, dibujando por encima la densidad de χ^2_{n-1}.

En la Figura 2.1 podemos observar las aproximaciones de la distribución obtenidas con el método bootstrap para las dos muestras distintas, que se asemejan en cierta medida a la densidad de la χ^2_{n-1}.

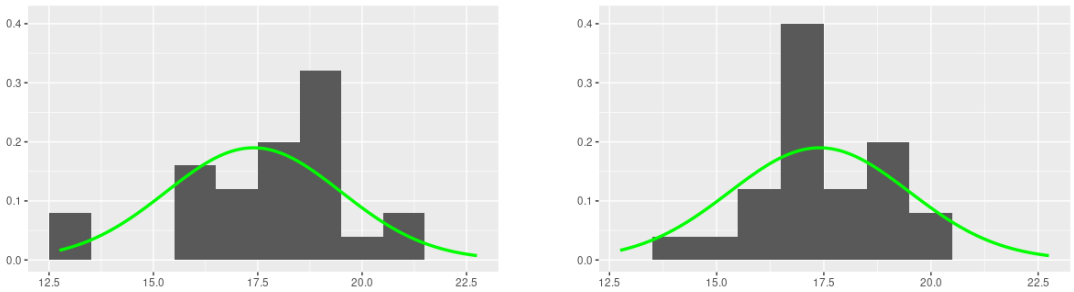


(a) Histograma de T^* (muestra 1). (b) Histograma de T^* (muestra 2).

Figura 2.1: Histogramas de T^* junto con la densidad de χ^2_{n-1}.

Como podemos observar, la distribución bootstrap de la muestra 2 (Figura 2.1b) parece

aproximar un poco mejor la densidad de la χ^2 en la parte izquierda, lo que nos quiere decir que el método bootstrap depende de la muestra inicial y de lo bien que represente a la población. En la Figura 2.2 vemos los histogramas de las muestras iniciales, donde podemos observar que la muestra 1 presenta una ligera asimetría así como un valor relativamente alejado del grueso de la muestra, mientras que la muestra 2 tiene la mayor parte de sus valores concentrados alrededor de la media.



(a) Histograma de la muestra 1.

(b) Histograma de la muestra 2.

Figura 2.2: Histogramas de las muestras junto con la densidad de $\mathcal{N}(17.4, 2.1)$.

Ahora vamos a hallar los momentos de $T \sim \chi^2_k$ para compararlos con los momentos de las dos distribuciones bootstrap que hemos generado. Usando la siguiente expresión cerrada para los momentos de una χ^2_k:

$$\mathrm{E}\left(T^m\right) = \int_{\mathbb{R}} t^m \, f_{\chi^2_k}(t) \, \mathrm{d}t = 2^m \frac{\Gamma(m + k/2)}{\Gamma(k/2)}, \tag{2.2}$$

y evaluando para $m \in \{1, 2, 3, 4\}$ y para $k = n - 1 = 24$ obtenemos:

$$\mathrm{E}(T) = n - 1 = 24, \quad \mathrm{E}\left(T^2\right) = 624, \quad \mathrm{E}\left(T^3\right) = 17472, \quad \mathrm{E}\left(T^4\right) = 524160. \tag{2.3}$$

Por otro lado, usando las distribuciones que obtuvimos con el método bootstrap podemos seguir los siguientes pasos (ver el Código A.2):

(1) Para la distribución bootstrap de la muestra i-ésima, con $i \in \{1, 2\}$, aproximamos el momento j-ésimo con la media de $(T_i^*)^j$ (en R podemos elevar el vector con la distribución bootstrap a j y usar la función `mean`).

(2) Iterar con un bucle doble *for* en i de 1 a 2 y en j de 1 hasta 4 el punto anterior.

Los resultados obtenidos son los siguientes:

$$\mathrm{E}^*(T_1^*) = 23.03, \quad \mathrm{E}^*\left((T_1^*)^2\right) = 586.31, \quad \mathrm{E}^*\left((T_1^*)^3\right) = 16186.80, \quad \mathrm{E}^*\left((T_1^*)^4\right) = 478228.68,$$
$$\mathrm{E}^*(T_2^*) = 23.04, \quad \mathrm{E}^*\left((T_2^*)^2\right) = 569.06, \quad \mathrm{E}^*\left((T_2^*)^3\right) = 14954.23, \quad \mathrm{E}^*\left((T_2^*)^4\right) = 415460.73.$$

Recordando que la expresión para el error relativo en porcentaje entre una cantidad observada x_{obs} y su valor teórico x_{teo} es:

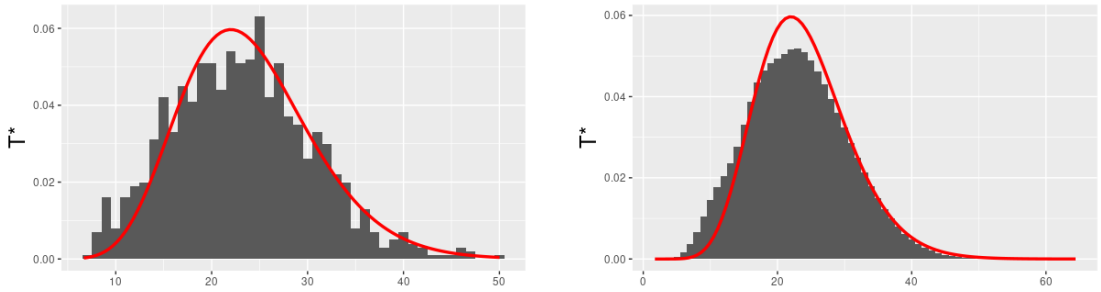
$$\varepsilon = \frac{|x_{obs} - x_{teo}|}{x_{teo}} \cdot 100\,\%,$$

podemos calcular el error relativo en porcentaje para cada muestra y momento con respecto a los valores que obtuvimos en la Ecuación (2.3):

$$\begin{aligned} &\varepsilon\left(T, T_1^*\right) = 4\,\%, \quad \varepsilon\left(T^2, (T_1^*)^2\right) = 6\,\%, \quad \varepsilon\left(T^3, (T_1^*)^3\right) = 7\,\%, \quad \varepsilon\left(T^4, (T_1^*)^4\right) = 9\,\%, \\ &\varepsilon\left(T, T_2^*\right) = 4\,\%, \quad \varepsilon\left(T^2, (T_2^*)^2\right) = 9\,\%, \quad \varepsilon\left(T^3, (T_2^*)^3\right) = 14\,\%, \quad \varepsilon\left(T^4, (T_2^*)^4\right) = 21\,\%. \end{aligned} \tag{2.4}$$

En este caso, la distribución bootstrap de la muestra 1 aproxima mejor los momentos de la χ^2 que la de la muestra 2.

Hemos resuelto el ejemplo para unos valores fijados de n y B, pero claramente el método bootstrap depende de ellos. Podemos probar a aumentar y disminuir el número de réplicas B, por ejemplo para $B = 10^3$ y 10^7, pero fijando $n = 25$ en la muestra 1. Reutilizando el código anterior y cambiando el valor de B por el deseado, podemos ver los resultados en la Figura 2.3.



(a) Histograma de T^* para $B = 10^3$.

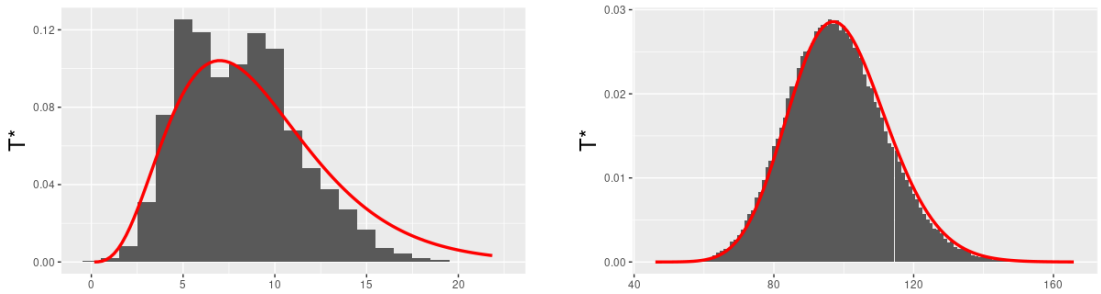
(b) Histograma de T^* para $B = 10^7$.

Figura 2.3: Histogramas de T^* para la muestra 1 variando B.

Como podemos comprobar en la Figura 2.3a, al reducir el número de réplicas bootstrap perdemos parte de la "regularidad" (o "suavidad") que teníamos en la Figura 2.1a (podríamos

recuperarla si aumentamos el tamaño de las celdas del histograma). Sin embargo, en la Figura 2.3b vemos que aumentar el número de réplicas 2 órdenes de magnitud no cambia demasiado el perfil de la antes mencionada Figura 2.1a.

Ahora fijemos $B = 10^5$ y tomemos $n = 10$ y $n = 100$. De nuevo, vamos a reutilizar el código anterior y cambiaremos el valor del tamaño muestral por el elegido. Los nuevos histogramas pueden observarse en la Figura 2.4.



(a) Histograma de T^* para $n = 10$.

(b) Histograma de T^* para $n = 100$.

Figura 2.4: Histogramas de T^* variando n.

Observando la Figura 2.4 podemos concluir que al disminuir n perdemos gran parte del perfil de la χ^2, pero al aumentar n conseguimos una aproximación muy correcta debido a que estamos incluyendo más información en la muestra inicial, en contraposición de lo que obtuvimos en la Figura 2.3b al aumentar B, que además de no mejorar el resultado para 10^5 réplicas suponía un alto coste computacional.

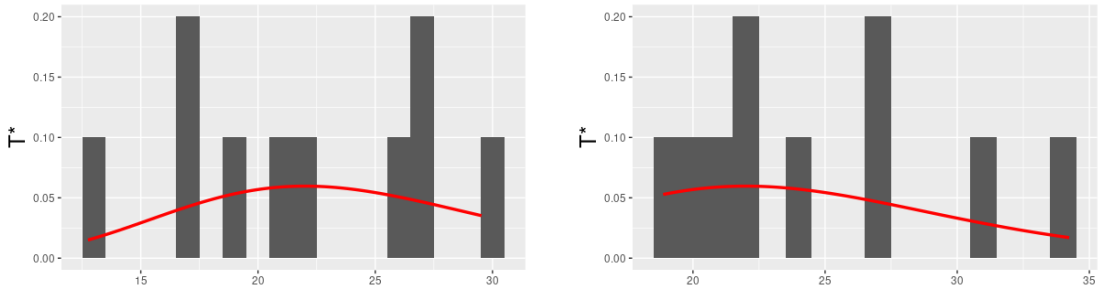
Del ejemplo anterior podemos extraer las siguientes conclusiones:

- El método bootstrap aproxima el comportamiento del método teórico. Aunque lo que más nos interesa es usar el método exacto, no siempre es posible resolver analíticamente, y es aquí donde el método bootstrap adquiere un papel relevante (ver el Ejemplo 2.2).

- La distribución bootstrap que obtenemos del método depende de la muestra inicial y de lo representativa que sea esta muestra con respecto a la población, especialmente si el tamaño muestral es pequeño.

- Para el método bootstrap no hemos usado que la distribución sea normal, tan solo hemos calculado la cuasivarianza y hallado el estadístico T de la Ecuación (2.1).

- El método bootstrap mejora considerablemente si el tamaño muestral es mayor, pero no lo hace tanto cuando el número de réplicas es mayor. No obstante, el método bootstrap requiere de un número mínimo de réplicas bootstrap para funcionar correctamente, como se puede apreciar en la Figura 2.5.



(a) Histograma de T^* para $B = 10$ de la muestra 1.

(b) Histograma de T^* para $B = 10$ de la muestra 2.

Figura 2.5: Histogramas de T^* para $B = 10$.

Sin embargo, en el Ejemplo 2.1 no vemos la ventaja de usar el método bootstrap puesto que conocemos la distribución del estadístico $\widehat{S}_X^2$, y esto se debe a las buenas propiedades de la distribución normal. Veamos el siguiente ejemplo en el que no resulta tan inmediato obtener la distribución del estadístico $\widehat{S}_X^2$.

Ejemplo 2.2. Se han observado 30 datos de una variable aleatoria que sigue una distribución exponencial de parámetro 2. Halla la distribución de la cuasivarianza muestral y los valores de sus momentos poblacionales.

Procederemos como en el Ejemplo 2.1 siguiendo los siguientes pasos (ver el Código A.3):

(1) Creamos una muestra aleatoria de una variable exponencial con los parámetros del enunciado (en R podemos usar la función `rexp`).

(2) Calculamos la cuasivarianza de la muestra.

(3) Construimos una remuestra (en R utilizamos la función `sample`) y calculamos la cuasivarianza de esta remuestra.

(4) Reiteramos el punto anterior un cierto número de veces $B = 10^5$ (con la función `replicate` en R).

(5) Representamos la distribución obtenida.

En la Figura 2.6 podemos ver la distribución que hemos obtenido para la cuasivarianza muestral de una variable exponencial.

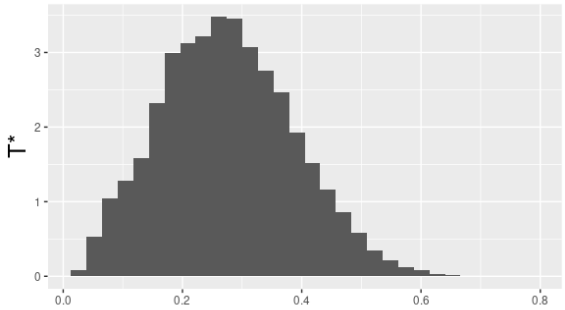


Figura 2.6: Distribución de la cuasivarianza muestral para una exponencial con $\lambda = 2$.

Calculemos los momentos de la distribución bootstrap que hemos hallado atendiendo a los pasos siguientes (ver el Código A.4):

(1) Para la distribución bootstrap aproximamos el momento j-ésimo con la media de $(T^*)^j$ (en R podemos elevar el vector con la distribución bootstrap a j y usar la función `mean`).

(2) Iterar con un bucle *for* en j de 1 hasta 4 el punto anterior.

Los resultados obtenidos fueron los siguientes:

$$\mathrm{E}^*(T^*) = 0.275, \quad \mathrm{E}^*\left((T^*)^2\right) = 0.088, \quad \mathrm{E}^*\left((T^*)^3\right) = 0.031, \quad \mathrm{E}^*\left((T^*)^4\right) = 0.012.$$

Ahora vemos la gran ventaja del método bootstrap: con una muestra podemos reconstruir de manera aproximada el comportamiento de la población aunque desconozcamos la distribución real (o no podamos determinar su comportamiento de manera exacta). Y una vez tenemos la distribución del estimador, podemos aproximar sus momentos para calcular el sesgo o la varianza de dicho estimador.

Capítulo 3

Tipos de bootstrap

En el Capítulo §2 motivamos el estudio del método bootstrap tomando como parámetro de interés la función de distribución F de la m.a.s., es entonces donde aparecía como estimación de F la función de distribución empírica $\widehat{F} = F_n$ y a esta forma de proceder en el método bootstrap la habíamos llamado bootstrap uniforme.

Esta estimación es adecuada cuando no hacemos (o no tenemos) consideraciones adicionales sobre la población de la que proviene la muestra. Sin embargo, si se conocen ciertas propiedades de la población inicial, debemos explotar esta información extra añadiéndola a nuestro remuestreo, y dependiendo de las propiedades que tenga nuestra población usaremos un método u otro. Veamos algunos de los tipos más utilizados.

3.1. Bootstrap paramétrico

Supongamos que la función de distribución poblacional es tal que $F = F_\theta$, con $\theta \in \Theta \subset \mathbb{R}^d$, esto es, sabemos que la función de distribución pertenece a una familia paramétrica. Entonces podemos usar un estimador de θ (como puede ser el estimador máximo verosímil), al que denotaremos por $\widehat{\theta}$, y así usar $F_{\widehat{\theta}}$ para generar las remuestras. Notemos que la distribución del estadístico $\widehat{\theta}$ no es relevante, tan solo nos interesa la estimación para introducirla en $F_{\widehat{\theta}}$.

Resumiendo: si nuestra variable aleatoria X tiene una función de distribución $F = F_\theta$, $\theta \in \Theta \subset \mathbb{R}^d$, y tenemos una m.a.s. $\vec{X} = (X_1, \ldots, X_n)$ a partir de X, podemos obtener un vector de remuestreo:

$$\vec{X}^* = (X_1^*, \ldots, X_n^*)\,,$$

donde las X_i^* tienen por función de distribución a $\widehat{F} = F_{\widehat{\theta}}$ para cada $i \in \{1, \ldots, n\}$. Con todo, podemos proceder como al inicio del Capítulo §2 y obtener el estadístico:

$$T^* = T\left(\vec{X}^*, F_{\widehat{\theta}}\right).$$

Retomemos el Ejemplo 2.1 pero utilizando el nuevo método.

Ejemplo 3.1. Se han registrado 25 observaciones de una variable aleatoria que sigue una distribución normal de media 17.4 (desconocido) y desviación típica 2.1 (desconocido). Halla la distribución de la cuasivarianza muestral y los valores de sus momentos poblacionales utilizando el hecho de que la variable aleatoria es una normal.

En este caso, resolveremos el ejemplo utilizando solamente el método bootstrap (la forma clásica es idéntica a la ya realizada en el capítulo anterior). Lo primero que debemos hacer es hallar un estimador para σ: en esta sección proponíamos usar el estimador máximo verosímil así que vamos a calcularlo. Sea $L(\vec{x}, \mu, \sigma)$ la función de verosimilitud de la realización muestral $\vec{x} = (x_1, \ldots, x_n)$ procedente de una variable aleatoria X normal de media μ y σ desconocidas. Entonces tenemos lo siguiente:

$$L(\vec{x}, \mu, \sigma) = \frac{1}{(2\pi)^{n/2}\,\sigma^n} \exp\left(\frac{-1}{2\sigma^2}\sum_{i=1}^{n}(x_i - \mu)^2\right). \tag{3.1}$$

Ya que la función $\ln(\cdot)$ es creciente en $\mathbb{R}^+$ y deja invariante los extremos relativos de una función[1], podemos transformar la Ecuación (3.1) para obtener:

$$\ln(L(\vec{x}, \mu, \sigma)) = -n\ln(\sigma) - \frac{n}{2}\ln(2\pi) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i - \mu)^2, \tag{3.2}$$

derivando la Ecuación (3.2) con respecto a μ y a σ e igualando a cero obtenemos el máximo de la función verosimilitud:

$$\begin{cases} \dfrac{\partial \ln(L)}{\partial \mu} = \dfrac{1}{\sigma^2}\displaystyle\sum_{i=1}^{n}(x_i - \mu) = 0, \\ \dfrac{\partial \ln(L)}{\partial \sigma} = \dfrac{-n}{\sigma} + \dfrac{1}{\sigma^3}\displaystyle\sum_{i=1}^{n}(x_i - \mu)^2 = 0. \end{cases} \Rightarrow \begin{cases} \widehat{\mu}_{MV} = \dfrac{1}{n}\displaystyle\sum_{i=1}^{n} X_i = \overline{X}, \\ \widehat{\sigma}^2_{MV} = \dfrac{1}{n}\displaystyle\sum_{i=1}^{n}\left(X_i - \overline{X}\right)^2. \end{cases} \tag{3.3}$$

[1]Notemos que la derivada de $\ln(f(x))$ es $f'(x)/f(x)$, que se anula si y solo si $f'(x) = 0$, es decir, los extremos relativos de $f(x)$ son los de $\ln(f(x))$ cuando $f \neq 0$.

Calculemos la matriz hessiana asociada a L para comprobar que efectivamente maximizan la función de verosimilitud:

$$H_L(\mu,\sigma) = \begin{pmatrix} \dfrac{-n}{\sigma^2} & \dfrac{-2}{\sigma^3}\displaystyle\sum_{i=1}^{n}(x_i-\mu) \\ \dfrac{-2}{\sigma^3}\displaystyle\sum_{i=1}^{n}(x_i-\mu) & \dfrac{n}{\sigma^2} - \dfrac{3}{\sigma^4}\displaystyle\sum_{i=1}^{n}(x_i-\mu)^2 \end{pmatrix}.$$

Evaluando en los valores críticos $\widehat{\mu}_{MV}, \widehat{\sigma}_{MV}$ de la Ecuación (3.3) obtenemos:

$$H_L(\widehat{\mu}_{MV},\widehat{\sigma}_{MV}) = \begin{pmatrix} \dfrac{-n}{\widehat{\sigma}^2_{MV}} & 0 \\ 0 & \dfrac{-2n}{\widehat{\sigma}^2_{MV}} \end{pmatrix}, \tag{3.4}$$

ya que $\widehat{\sigma}^2_{MV} > 0$, tenemos que los menores principales de la matriz de la Ecuación (3.4) son $-n/\widehat{\sigma}^2_{MV} < 0$ y $2n^2/\widehat{\sigma}^4_{MV} > 0$ y aplicando el criterio de Sylvester sobre los menores anteriores obtenemos que $(\widehat{\mu}_{MV}, \widehat{\sigma}_{MV})$ es efectivamente un máximo de L.

Ahora que tenemos el estimador máximo verosímil, vamos a usarlo en el método bootstrap: generaremos remuestras aleatorias a partir de una normal de la que estimamos su media y su desviación estándar con la Ecuación (3.3) usando nuestra muestra inicial. A partir de aquí procedemos de igual manera que en el Ejemplo 2.1, obteniendo puntos de una distribución que se asemejará a una χ^2_{n-1} tal y como sabemos por la Ecuación (2.1). Para programar el cálculo podemos seguir los siguientes pasos (ver el Código A.5):

(1) Generamos una muestra aleatoria de una normal con los parámetros del enunciado, que será nuestra muestra original (en R podemos usar el comando `rnorm`). De ella calculamos la estimación máximo verosímil para μ y σ^2, que llamaremos $\widehat{\mu}_{MV}$ y $\widehat{\sigma}^2_{MV}$, utilizando la Ecuación (3.3). En este caso $\widehat{\mu}_{MV} = 17.75$ y $\widehat{\sigma}^2_{MV}(\vec{x}) = 3.95$.

(2) Creamos una remuestra a partir de una normal de media $\widehat{\mu}_{MV}$ y desviación estándar $\widehat{\sigma}_{MV}(\vec{x}) = 1.99$, que es la raíz del valor que habíamos guardado del punto anterior (en R podemos usar `rnorm` para generar la remuestra). De esta remuestra, calculamos su cuasivarianza bootstrap $\widehat{S}^{2(b)}_X$ (con el comando `var` en R) con $b \in \{1,\dots,B\}$ y junto con el valor $\widehat{\sigma}^2_{MV}$ que habíamos calculado en el punto anterior hallamos el valor del estadístico T de la Ecuación (2.1), sustituyendo en dicha ecuación σ^2 por $\widehat{\sigma}^2_{MV}$ y $\widehat{S}^2_X$ por $\widehat{S}^{2(b)}_X$ y almacenamos este valor.

(3) Reiteramos el punto anterior un número B de veces, en este caso $B = 10^5$ (esta reiteración puede hacerse en R con la instrucción `replicate`).

(4) Representamos el vector que hemos obtenido del punto anterior, dibujando encima la densidad de la χ^2_{n-1}.

En la Figura 3.1 podemos observar el histograma de los datos obtenidos.

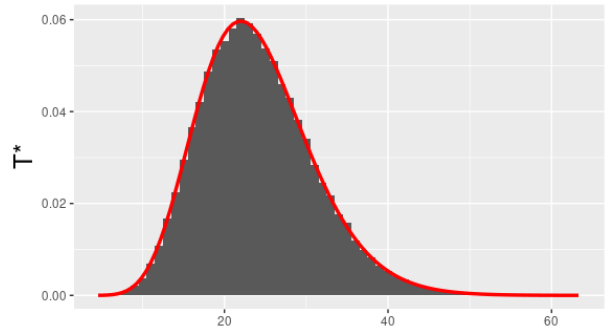


Figura 3.1: Histograma de T^* junto con la densidad de χ^2_{n-1} (bootstrap paramétrico).

Vemos una clara mejoría con respecto a la Figura 2.1a sin necesidad de aumentar el tamaño muestral o el número de réplicas bootstrap, lo que nos sugiere que la hipótesis de normalidad de los datos es muy fuerte. Debemos recordar que en la práctica no siempre sabemos si una variable aleatoria es normal y por ello no siempre podremos utilizar este método.

Para finalizar con este ejemplo, veamos los valores de los momentos poblacionales aproximados mediante el método bootstrap paramétrico. Para calcularlos procederemos de manera idéntica al Ejemplo 2.1 (ver el Código A.6):

(1) Para la distribución bootstrap aproximamos el momento j-ésimo con la media de $(T^*)^j$ (en R podemos elevar el vector con la distribución bootstrap a j y usar la función `mean`), con este valor usamos la Ecuación (2.2) para $k = n - 1$ y $m = j$ y calculamos el error relativo entre ambos resultados.

(2) Iterar con un bucle *for* en j de 1 hasta 4 el punto anterior.

Los resultados fueron los siguientes (comparar con la Ecuación (2.3) de los momentos analíticos):

$$\mathrm{E}^*(T^*) = 24.04, \quad \mathrm{E}^*\left((T^*)^2\right) = 625.87, \quad \mathrm{E}^*\left((T^*)^3\right) = 17550.97, \quad \mathrm{E}^*\left((T^*)^4\right) = 527335.18.$$

Y el error relativo entre el método bootstrap paramétrico y el método analítico ($T \sim \chi^2_{n-1}$ según la Ecuación (2.1)):

$$\varepsilon(T, T^*) = 0.2\,\%,\ \varepsilon\left(T^2, (T^*)^2\right) = 0.3\,\%,\ \varepsilon\left(T^3, (T^*)^3\right) = 0.5\,\%,\ \varepsilon\left(T^4, (T^*)^4\right) = 0.6\,\%. \quad (3.5)$$

En la Ecuación (2.4) teníamos el error relativo que habíamos obtenido para cada muestra y momento con el bootstrap uniforme, y todos ellos superan el error que hemos obtenido por el método paramétrico, que como mucho es de 0.6 %, lo que pone de manifiesto la gran ventaja de usar el bootstrap paramétrico en lugar del bootstrap uniforme.

3.2. Bootstrap simetrizado

La hipótesis que vamos a suponer en este apartado sobre la variable X (que consideraremos continua) que queremos estudiar es que su función de distribución es simétrica respecto de cierto parámetro $c \in \mathbb{X}$, siendo $\mathbb{X} = \text{sop}(X)$ el soporte de la variable X. Entonces tenemos la siguiente igualdad:

$$F(x + c) = 1 - F(c - x), \quad \forall x \in \mathbb{X}. \quad (3.6)$$

Haciendo el cambio $x \to x - c$ en la Ecuación (3.6) encontramos:

$$F(x) = 1 - F(2c - x), \quad \forall x \in \mathbb{X}. \quad (3.7)$$

Ahora que hemos visto las implicaciones de la simetría sobre la función de distribución, vamos a probar diferentes resultados relacionados con una variable aleatoria continua simétrica.

Lema 3.1. *Dada una variable aleatoria continua X con función de distribución simétrica respecto de cierto parámetro $c \in \mathbb{X}$ y con mediana única, se tiene que* $\text{Me}(X) = c$, *donde* $\text{Me}(X)$ *denota la mediana de* X.

Demostración. Partamos de la Ecuación (3.7) con $x = c$:

$$F(c) = 1 - F(c) \Rightarrow F(c) = 1/2,$$

de donde concluimos que $\text{Me}(X) = c$, puesto que $\text{Me}(X)$ es única por hipótesis. □

Lema 3.2. *Dada una variable aleatoria continua X con función de distribución simétrica respecto de cierto parámetro $c \in \mathbb{X}$ y con esperanza finita, se tiene que $\mu = \text{E}(X) = c$, i.e., el centro de simetría se corresponde con la esperanza de la variable.*

Demostración. Como la variable X es continua podemos derivar ambos lados de la Ecuación (3.6) respecto de x y hallar:

$$f(x+c) = f(c-x), \quad \forall x \in \mathbb{X}. \tag{3.8}$$

Por definición, la esperanza de X es:

$$\mu = \mathrm{E}(X) = \int_{\mathbb{R}} x f(x)\,\mathrm{d}x, \tag{3.9}$$

entonces podemos reescribir la Ecuación (3.9) de la siguiente manera[2]:

$$\begin{aligned} \mu &= \int_{\mathbb{R}} (x+c) f(x+c)\,\mathrm{d}x = c\int_{\mathbb{R}} f(x+c)\,\mathrm{d}x + \int_{\mathbb{R}} x f(x+c)\,\mathrm{d}x = \\ &= c\int_{\mathbb{R}} f(x)\,\mathrm{d}x + \int_{\mathbb{R}} x f(x+c)\,\mathrm{d}x = c + \int_{\mathbb{R}} x f(x+c)\,\mathrm{d}x. \end{aligned} \tag{3.10}$$

Pero también podríamos haber reescrito la Ecuación (3.9) como sigue:

$$\mu = \int_{\mathbb{R}} (c-x) f(c-x)\,\mathrm{d}x,$$

y aplicando la Ecuación (3.8):

$$\mu = \int_{\mathbb{R}} (c-x) f(x+c)\,\mathrm{d}x = c\int_{\mathbb{R}} f(x+c)\,\mathrm{d}x - \int_{\mathbb{R}} x f(x+c)\,\mathrm{d}x = c - \int_{\mathbb{R}} x f(x+c)\,\mathrm{d}x. \tag{3.11}$$

Finalmente, sumando la Ecuación (3.10) con la Ecuación (3.11) tenemos:

$$\mu = c,$$

de donde deducimos el resultado. □

Entonces aplicando el Lema 3.2 a una variable aleatoria simétrica y continua tenemos que la función de distribución F es simétrica con respecto a μ.
Para incluir la información de la simetría de la variable respecto a su media (en caso de existir) usaremos el vector $\vec{Y} = (Y_1, \ldots, Y_{2n})$ en el remuestreo, definido por:

$$Y_i = \begin{cases} X_i, & \text{si } i \in \{1, \ldots, n\}, \\ 2\overline{X} - X_{i-n}, & \text{si } i \in \{n+1, \ldots, 2n\}, \end{cases} \tag{3.12}$$

[2]Como la esperanza de la variable X es finita por hipótesis, todas las integrales están bien definidas y son finitas.

donde hemos sustituido la media poblacional μ por el estimador media muestral y n es el tamaño muestral de la muestra original $\vec{X} = (X_1, \ldots, X_n)$. El estadístico que vamos a usar para obtener remuestras es la función de distribución empírica $\widehat{F} = F_n^{sim}$ de este vector $\vec{Y}$:

$$F_n^{sim}(x) = \frac{1}{2n} \sum_{i=1}^{2n} \mathbb{1}(Y_i \leq x). \tag{3.13}$$

Podemos reescribir la Ecuación (3.13) separando el sumatorio en dos partes:

$$\begin{aligned} F_n^{sim}(x) &= \frac{1}{2} \left(\frac{1}{n} \sum_{i=1}^{n} \mathbb{1}(X_i \leq x) + \frac{1}{n} \sum_{i=n+1}^{2n} \mathbb{1}\left(2\overline{X} - X_{i-n} \leq x\right) \right) = \\ &= \frac{1}{2} \left(F_n(x) + \frac{1}{n} \sum_{j=1}^{n} \mathbb{1}\left(2\overline{X} - X_j \leq x\right) \right) = \\ &= \frac{1}{2} \left(F_n(x) + \frac{1}{n} \sum_{j=1}^{n} \mathbb{1}\left(X_j \geq 2\overline{X} - x\right) \right) = \\ &= \frac{1}{2} \left(F_n(x) + 1 - F_n\left(2\overline{X} - x\right) \right), \end{aligned}$$

donde en la última igualdad hemos usado que X es una variable aleatoria continua.

Podemos seguir estudiando las propiedades de F_n^{sim}, como por ejemplo su media y momentos centrales.

Propiedades 3.3. *Sean una variable aleatoria continua X y simétrica con esperanza finita, una muestra aleatoria simple $\vec{X} = (X_1, \ldots, X_n)$ a partir de X y el vector $\vec{Y} = (Y_1, \ldots, Y_{2n})$ definido como en la Ecuación (3.12). Una variable aleatoria que tenga por función de distribución la función de distribución empírica simetrizada F_n^{sim} verifica:*

(i) Su esperanza es la media muestral $\overline{X}$.

(ii) Sus momentos centrales de orden impar son nulos.

(iii) Sus momentos centrales de orden par coinciden con los momentos centrales de orden par de la muestra.

Demostración. Llamemos Z a la variable con función de distribución F_n^{sim}.

(i) La esperanza de Z se calcula usando las definiciones de la Ecuación (3.13) y de la

Ecuación (3.12):

$$\mathrm{E}(Z) = \sum_{i=1}^{2n} Y_i\, P(Z = Y_i) = \frac{1}{2n}\sum_{i=1}^{2n} Y_i = \frac{1}{2n}\left(\sum_{i=1}^{n} X_i + \sum_{i=n+1}^{2n}(2\overline{X} - X_{i-n})\right) =$$
$$= \frac{1}{2n}\left(\sum_{i=1}^{n} X_i + \sum_{i=1}^{n}(2\overline{X} - X_i)\right) = \overline{X},$$

de donde obtenemos el resultado.

(ii) Sea $k \in \mathbb{N}$, vamos a hallar la esperanza de $(Z - \mathrm{E}(Z))^{2k-1}$ usando el apartado anterior:

$$\begin{aligned}
\mathrm{E}\left((Z - \mathrm{E}(Z))^{2k-1}\right) &= \frac{1}{2n}\sum_{i=1}^{2n}(Y_i - \overline{X})^{2k-1} = \\
&= \frac{1}{2n}\left(\sum_{i=1}^{n}(X_i - \overline{X})^{2k-1} + \sum_{i=n+1}^{2n}(\overline{X} - X_{i-n})^{2k-1}\right) = \\
&= \frac{1}{2n}\left(\sum_{i=1}^{n}(X_i - \overline{X})^{2k-1} + \sum_{i=1}^{n}(\overline{X} - X_i)^{2k-1}\right) = \\
&= \frac{1}{2n}\left(\sum_{i=1}^{n}(X_i - \overline{X})^{2k-1} - \sum_{i=1}^{n}(X_i - \overline{X})^{2k-1}\right) = \\
&= 0,
\end{aligned}$$

y hemos llegado al resultado.

(iii) Tomemos otra vez $k \in \mathbb{N}$, directamente:

$$\begin{aligned}
\mathrm{E}\left((Z - \mathrm{E}(Z))^{2k}\right) &= \frac{1}{2n}\sum_{i=1}^{2n}(Y_i - \overline{X})^{2k} = \\
&= \frac{1}{2n}\left(\sum_{i=1}^{n}(X_i - \overline{X})^{2k} + \sum_{i=n+1}^{2n}(\overline{X} - X_{i-n})^{2k}\right) = \\
&= \frac{1}{2n}\left(\sum_{i=1}^{n}(X_i - \overline{X})^{2k} + \sum_{i=1}^{n}(\overline{X} - X_i)^{2k}\right) = \\
&= \frac{1}{2n}\left(\sum_{i=1}^{n}(X_i - \overline{X})^{2k} + \sum_{i=1}^{n}(X_i - \overline{X})^{2k}\right) = \\
&= \frac{1}{n}\sum_{i=1}^{n}(X_i - \overline{X})^{2k},
\end{aligned}$$

este último resultado es el momento central de orden $2k$ de la muestra, por lo que concluimos la demostración.

□

Las propiedades anteriores las manifestará nuestro remuestreo bootstrap, ya que es a partir de F_n^{sim} de donde obtendremos las remuestras simétricas.

Ahora que hemos visto las propiedades básicas de una variable aleatoria simétrica y de F_n^{sim} vamos a recuperar el Ejemplo 2.1 admitiendo que los datos proceden de una variable aleatoria simétrica, como es la distribución normal.

Ejemplo 3.2. Se han registrado 25 observaciones de una variable aleatoria que sigue una distribución normal de media 17.4 (desconocido) y desviación típica 2.1 (desconocido). Halla la distribución de la cuasivarianza muestral y los valores de sus momentos poblacionales utilizando el hecho de que la variable aleatoria es simétrica.

Vamos a aplicar el nuevo método siguiendo los siguientes pasos para programar los cálculos (ver el Código A.7):

(1) Generamos una muestra aleatoria a partir de una normal de media 17.4 y desviación típica 2.1 (en R podemos usar la función `rnorm`). De ella, guardamos su cuasivarianza $\widehat{S}_X^2$.

(2) A partir de la muestra anterior, creamos una muestra simetrizada: si la muestra inicial era el vector $\vec{X}$ de tamaño n, fabricamos un nuevo vector $\vec{Y}$ de tamaño $2n$ cuyas n primeras entradas sean los valores de $\vec{X}$ y las n siguientes sean $2\overline{X} - X_{i-n}$ con $i \in \{n+1, \ldots, 2n\}$. Esto es, siguiendo la Ecuación (3.12).

(3) Generamos una remuestra de tamaño n de la muestra simetrizada (en R utilizamos la función `sample` con reemplazamiento). De esta remuestra calculamos su cuasivarianza bootstrap $\widehat{S}_X^{2(b)}$ (con $b \in \{1, \ldots, B\}$) y en el estadístico T de la Ecuación (2.1) sustituimos σ^2 por $\widehat{S}_X^2$ y $\widehat{S}_X^2$ por $\widehat{S}_X^{2(b)}$, y guardamos este valor.

(4) Repetimos el punto anterior un número B de veces, en nuestro caso $B = 10^5$ (en R podemos usar la función `replicate`).

(5) Representamos en un histograma el vector que obtenemos del punto anterior, dibujando encima la densidad de χ^2_{n-1}.

En la Figura 3.2 podemos observar los resultados obtenidos.

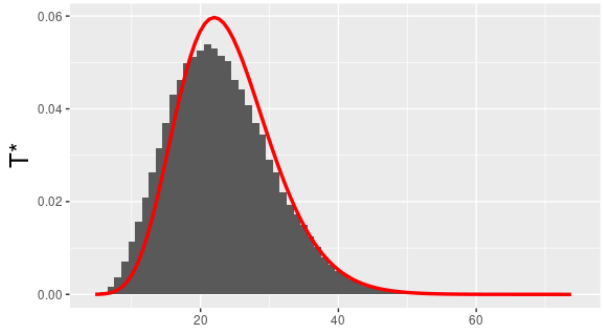


Figura 3.2: Histograma de T^* junto con la densidad de χ^2_{n-1} (bootstrap simétrico).

Respecto de la Figura 2.1a no observamos una gran mejoría al haber usado la simetría de la distribución normal, aunque el máximo de la distribución se ha suavizado con respecto a los resultados del bootstrap uniforme.
Veamos a continuación los valores de los primeros cuatro momentos de la distribución bootstrap. Para el cálculo de estos resultados hemos seguido los siguientes pasos (ver el Código A.8):

(1) Aproximamos el j-ésimo momento de la distribución bootstrap con la media de $(T^*)^j$ (en R elevamos el vector que contiene los puntos de la distribución bootstrap a j y usamos la función `mean`), y con este valor y el de la Ecuación (2.2) para $k = n - 1$ y $m = j$ calculamos el error relativo entre los dos.

(2) Iteramos el punto anterior con un bucle *for* de $j = 1$ hasta 4.

Podemos observar los resultados a continuación (comparar con la Ecuación (2.3) de los momentos analíticos):

$$\mathrm{E}^*(T^*) = 23.06, \quad \mathrm{E}^*\left((T^*)^2\right) = 587.75, \quad \mathrm{E}^*\left((T^*)^3\right) = 16349.07, \quad \mathrm{E}^*\left((T^*)^4\right) = 491354.17.$$

Y el error relativo entre el método bootstrap paramétrico y el método analítico (siendo $T \sim \chi^2_{n-1}$ de acuerdo con la Ecuación (2.1)):

$$\varepsilon(T, T^*) = 4\,\%, \quad \varepsilon\left(T^2, (T^*)^2\right) = 6\,\%, \quad \varepsilon\left(T^3, (T^*)^3\right) = 6\,\%, \quad \varepsilon\left(T^4, (T^*)^4\right) = 6\,\%. \quad (3.14)$$

Los errores de la Ecuación (3.14) mejoran también a los de la Ecuación (2.4) como sucedía en el caso del bootstrap paramétrico, de hecho el mayor error relativo para el bootstrap

simetrizado es de 6 %, mientras que para el bootstrap uniforme teníamos un error relativo máximo del 9 % (para la muestra 1). Sin embargo, en la Ecuación (3.5) que obtuvimos con el método paramétrico el error relativo era de un orden de magnitud menor que en el caso simetrizado, dejándonos claro que el método bootstrap paramétrico es el que mejor aproxima el comportamiento real. Este comportamiento era esperado, pues en el Ejemplo 3.1 del método paramétrico asumimos simetría (aunque no lo hayamos especificado) al tomar muestras a partir de una variable normal (que es simétrica). Es decir, el método paramétrico incluye al método simetrizado porque las remuestras que se escogen provienen de la misma familia de distribuciones y por lo tanto serán simétricas también. A todo lo anterior se le añade que asumir que la variable provenga de una distribución normal es una suposición mucho más fuerte que asumir que la variable sea solamente simétrica.

3.3. Bootstrap suavizado

En el bootstrap suavizado la hipótesis que suponemos sobre la variable de estudio es que es continua, por lo tanto tenemos la siguiente relación entre su función de distribución y densidad: $F' = f$. Es por esto que vamos a estimar la función de densidad de forma no paramétrica.

En este contexto vamos a usar una muestra $\vec{X} = (X_1, \ldots, X_n)$ a partir de la variable continua X y el estimador tipo núcleo que propusieron Parzen [9] y Rosenblatt [12], que viene dado por:

$$\widehat{f}_h(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right), \tag{3.15}$$

siendo $h > 0$ un parámetro de suavizado que llamaremos tamaño de la ventana y K es la función núcleo que usualmente es una densidad simétrica en torno al cero, esto es:

$$K(u) \geq 0, \ \forall u \in \mathbb{R}, \quad \int_{\mathbb{R}} K(u)\,\mathrm{d}u = 1, \quad K(-u) = K(u), \ \forall u \in \mathbb{R}.$$

Por ejemplo, esta densidad K puede ser la densidad de una uniforme $\mathcal{U}(-1, 1)$ y entonces el núcleo utilizado sería rectangular. Si fuese la densidad de una normal $\mathcal{N}(0, 1)$ el núcleo sería de tipo gaussiano. Existen diversos tipos de núcleos, pero en el estudio del método bootstrap suavizado nos restringiremos a los dos anteriores y al núcleo de tipo Epanechnikov, nombrado así por V. A. Epanechnikov [6]. Como veremos más adelante, la elección del núcleo no afecta de manera significativa al bootstrap suavizado. No obstante, el tamaño de ventana h sí juega

un papel relevante en cuanto a la sensibilidad de los resultados del método suavizado, como comprobaremos en los apartados siguientes.

Antes de plantear el esquema que seguiremos a la hora de realizar el método bootstrap suavizado, vamos a ver la función de distribución que está asociada al estimador no paramétrico de la función de densidad definido en la Ecuación (3.15):

$$\begin{aligned} \widehat{F}_h(x) &= \int_{-\infty}^{x} \widehat{f}_h(y)\,\mathrm{d}y = \frac{1}{nh}\sum_{i=1}^{n}\int_{-\infty}^{x} K\left(\frac{y-X_i}{h}\right)\,\mathrm{d}y = \\ &= \frac{1}{n}\sum_{i=1}^{n}\int_{-\infty}^{\frac{x-X_i}{h}} K(z)\,\mathrm{d}z = \frac{1}{n}\sum_{i=1}^{n}\mathbb{K}\left(\frac{x-X_i}{h}\right), \end{aligned} \tag{3.16}$$

siendo $\mathbb{K}(u)$ la función de distribución del núcleo K.

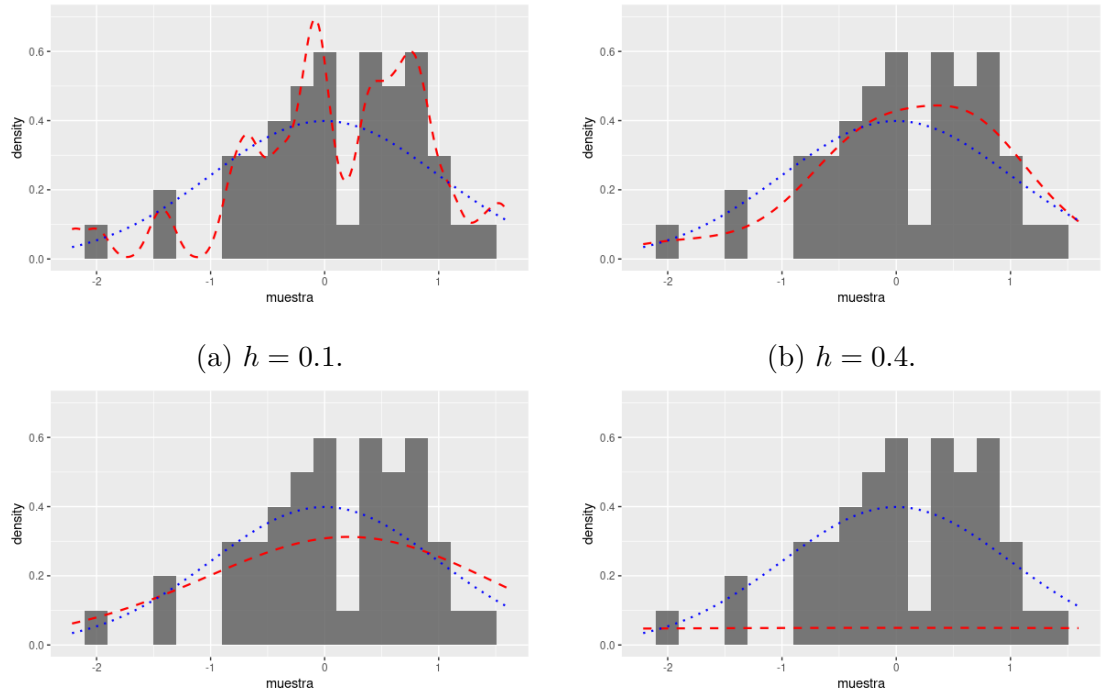
Para ilustrar un poco la estimación no paramétrica de la función de densidad veamos el siguiente ejemplo. Consideremos una muestra aleatoria simple de una variable aleatoria continua, por ejemplo una normal de media 0 y desviación típica 1. Vamos a estimar su función de densidad con un núcleo gaussiano y un ancho de ventana distinto, por ejemplo vamos a probar con los siguientes anchos de ventana $h \in \{0.1, 0.4, 1, 8\}$. Para programar el resultado seguimos los pasos siguientes (ver el Código A.9):

(1) Creamos una muestra aleatoria a partir de una normal con $n = 50$ valores. En R usaremos `rnorm`.

(2) Creamos un vector con las ventanas anteriores.

(3) Realizamos un bucle *for* que itere los anchos del punto (2) y estimamos la función de densidad de la muestra para el ancho de banda que estemos iterando. En R usaremos el comando `geom_density` del paquete `ggplot2`. Seguidamente, hacemos un histograma de la muestra y la densidad estimada.

Podemos observar los resultados en la Figura 3.3.



(a) $h = 0.1$.

(b) $h = 0.4$.

(c) $h = 1$.

(d) $h = 8$.

Figura 3.3: Histogramas de la muestra y sus estimaciones de la densidad (en rojo) para distintos valores de h, junto con la densidad de $\mathcal{N}(0, 1)$ (en azul).

Podemos comprobar en la Figura 3.3d que aumentar el tamaño de h hace que la distribución normal que usamos como núcleo se aplane, oscureciendo parte de la estructura de la que provienen los datos. Sin embargo, reducir mucho el parámetro de suavizado hace que en la estimación de la función de densidad se tengan en cuenta demasiados datos que no son tan relevantes (ver Figura 3.3a).

Ahora estamos en condiciones de aplicar el método bootstrap suavizado: a partir de la m.a.s. $\vec{X} = (X_1, \ldots, X_n)$ de la variable continua X y una ventana h calculamos el estimador tipo núcleo de la función de densidad $\widehat{f}_h$, y es a partir de la densidad que hemos estimado de donde obtenemos una remuestra bootstrap $\vec{X}^* = (X_1^*, \ldots, X_n^*)$. Con esta última calcularíamos el estadístico $T^* = T\left(\vec{X}^*, \widehat{F}_h\right)$ como hicimos al comienzo del Capítulo §2.
Para llevar a cabo el remuestreo a partir de $\widehat{f}_h$ haremos lo siguiente: para cada $i \in \{1, \ldots, n\}$ arrojamos $U_i \sim \mathcal{U}(0, 1)$ y V_i con densidad dada por el núcleo K, y hacemos $X_i^* = X_{\lfloor nU_i \rfloor + 1} + hV_i$. La equivalencia entre estas dos maneras de generar las remuestras está dada por lo siguiente: sean X^* una variable aleatoria discreta que toma valores X_i^* con $i \in \{1, \ldots, n\}$

con equiprobabilidad, $U \sim \mathcal{U}(0,1)$, $I = \lfloor nU \rfloor + 1$ (notemos que I es una variable aleatoria uniforme discreta que toma valores $\{1, \ldots, n\}$) y V una variable aleatoria independiente de U que tiene por función de densidad a K, entonces:

$$\begin{aligned} P(X^* \leq x) &= P\left(\bigcup_{i=1}^{n}(\{X_i^* \leq x|_{I=i}\} \cap \{I = i\})\right) = \sum_{i=1}^{n} P(\{X_i^* \leq x|_{I=i}\} \cap \{I = i\}) = \\ &= \sum_{i=1}^{n} P\left(X^* \leq x|_{I=i}\right) P(I = i) = \frac{1}{n}\sum_{i=1}^{n} P\left(X_i + hV \leq x|_{I=i}\right) = \\ &= \frac{1}{n}\sum_{i=1}^{n} P\left(V \leq \left.\frac{x - X_i}{h}\right|_{X_i}\right) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{K}\left(\frac{x - X_i}{h}\right), \end{aligned}$$

que es precisamente la función de distribución $\widehat{F}_h$ que hallamos en la Ecuación (3.16). Es decir, remuestrear de la densidad $\widehat{f}_h$ es equivalente a remuestrear de una variable $X_i^* = X_{\lfloor nU_i \rfloor + 1} + hV_i$. Podemos entender este último remuestreo como una perturbación al bootstrap uniforme, cuya magnitud está regida por el tamaño de la ventana h. De hecho, en el límite $h = 0$ encontramos el bootstrap uniforme.

Aunque haremos más adelante un estudio de simulación sobre el parámetro de suavizado h, la determinación de dicho parámetro es un problema por sí mismo. De hecho, existen numerosas formas de obtener este ancho de ventana, como podemos ver en [14] que utiliza una regla práctica (de Silverman) para aproximar el valor de h, que programas estadísticos como R (que es el que utilizaremos a la hora de obtener resultados) utilizan como ancho de ventana predeterminado. La expresión de este parámetro es:

$$h_{\text{opt}} = 0.9\, n^{-1/5}\, \text{mín}\left\{\widehat{\sigma}, \frac{IQR}{1.34}\right\}, \tag{3.17}$$

siendo IQR el rango intercuartílico definido como la diferencia del tercer cuartil con el primero, i.e., $Q_3 - Q_1$, $\widehat{\sigma}$ la desviación típica de la muestra y n el tamaño muestral.

Para poner en práctica lo que acabamos de ver consideremos de nuevo el Ejemplo 2.1.

Ejemplo 3.3. Se han registrado 25 observaciones de una variable aleatoria que sigue una distribución normal de media 17.4 (desconocido) y desviación típica 2.1 (desconocido). Halla la distribución de la cuasivarianza muestral y los valores de sus momentos poblacionales usando que la variable aleatoria es continua.

Para llevar a cabo el ejercicio, seguiremos los siguientes pasos (ver el Código A.10):

(1) Generamos la muestra inicial usando los parámetros del enunciado, i.e., una normal de media 17.4 y desviación típica 2.1 (en R podemos usar la función `rnorm`). De esta muestra calculamos su cuasivarianza $\widehat{S}_X^2$.

(2) Ahora vamos a estimar la función de densidad con la Ecuación (3.15) escogiendo K un núcleo gaussiano y h dada por la Ecuación (3.17) (en R podemos usar la función `density` aplicada a la muestra inicial). Además, guardamos en una variable el valor de h.

(3) Generamos una remuestra de tamaño n a partir de la muestra original y la perturbamos sumando el término hV_i con $V_i \sim \mathcal{N}(0,1)$ para $i \in \{1,\dots,n\}$ (recordemos que en el paso anterior habíamos elegido un núcleo gaussiano). Este paso se puede realizar en R usando la función `sample` con reemplazamiento y generando muestras aleatorias de una normal con el comando `rnorm`. De esta remuestra hallamos su cuasivarianza bootstrap $\widehat{S}_X^{2(b)}$ (con $b \in \{1,\dots,B\}$) y calculamos el valor del estadístico T de la Ecuación (2.1) sustituyendo σ^2 por $\widehat{S}_X^2$ y $\widehat{S}_X^2$ por $\widehat{S}_X^{2(b)}$. Almacenamos el resultado de esta operación.

(4) Repetimos el paso anterior un número B de veces, en nuestro caso $B = 10^5$ (en R tenemos la función `replicate`).

(5) Representamos en un histograma los resultados obtenidos.

Los resultados pueden observarse en la Figura 3.4:

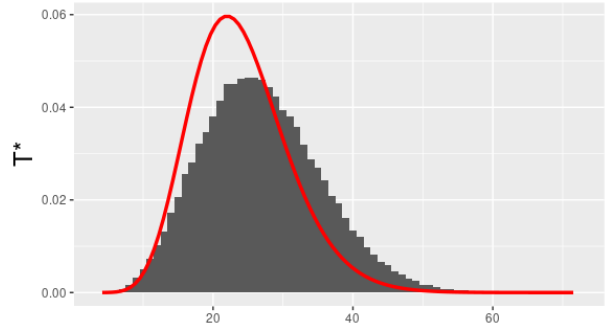


Figura 3.4: Histograma de T^* junto con la densidad de χ^2_{n-1} (bootstrap suavizado).

La distribución de la Figura 3.4 se ajusta peor que las distribuciones que obtuvimos en la Figura 3.1 y en la Figura 3.2 que se corresponden con los métodos paramétrico y simétrico

respectivamente, al menos para esta muestra y tamaño muestral. Con respecto a la distribución que hallamos para el método uniforme, que vemos en la Figura 2.1a, parece que el método suavizado no ha mejorado. Más adelante veremos que podemos subsanar este comportamiento escogiendo un parámetro de suavizado más fino.

Veamos seguidamente los valores de los cuatro primeros momentos de la distribución bootstrap. Hemos seguido los siguientes pasos para obtener los resultados (ver el Código A.11):

(1) Aproximamos el j-ésimo momento de la distribución bootstrap con la media de $(T^*)^j$ (en R elevamos el vector a que contiene los puntos de la distribución bootstrap a j y aplicamos el comando `mean`), con este valor y el de la Ecuación (2.2) para $k = n-1$ y $m = j$ calculamos el error relativo entre ambos.

(2) Iteramos el punto anterior con un bucle *for* desde $j = 1$ hasta 4.

Estos fueron los resultados para los momentos de la distribución bootstrap:

$$\mathrm{E}^*(T^*) = 26.89, \quad \mathrm{E}^*\left((T^*)^2\right) = 794.90, \quad \mathrm{E}^*\left((T^*)^3\right) = 25465.64, \quad \mathrm{E}^*\left((T^*)^4\right) = 874771.69.$$

Y a continuación los errores relativos (con $T \sim \chi^2_{n-1}$ según la Ecuación (2.1)):

$$\varepsilon(T, T^*) = 12\,\%, \quad \varepsilon\left(T^2, (T^*)^2\right) = 27\,\%, \quad \varepsilon\left(T^3, (T^*)^3\right) = 46\,\%, \quad \varepsilon\left(T^4, (T^*)^4\right) = 67\,\%. \tag{3.18}$$

Los errores que obtuvimos con el método suavizado han sido los más altos de todos los métodos, incluso más que el uniforme, como podíamos intuir de la comparación entre los histogramas que habíamos hecho. Esto motiva el estudio de la sensibilidad del parámetro h que haremos en la siguiente sección.

3.3.1. Estudio de la sensibilidad del parámetro de suavizado

Vamos a realizar un pequeño estudio de simulación en el que vamos a variar el ancho de ventana de la Ecuación (3.17). En particular, consideraremos 100 muestras aleatorias que provienen de una distribución normal de media 17.4 y desviación típica 2.1 (es decir, como en el Ejemplo 3.3) y realizaremos un bootstrap suavizado para cada una de ellas cambiando el valor de h de la Ecuación (3.17) por λh, donde $\lambda \in \left\{1, \frac{1}{2}, \frac{1}{3}, \frac{1}{4}\right\}$, manteniendo el núcleo gaussiano.

Para comparar los resultados usaremos los momentos de cada distribución bootstrap y calcularemos sus errores relativos. Ya que tenemos 100 muestras, para cada método y cada

muestra hallaremos su media y el error relativo de esta última con respecto del momento analítico que tenemos en la Ecuación (2.2). Una vez hemos hecho esto, obtendremos el promedio de los errores relativos para cada método haciendo la media de los 100 errores relativos que teníamos para cada valor de h distinto. Para programar el cálculo hemos seguido los siguientes pasos (ver el Código A.12):

(1) Como vamos a simular 100 muestras de tamaño 25, creamos una matriz 25×100 (en R usamos el comando `matrix`) en la que guardaremos nuestras muestras. En la i-ésima columna de la matriz colocamos una muestra aleatoria a partir de una normal de media 17.4 y desviación típica 2.1 (en R podemos hacer esto con `rnorm`). Iterando con un bucle *for* desde $i = 1$ hasta 100 obtenemos la matriz buscada.

(2) Hallamos la cuasivarianza de cada muestra, es decir, guardamos en un vector la cuasivarianza de las columnas de la matriz de muestras (en R aplicamos el comando `apply` por columnas a la matriz de muestras junto con la cuasivarianza, cuyo comando es `var`). Obtendremos así un vector de cuasivarianzas de longitud 100.

(3) Creamos un vector de tamaño 100 que contenga los anchos de ventana h de cada muestra, aplicando la Ecuación (3.17). Para hacer este paso en R vamos a crear un vector que tenga por componentes la salida de la función `density` aplicada a cada muestra, y para llevar a cabo lo anterior usamos el comando `apply` sobre las columnas de la matriz de muestras junto con la función `density`. Como lo que queríamos era un vector con los parámetros de suavizado, vamos a acceder a la i-ésima entrada del vector que tiene las densidades estimadas y extraemos el ancho de ventana con la expresión `$bw`, para finalizar iteramos con un bucle *for* desde $i = 1$ hasta 100 guardando dichos anchos de ventana.

(4) Creamos cuatro matrices (una por cada modificación del parámetro h) de dimensiones $B \times 100$, donde B es el número de réplicas bootstrap (en nuestro caso $B = 10^5$). Vamos a realizar el método bootstrap suavizado para la i-ésima muestra:

   (a) Generamos una remuestra a partir de la i-ésima muestra, en R podemos hacer esto con el comando `sample` con reemplazamiento. Vamos a perturbar esta remuestra de cuatro formas distintas: sumando a la remuestra la cantidad $\lambda h V$,

con $\lambda \in \{1, 1/2, 1/3, 1/4\}$ y $V \sim \mathcal{N}(0,1)$, pues suponíamos que el núcleo era gaussiano. Para hacerlo en R, rescatamos el vector en el que habíamos guardado los valores de h para cada muestra, accedemos a su i-ésima componente y la multiplicamos por el factor λ correspondiente. Para cada una de estas remuestras (serán cuatro) calculamos su cuasivarianza $\widehat{S}_X^{2(b)}$ (en R con el comando `var`) con $b \in \{1, \ldots, B\}$. Con la i-ésima componente del vector que contenía las cuasivarianzas de las muestras, que vamos a llamar $\widehat{S}_X^2$, calculamos el estadístico T de la Ecuación (2.1) cambiando σ^2 por $\widehat{S}_X^2$ y $\widehat{S}_X^2$ por $\widehat{S}_X^{2(b)}$ (para cada modificación de h).

(b) Repetimos el paso (a) B veces, en R podemos utilizar la función `replicate`.

(c) Guardamos los resultados de cada modificación en las columnas de las matrices que habíamos creado al principio del punto (4).

(5) Iteramos con un bucle *for* el punto (4) desde $i = 1$ hasta 100.

(6) Llegados aquí, tenemos la distribución bootstrap de cada muestra y de cada modificación del parámetro h, ahora vamos a calcular los errores relativos como sigue:

(a) Hallamos el j-ésimo momento de la i-ésima muestra de una de las cuatro modificaciones del método como la media de la i-ésima columna de la matriz que contenga las distribuciones bootstrap elevada a j. En R usaremos la función `apply` sobre las columnas de la distribución bootstrap que corresponda al método junto con la función `mean`. Así, obtendremos un vector de longitud 100 con los momentos de orden j de cada distribución bootstrap asociada a cada muestra. Para cada entrada de dicho vector calculamos el error relativo con el resultado de la Ecuación (2.2) para $k = n - 1$ y $m = j$. Por último, hallamos la media de los errores relativos, en R con el comando `mean`.

(b) Repetimos el apartado (a) para cada modificación del parámetro h, en total cuatro veces.

(7) Iteramos con un bucle *for* desde $j = 1$ hasta 4 (solo queremos los primeros cuatro momentos).

Los resultados para la media de los errores relativos fueron los siguientes:

Errores relativos (%)	Orden 1	Orden 2	Orden 3	Orden 4
Factor 1 (∗)	15	31	49	67
Factor 1/2 (∗)	1	1	4	8
Factor 1/3	2	5	9	15
Factor 1/4	3	7	11	17

Tabla 3.1: Errores relativos para cada modificación de h y para cada orden.

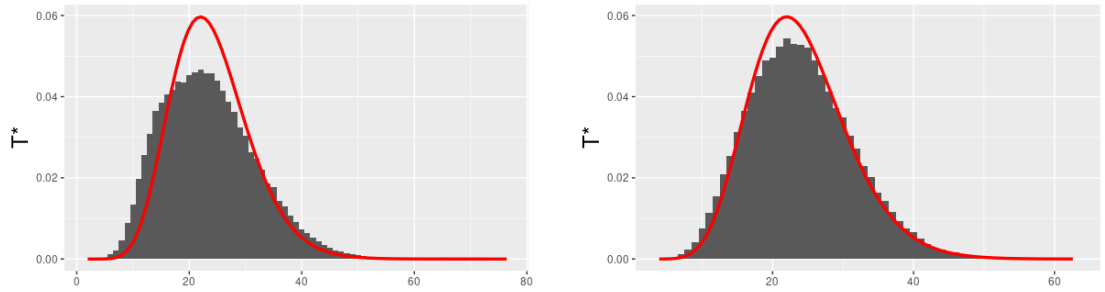
En la Tabla 3.1 podemos ver en rojo la modificación con mayor error relativo en todos los órdenes y en verde aquella modificación con menor error relativo. Los errores relativos que obteníamos en la Ecuación (3.18) eran similares a los que hemos encontrado en la primera fila de la Tabla 3.1, que se corresponde con el método bootstrap sin modificar (factor 1). Como también podemos observar en la Tabla 3.1, disminuir el parámetro h todo lo posible no reduce el error relativo, de hecho, para los cuatro momentos de la distribución bootstrap, el factor que más reduce el error relativo es 1/2, si continuamos disminuyendo el valor de h el error relativo no decrece.

Para acabar el estudio, vamos a escoger dos de las distribuciones que hemos aproximado por el método bootstrap suavizado modificado por 1/2 y las representaremos. Estas dos distribuciones serán la distribución con mayor error relativo respecto del primer momento poblacional, y la distribución con menor error relativo. Seguiremos los siguientes pasos (ver el Código A.13):

(1) A la distribución bootstrap modificada por 1/2, le aplicamos la media por columnas (en R con los comandos `apply` y `mean`), y del resultado hallamos el error relativo. Así, podemos hallar las muestras que maximizan y minimizan el error, en R usaremos las funciones `which.max` y `which.min`.

(2) Representamos cada distribución, dibujando encima la densidad de χ^2_{n-1}.

En la Figura 3.5 observamos las distribuciones obtenidas.



(a) T^* que maximiza el error relativo ($h/2$). (b) T^* que minimiza el error relativo ($h/2$).

Figura 3.5: Distribuciones T^* con error relativo máximo y mínimo.

Debemos tener en cuenta que este estudio es válido tan solo para la distribución que estamos considerando, una normal $\mathcal{N}(17.4, 2.1)$, en principio para otras distribuciones no tiene por qué verificarse que el parámetro de suavizado que mejores resultados produce sea $h/2$. Debemos subrayar que el problema de la elección de este parámetro de suavizado es complejo y un análisis más extenso se escapa de los objetivos de este trabajo.

3.3.2. Estudio sobre el efecto de la elección del núcleo

Para terminar con el método suavizado, estudiaremos la dependencia de los resultados con el núcleo (o kernel) elegido. En este caso, tendremos en cuenta tres núcleos: gaussiano, rectangular y de Epanechnikov. Para el núcleo gaussiano ya hemos visto que a la hora de realizar el método bootstrap basta perturbar la remuestra sumando un término hV, con $V \sim \mathcal{N}(0,1)$, mientras que para el núcleo rectangular tomaremos $V \sim \mathcal{U}(-1,1)$. Por último, para un núcleo del tipo Epanechnikov tenemos que la función de densidad es (de acuerdo con [6]):

$$K_E(u) = \begin{cases} \dfrac{3}{4\sqrt{5}} - \dfrac{3u^2}{20\sqrt{5}}, & \text{si } |u| \leq \sqrt{5}, \\ 0, & \text{si } |u| > \sqrt{5}. \end{cases} \tag{3.19}$$

La función definida en la Ecuación (3.19) verifica $K_E(u) \geq 0$ y que es simétrica respecto del cero. Veamos que su integral en $\mathbb{R}$ es igual a 1:

$$\int_{\mathbb{R}} K_E(u)\,\mathrm{d}u = \int_{-\sqrt{5}}^{\sqrt{5}} \left(\frac{3}{4\sqrt{5}} - \frac{3u^2}{20\sqrt{5}} \right) \mathrm{d}u = \frac{3}{2} - \frac{1}{2} = 1.$$

Deducimos que K_E es, en efecto, una función de densidad simétrica respecto del cero y por lo tanto es un núcleo. Sin embargo, no es este núcleo el que vamos a usar y lo modificaremos

como sigue:

$$K_E^*(u) = \sqrt{5}K_E(\sqrt{5}u) = \begin{cases} \frac{3}{4}\left(1-u^2\right), & \text{si } |u| \leq 1, \\ 0, & \text{si } |u| > 1, \end{cases} \tag{3.20}$$

pues es la forma más común de utilizar el núcleo de Epanechnikov (de hecho, R usa esta definición). Vamos a probar que la función definida en la Ecuación (3.20) es una función de tipo núcleo. Para ello, enunciaremos un resultado más general.

Lema 3.4. *Sea K una función de tipo núcleo, i.e., una función de densidad simétrica en torno al cero, entonces la función definida por $K^*(u) = \lambda K(\lambda u)$, con $\lambda > 0$, también es una función de tipo núcleo.*

Demostración. Probaremos que K^* es una función no negativa, simétrica y cuya integral en $\mathbb{R}$ es igual a 1.

- $K^*(u) \geq 0$, $\forall u \in \mathbb{R}$: sea $u \in \mathbb{R}$ cualquiera pero fijo, tenemos que:

  $$K^*(u) = \lambda K(\lambda u) = \lambda K(v) \geq 0,$$

  ya que $K(v) \geq 0$, $\forall v \in \mathbb{R}$ y $\lambda > 0$.

- K^* simétrica respecto del cero: sea $u \in \mathbb{R}$ cualquiera pero fijo, se cumple lo siguiente:

  $$K^*(u) = \lambda K(\lambda u) = \lambda K(-\lambda u) = K^*(-u),$$

  donde hemos utilizado el hecho que de K es simétrica en torno al cero.

- La integral de K^* en $\mathbb{R}$ es igual a 1: directamente:

  $$\int_{\mathbb{R}} K^*(u)\,\mathrm{d}u = \lambda \int_{\mathbb{R}} K(\lambda u)\,\mathrm{d}u = \int_{\mathbb{R}} K(v)\,\mathrm{d}v = 1,$$

  donde en la última igualdad hemos usado que K es una función de densidad.

Queda probado entonces que K^* es una función de tipo núcleo. □

Aplicando el Lema 3.4 a la función K_E^* definida en la Ecuación (3.20) obtenemos que K_E^* es una función de tipo núcleo.

A la hora de generar muestras aleatorios a partir de una variable con densidad dada por la Ecuación (3.20) usaremos el método de aceptación-rechazo. Este método se basa en usar una

acotación de la función de densidad de la que queremos extraer muestras, dicha acotación debe ser lo más próxima posible a la densidad en cuestión pero que podamos invertir de forma sencilla. Buscamos que la acotación sea lo más cercana a la densidad original para que el número de rechazos no sea muy elevado y aumente el tiempo de computación necesario. En este caso usaremos la siguiente función:

$$g(u) = \begin{cases} 3/4, & \text{si } |u| \leq 1, \\ 0, & \text{si } |u| > 1, \end{cases}$$

que tiene por función acumulada:

$$G(u) = \begin{cases} 0, & \text{si } u < -1, \\ \dfrac{3}{4}(u+1), & \text{si } -1 \leq u \leq 1, \\ 3/2, & \text{si } u > 1. \end{cases}$$

Vamos a necesitar la inversa de G:

$$G^{-1}(u) = \frac{4}{3}u - 1, \quad u \in [-1, 1].$$

La finalidad de este método es generar un valor t por el método de inversión aplicado a g. El método de inversión se vale de que si F es la función de distribución de una variable X entonces $F(X) \sim \mathcal{U}(0,1)$ y lo que se consigue son valores de X haciendo $F^{-1}(\mathcal{U}(0,1))$, pero como en este caso G no es una función de distribución (basta notar que $G(u) = 3/2$ cuando $u \to \infty$) debemos reescalar por este parámetro la uniforme. Entonces generamos un valor aleatorio a partir de $\mathcal{U}(0, 3/2)$ y le aplicamos G^{-1}. Finalmente, aceptaremos el valor t que procede de G^{-1} con probabilidad $K_E^*(t)/g(t)$, y repetimos este procedimiento hasta que t sea aceptado. Así, podemos programar la generación de números aleatorios a partir de la densidad de Epanechnikov como sigue (ver el Código A.14):

(1) Consideramos un valor aleatorio u a partir de la $\mathcal{U}(0, 3/2)$ (en R generamos muestras a partir de una uniforme con el comando `runif`). Por el método de inversión, obtenemos un valor con densidad g al aplicarle a u la función G^{-1}, esto es, $t = G^{-1}(u)$ y aceptamos este valor con probabilidad $\dfrac{K_E^*(t)}{g(t)}$.

(2) Repetimos el punto (1) hasta que aceptemos un valor, podemos hacer esto con un bucle *while*.

(3) Repetimos los apartados anteriores tantas veces como valores aleatorios queramos de la distribución, en este caso $k = 10^5$.

Los valores aleatorios de una variable que tiene por función de densidad a K_E^* pueden verse en la Figura 3.6.

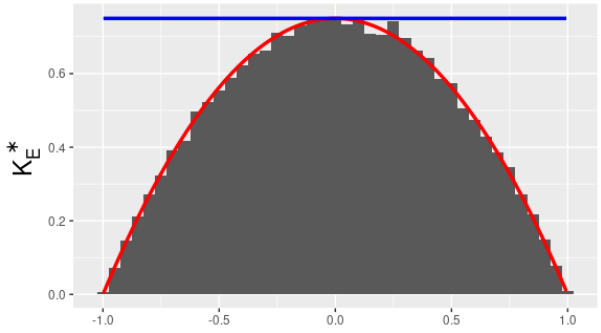


Figura 3.6: Distribución de Epanechnikov (ver la Ecuación (3.20)), en rojo la función de densidad de Epanechnikov y en azul la densidad que usamos para la acotación.

Notemos que el núcleo gaussiano tiene soporte infinito, mientras que los dos siguientes tienen soporte en $[-1, 1]$. Así, usar el núcleo gaussiano implica que la estimación de la función de densidad dará más peso a las colas que los núcleos rectangular y de Epanechnikov (que tienen soporte finito y por lo tanto se anulan fuera de él).

En este momento estamos preparados para iniciar el estudio sobre la sensibilidad de los resultados al cambiar el núcleo usado en la estimación de la función de densidad. Para ello, seguiremos unos pasos similares a los de la Sección §3.3.1, generaremos 100 muestras a partir de una normal de media 17.4 y desviación típica 2.1, y para cada una de ellas realizaremos el método bootstrap suavizado cambiando el tipo de kernel utilizado en la estimación de la función de densidad, para terminar calculando el error relativo medio de cada variación. Vamos a seguir los siguientes pasos (ver el Código A.15):

(1) En primer lugar, generamos 100 muestras aleatorias de tamaño 25 a partir de una normal $\mathcal{N}(17.4, 2.1)$ y las guardamos en una matriz 25×100. En R podemos hacer esto con los comandos `rnorm` y `matrix`, y un bucle *for* que itere las columnas de la matriz.

(2) Hallamos la cuasivarianza de cada muestra y guardamos los resultados en un vector.

En R utilizamos los comandos `apply` sobre las columnas de la matriz del punto (1) y `var`.

(3) Creamos un vector que contenga los anchos de ventana de cada muestra siguiendo la Ecuación (3.17). En R podemos proceder usando la función `density` aplicada a cada muestra con la función `apply` por columnas a la matriz que contiene las muestras y luego iterar sobre la salida de los anteriores comandos con un bucle *for*, en donde se extraiga el parámetro de suavizado con la expresión `$bw`.

(4) Para obtener números aleatorios que tengan por función de densidad la Ecuación (3.20) reutilizaremos el código usado para obtener la Figura 3.6, en particular las expresiones de las funciones $g(u), G^{-1}(u)$ y la cota $C = 3/2$.

(5) Creamos tres matrices (una para cada tipo de núcleo) de dimensiones $B \times 100$, donde B es el número de réplicas bootstrap (por ejemplo, $B = 10^5$). Haremos lo siguiente para la i-ésima muestra:

   (a) Generamos un vector de $n = 25$ elementos que tengan por función de densidad a la función K_E^* de la Ecuación (3.20). Para hacerlo en R tomaremos prestado de nuevo el fragmento de código usado para la obtención de la Figura 3.6.

   (b) Creamos una remuestra a partir de la i-ésima muestra (en R con el comando `sample` con reemplazamiento), y la perturbamos de tres maneras distintas: primero con una normal $\mathcal{N}(0, 1)$, seguida de una uniforme $\mathcal{U}(-1, 1)$ y por último con una variable con densidad K_E^*, todas ellas multiplicadas por el parámetro de suavizado asociado a la i-ésima muestra. Para hacerlo en R, podemos usar `rnorm` y `runif` para los dos primeros núcleos, mientras que para el núcleo de Epanechnikov usaremos los n valores que obtuvimos en el apartado (a).

   (c) Calculamos las cuasivarianzas de los vectores que hemos obtenido de cada método, $\widehat{S}_X^{2(b)}$ (podemos hacerlo con `var` en R) con $b \in \{1, \ldots, B\}$ y hallamos el valor estadístico T de la Ecuación (2.1) con cada una de estas cuasivarianzas cambiando σ^2 por S_X^2 de la i-ésima muestra y $\widehat{S}_X^2$ por $\widehat{S}_X^{2(b)}$.

   (d) Repetimos los apartados (a)-(c) un número B de veces y guardamos los resultados en las columnas de las matrices que creamos al inicio del punto (5). En R usaremos

la instrucción `replicate`.

(6) Iteramos con un bucle *for* desde $i = 1$ hasta 100 el punto (5).

(7) A continuación, hallaremos la media de los errores relativos de las cien muestras para cada método.

   (a) Hallamos el j-ésimo momento de la i-ésima muestra de uno de los tres métodos como la media de la i-ésima columna de la matriz que contenga las distribuciones bootstrap elevada a j. En R usaremos la función `apply` sobre las columnas de la distribución bootstrap que corresponda al método junto con la función `mean`. Así, obtendremos un vector de longitud 100 con los momentos de orden j de cada distribución bootstrap asociada a cada muestra. Para cada entrada de dicho vector calculamos el error relativo con el resultado de la Ecuación (2.2) para $k = n - 1$ y $m = j$. Por último, hallamos la media de los errores relativos, en R con el comando `mean`.

   (b) Repetimos el apartado (a) para cada método, en total tres veces.

(8) Iteramos con un bucle *for* desde $j = 1$ hasta 4 (solo queremos los primeros cuatro momentos).

Podemos observar los resultados en la Tabla 3.2.

Errores relativos (%)	Orden 1	Orden 2	Orden 3	Orden 4
Núcleo gaussiano ($*$)	15	31	49	67
Núcleo rectangular ($*$)	2	4	5	7
Núcleo Epanechnikov ($*$)	0.7	2	5	10

Tabla 3.2: Errores relativos para cada modificación del núcleo y para cada orden.

Como podemos ver, los núcleos rectangular y de Epanechnikov mejoran mucho el error relativo respecto del núcleo gaussiano, lo que entra en contradicción con lo que habíamos mencionado en la Sección §3.3, pues esperábamos que el cambio del núcleo no afectase mucho a los resultados obtenidos.

Las discrepancias observadas pueden explicarse en términos de la varianza de los núcleos utilizados en cada caso. Notemos que en el caso gaussiano hemos usado una normal $\mathcal{N}(0,1)$ escalada por h, es decir, hemos perturbado la remuestra con una variable $X_1 \sim h\,\mathcal{N}(0,1)$, cuya varianza es $\mathrm{Var}(X_1) = h^2$. Para el caso rectangular, perturbamos la remuestra con una variable uniforme $X_2 \sim h\,\mathcal{U}(-1,1)$ cuya varianza es $\mathrm{Var}(X_2) = h^2/3$. Y, por último, para el núcleo de Epanechnikov hemos perturbado la remuestra con una variable $X_3 \sim h\,\mathrm{Ep}$, donde Ep es una variable aleatoria que tiene por densidad la función definida en la Ecuación (3.20). Calculemos la varianza de una variable $Y \sim \mathrm{Ep}$:

$$\mathrm{Var}(Y) = \mathrm{E}(Y^2) - \mathrm{E}(Y)^2 = \frac{3}{4}\int_{-1}^{1}(1-y^2)y^2\,\mathrm{d}y - \frac{3}{4}\int_{-1}^{1}(1-y^2)y\,\mathrm{d}y = \frac{1}{5},$$

por lo tanto $\mathrm{Var}(X_3) = h^2/5$. Acabamos de ver que la varianza de la variable gaussiana es mayor que la de los otros dos núcleos y por eso los resultados son peores, pues la dispersión respecto de la media es mayor en el caso gaussiano. Es por ello que en la Sección §3.3.1 al reescalar el parámetro de suavizado a $h/2$ disminuimos el error, pues la varianza de $X_4 \sim h/2\,\mathcal{N}(0,1)$ es $\mathrm{Var}(X_4) = h^2/4$, produciendo de esta manera unos errores relativos (ver la Tabla 3.1 factor 1/2) similares a los que obtuvimos en la Tabla 3.2 para los núcleos rectangular y de Epanechnikov.

Por lo tanto, podemos concluir que las diferencias observadas entre los resultados de la Tabla 3.2 se deben a que estamos comparando núcleos que no tienen la misma varianza, y entonces las estimaciones de los núcleos con una varianza menor (a igualdad de h) producen resultados con menor error relativo. En la Figura 3.7 representamos las densidades de $1/2\mathcal{N}(0,1)$, $\mathcal{U}(-1,1)$ y una variable con densidad K_E^* dada por la Ecuación (3.20), junto con el núcleo gaussiano $\mathcal{N}(0,1)$. Podemos observar que efectivamente la distribución $\mathcal{N}(0,1)$ es más ancha que el resto de núcleos, debido a que su varianza es mayor.

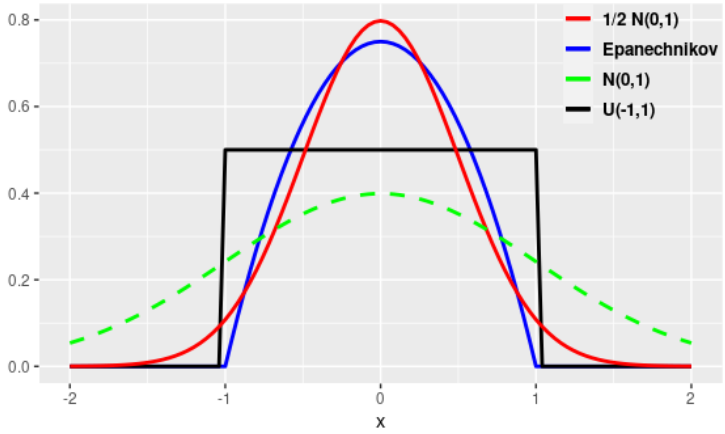


Figura 3.7: Funciones de densidad de $1/2\mathcal{N}(0,1)$, $\mathcal{U}(-1,1)$ y una variable con densidad K_E^*, junto con la densidad de $\mathcal{N}(0,1)$ (línea discontinua verde).

3.4. Estudio comparativo entre los métodos

Para finalizar este capítulo, haremos un estudio de los cuatro métodos que hemos visto hasta ahora: bootstrap uniforme, paramétrico, simétrico y suavizado. Para ello, repetiremos el Ejemplo 2.1 para los cuatro métodos con 100 muestras que proceden de una normal $\mathcal{N}(17.4, 2.1)$. En el caso del bootstrap suavizado elegiremos un núcleo gaussiano con parámetro de suavizado escalado por un factor $1/2$, pues como hemos visto en la Sección §3.3.1 mejoraba el método suavizado. Para el programa del cálculo podemos seguir los siguientes pasos (ver el Código A.16):

(1) Creamos una matriz 25×100 que tenga por columnas las muestras aleatorias de la normal que estamos considerando. En R podemos iterar con un bucle *for* las columnas de una matriz creada con `matrix` e introduciendo en ellas las muestras creadas con el comando `rnorm`.

(2) Calculamos la cuasivarianza y los valores de $\widehat{\mu}_{MV}$ y $\widehat{\sigma}^2_{MV}$ de cada muestra y las guardamos en tres vectores (estos dos últimos dados por la Ecuación (3.3)). En R usamos la función `apply` junto con los comandos `mean` y `var` aplicados a las columnas de la matriz de muestras.

(3) Ahora vamos a crear una matriz de muestras simetrizadas siguiendo la Ecuación (3.12). Para hacerlo en R, unimos por filas con el comando `rbind` la matriz de muestras

con la matriz que contiene la simetrización de las muestras. Para hacer esto último, aplicamos la función `sweep` a la matriz de muestras, eligiendo que la operación se haga por columnas, junto con $2\overline{X}_i$ para la i-ésima columna. Esto realiza la resta $\vec{X}_i - 2\overline{X}_i$, por lo que multiplicando por -1 el resultado de la función `sweep` obtenemos lo que buscábamos.

(4) Estimamos la función de densidad con la Ecuación (3.15) escogiendo K un núcleo gaussiano y h dado por la Ecuación (3.17) (en R podemos usar la función `density` aplicada a la muestra inicial). Además, guardamos en una variable el valor de h.

(5) Creamos cuatro matrices (una por cada método) de dimensiones $B \times 100$, siendo B el número de réplicas bootstrap (en este caso $B = 10^5$). Realizaremos los métodos bootstrap para la i-ésima muestra:

   (a) Bootstrap uniforme: generamos una remuestra de tamaño n a partir de la i-ésima muestra. En R utilizamos `sample` con reemplazamiento activado.

   (b) Bootstrap paramétrico: generamos una remuestra basada en una normal de media $\widehat{\mu}_{MV}$ y desviación típica $\widehat{\sigma}_{MV}$ de la i-ésima muestra. En R usamos la función `rnorm`.

   (c) Bootstrap simétrico: generamos una remuestra de tamaño n a partir de la i-ésima columna de la matriz que habíamos creado en el punto (3). Con R podemos usar `sample` con reemplazamiento.

   (d) Bootstrap suavizado: generamos una remuestra de tamaño n a partir de la i-ésima muestra y la perturbamos sumando $hV_j/2$ con $V_j \sim \mathcal{N}(0,1)$ y $j \in \{1,\ldots,25\}$. Podemos hacer esto en R con las funciones `sample` con reemplazamiento y `rnorm`.

   (e) Calculamos el valor del estadístico T de la Ecuación (2.1) en los puntos (a)-(d) anteriores, cambiando σ^2 por $\widehat{S}_X^2$ de la i-ésima muestra (salvo en el caso del bootstrap paramétrico que cambiaremos σ^2 por $\widehat{\sigma}_{MV}^2$) y $\widehat{S}_X^2$ por la cuasivarianza de los resultados de (a)-(d).

   (f) Repetimos los apartados (a)-(e) un número B de veces y guardamos los resultados en las i-ésimas columnas de las matrices que habíamos creado al inicio del punto (5). En R usamos la función `replicate`.

(6) Iteramos con un bucle *for* desde $i = 1$ hasta 100 el punto (5).

(7) Llegados aquí, tenemos la distribución bootstrap de cada muestra y de cada método, ahora vamos a calcular la media de los errores relativos como sigue:

   (a) Hallamos el j-ésimo momento de la i-ésima muestra de uno de los cuatro métodos como la media de la i-ésima columna de la matriz que contenga las distribuciones bootstrap elevada a j. En R usaremos la función `apply` sobre las columnas de la distribución bootstrap que corresponda al método junto con la función `mean`. Así, obtendremos un vector de longitud 100 con los momentos de orden j de cada distribución bootstrap asociada a cada muestra. Para cada entrada de dicho vector calculamos el error relativo con el resultado de la Ecuación (2.2) para $k = n - 1$ y $m = j$. Por último, hallamos la media de los errores relativos, en R con el comando `mean`.

   (b) Repetimos el apartado (a) para cada método, en total cuatro veces.

(8) Iteramos con un bucle *for* desde $j = 1$ hasta 4 (solo queremos los primeros cuatro momentos).

Los resultados para la media de los errores relativos pueden verse en la Tabla 3.3.

Errores relativos (%)	Orden 1	Orden 2	Orden 3	Orden 4
Uniforme (*)	4	9	14	21
Paramétrico (*)	0.07	0.1	0.2	0.3
Simetrizado (*)	4	9	14	21
Suavizado $h/2$ con kernel $\mathcal{N}(0,1)$	1	1	4	8

Tabla 3.3: Errores relativos para cada método bootstrap y para cada orden.

En color rojo podemos ver los métodos que más error relativo han tenido en media, que han sido los métodos uniforme y simetrizado. Este resultado es razonable, pues el método uniforme era el procedimiento más básico del bootstrap y el método simetrizado es el que menos información aporta en el remuestreo. Por otra parte, en color verde vemos el método con menor error relativo, que fue el bootstrap paramétrico como podíamos esperar del primer estudio que vimos en la Sección §3.1.

En definitiva, podemos asegurar que en media el mejor método bootstrap es el paramétrico, ya que bajo sus hipótesis conocemos la familia paramétrica de la que procede la muestra, y también podemos asegurar que los métodos uniforme y simetrizado son los que mayor error relativo producen.

Capítulo 4

Intervalos de confianza bootstrap

En este capítulo trabajaremos con intervalos de confianza calculados con la aproximación bootstrap y estudiaremos la cobertura y longitud de dichos intervalos. En este contexto, consideramos una variable aleatoria X con función de distribución F_θ, con $\theta \in \Theta \subset \mathbb{R}^d$, y sea $\vec{X} = (X_1, \ldots, X_n)$ una muestra aleatoria simple a partir de X, si existen T_1, T_2 estadísticos tales que:

$$P(\vec{x} \in (X_1, \ldots, X_n) \mid T_1(\vec{x}) \leq g(\theta) \leq T_2(\vec{x})) \geq 1 - \alpha, \quad \forall \theta \in \Theta,$$

donde $\vec{x} = (x_1, \ldots, x_n)$ es una realización muestral, se dirá que $[T_1(\vec{X}), T_2(\vec{X})]$ es un intervalo de confianza para $g(\theta)$ con nivel de confianza $1 - \alpha$.

Dentro del contexto bootstrap, aproximaremos la distribución del parámetro θ con uno de los métodos vistos en el Capítulo §3 y entonces podremos calcular un intervalo de confianza bootstrap, basado en la distribución bootstrap de θ. Para asegurar que la probabilidad de la cobertura sea lo más cercana a $1 - \alpha$, construiremos los intervalos de confianza de acuerdo a diferentes métodos: percentil básico, percentil t y percentil t simetrizado.

Para hallar un intervalo de confianza, muchas veces recurrimos al método de la función pivotal para la construcción de dicho intervalo. Una función pivotal es una función que depende de la muestra $(X_1, \ldots, X_n)$ y del parámetro θ, sea $T = T(X_1, \ldots, X_n, \theta)$, y cuya distribución está completamente especificada. Para construir el intervalo, primero escogemos valores α_1, α_2 tales que $\alpha_1 + \alpha_2 = \alpha$ (se suelen escoger $\alpha_1 = \alpha_2 = \alpha/2$), y de estos hallamos los cuantiles c_1, c_2 que verifican:

$$P(\vec{x} \in (X_1, \ldots, X_n) \mid T(X_1, \ldots, X_n, \theta) \leq c_1) \leq \alpha_1,$$
$$P(\vec{x} \in (X_1, \ldots, X_n) \mid T(X_1, \ldots, X_n, \theta) \leq c_2) \geq 1 - \alpha_2.$$

Así, obtenemos:

$$P(\vec{x} \in \vec{X} \mid c_1 \leq T(\vec{X}, \theta) \leq c_2) = P(\vec{x} \in \vec{X} \mid T(\vec{X}, \theta) \leq c_2) - P(\vec{x} \in \vec{X} \mid T(\vec{X}, \theta) \leq c_1) =$$
$$\geq 1 - \alpha_2 - \alpha_1 = 1 - \alpha.$$

Para obtener el intervalo de confianza bastaría entonces expresar $c_1 \leq T(\vec{X}, \theta) \leq c_2$ como un intervalo centrado en θ.

4.1. Método percentil básico

4.1.1. Desarrollo teórico

El método percentil básico se apoya en la construcción de un intervalo de confianza (bootstrap) usando el estadístico:

$$T_{\text{bas}} = \widehat{\theta} - \theta, \tag{4.1}$$

siendo $\widehat{\theta}$ un estimador del parámetro θ. Ahora aplicamos el método bootstrap: vamos a estimar la función de distribución F utilizando uno de los procedimientos vistos en el Capítulo §3 obteniendo $\widehat{F}$. Entonces el estadístico T_{bas} de la Ecuación (4.1) adopta la siguiente forma en el contexto bootstrap:

$$T^*_{\text{bas}} = \widehat{\theta}^* - \widehat{\theta}. \tag{4.2}$$

Con la distribución del estadístico de la Ecuación (4.2) podemos hallar valores c_1^*, c_2^* tales que:

$$P^*(T^*_{\text{bas}} \leq c_1^*) = \frac{\alpha}{2}, \quad P^*(T^*_{\text{bas}} \leq c_2^*) = 1 - \frac{\alpha}{2},$$

y así se verifica lo siguiente:

$$P^*(c_1^* \leq T^*_{\text{bas}} \leq c_2^*) = P^*(T^*_{\text{bas}} \leq c_2^*) - P^*(T^*_{\text{bas}} \leq c_1^*) = 1 - \frac{\alpha}{2} - \frac{\alpha}{2} = 1 - \alpha.$$

De lo anterior podemos deducir que $P(c_1^* \leq T_{\text{bas}} \leq c_2^*)$ es aproximadamente igual a $1 - \alpha$, esto es:

$$1 - \alpha \simeq P(c_1^* \leq T_{\text{bas}} \leq c_2^*) = P(c_1^* \leq \widehat{\theta} - \theta \leq c_2^*) = P\left(\widehat{\theta} - c_2^* \leq \theta \leq \widehat{\theta} - c_1^*\right),$$

y finalmente hemos obtenido un intervalo de confianza para θ en función de los valores c_1^*, c_2^* que hallamos con la distribución bootstrap:

$$I^* = \left[\widehat{\theta} - c_2^*, \widehat{\theta} - c_1^*\right]. \tag{4.3}$$

Para poner en práctica el método percentil básico vamos a resolver el siguiente ejemplo.

Ejemplo 4.1. Se han registrado 30 observaciones de una variable aleatoria que sigue una distribución normal de media 17.4 (desconocido) y desviación típica 2.1 (desconocido). Halla un intervalo de confianza con nivel de confianza 0.95 para la desviación típica utilizando el método percentil básico.

Para resolver el ejercicio, vamos a empezar con el método bootstrap uniforme para luego utilizar las modificaciones del bootstrap vistas en el Capítulo §3.

- Uniforme: supongamos que $\vec{X} = (X_1, \ldots, X_n)$ es una muestra aleatoria simple a partir de una variable aleatoria normal $X \sim \mathcal{N}(\mu, \sigma)$, entonces el estadístico que vamos a usar en este caso es:

$$\widehat{\theta} = \widehat{S}_X^2,$$

y como parámetro a estimar tomaremos por simplicidad $\theta = \sigma^2$, por lo tanto el estadístico del método percentil básico es:

$$T_{\text{bas}} = \widehat{\theta} - \theta. \tag{4.4}$$

Por otro lado, el estadístico bootstrap asociado a la b-ésima réplica bootstrap es:

$$T^*_{\text{bas}} = \widehat{S}_X^{2(b)} - \widehat{\theta} = \widehat{\theta}^* - \widehat{\theta},$$

con $b \in \{1, \ldots, B\}$. Así, generaremos B remuestras a partir de la muestra inicial y calcularemos para cada una de ellas su cuasivarianza bootstrap, con lo que tendremos un vector de longitud B del que podremos hallar los cuantiles c_1^*, c_2^* y concluiremos el cálculo usando la Ecuación (4.3). Para ello hemos seguido los siguientes pasos (ver el Código A.17):

(1) Generamos una muestra aleatoria de una normal de media 17.4 y desviación típica 2.1, en R podemos utilizar la instrucción `rnorm`. De esta muestra, calculamos su cuasivarianza (en R usamos el comando `var`).

(2) Generamos una remuestra de tamaño n a partir de la muestra del punto anterior, en R utilizamos `sample` con reemplazamiento, y calculamos su cuasivarianza muestral (de nuevo, con `var`). De esta forma tendremos un valor $S_X^{2(b)}$.

(3) Reiteramos el punto (2) para $b \in \{1, \ldots, B\}$, con $B = 10^5$ en nuestro caso. Este proceso puede realizarse con la función `replicate` en R.

(4) A partir del vector que obtenemos del punto (3) y la cuasivarianza del punto (1), aplicamos la Ecuación (4.4) obteniendo de esta forma la distribución de T^*_{bas}, y con ella calculamos los cuantiles c^*_1, c^*_2 asociados a las probabilidades $\alpha/2$ y $1-\alpha/2$ respectivamente (notemos que como buscamos un nivel de confianza de 0.95, necesariamente se tiene que $\alpha = 0.05$).

(5) Con los cuantiles c^*_1, c^*_2, usamos la Ecuación (4.3) para obtener el intervalo de confianza buscado.

El intervalo que obtenemos para σ^2 es:

$$I^*(\sigma^2) = [1.79,\ 5.74],$$

y por lo tanto el intervalo de confianza bootstrap para σ es:

$$I^*(\sigma) = [1.34,\ 2.40].$$

- Paramétrico: en este caso vamos a generar remuestras a partir de una variable aleatoria normal de media $\widehat{\mu}_{MV}$ y de desviación típica el valor $\widehat{\sigma}_{MV}$, los cuales conocemos por la Ecuación (3.3). Entonces tenemos que $\widehat{\theta} = \widehat{S}^2_X$ y por lo tanto $\widehat{\theta}^* = \widehat{S}^{2(b)}_X$ es el estimador asociado a la b-ésima réplica bootstrap, con $b \in \{1, \ldots, B\}$. Para llevar a cabo el cálculo, basta sustituir la forma de remuestrear del método uniforme por una muestra aleatoria de la normal que hemos mencionado anteriormente (ver el Código A.18).

  El intervalo para σ^2 es:

  $$I^*(\sigma^2) = [1.79,\ 5.52],$$

  y el intervalo de confianza para σ:

  $$I^*(\sigma) = [1.34,\ 2.35].$$

- Simétrico: partiendo de nuevo del procedimiento seguido en el bootstrap uniforme, creamos el vector que contiene la muestra simetrizada $\vec{Y}$ tal y como definimos sus componentes en la Ecuación (3.12). A partir de esta nueva muestra realizamos el remuestreo bootstrap y hallamos el valor del estimador $\widehat{\theta}^* = \widehat{S}^{2(b)}_X$ asociado a la b-ésima réplica bootstrap, donde $b \in \{1, \ldots, B\}$. Es por ello que en los pasos que seguimos en

el método uniforme solo tendríamos que calcular la remuestra a partir de la muestra simetrizada $\vec{Y}$ (ver el Código A.19).

El intervalo para σ^2 es el siguiente:

$$I^*(\sigma^2) = [1.70,\ 5.65],$$

mientras que el intervalo de confianza para σ es:

$$I^*(\sigma) = [1.30,\ 2.38].$$

- Suavizado: en este último caso, vamos a estimar la función de densidad de la muestra con una función de tipo núcleo. Como vimos en la Sección §3.3, la elección del núcleo no era demasiado relevante, en cambio una elección adecuada del parámetro de suavizado h proporcionaba unos resultados mejores. Por lo tanto vamos a elegir un núcleo gaussiano y vamos a reescalar el parámetro h por $1/2$ (como vimos en la Sección §3.3.1 era el que mejores resultados reproducía). Entonces tan solo tendremos que calcular el parámetro de suavizado usando la Ecuación (3.17) y modificar el remuestreo del método uniforme perturbando la remuestra: sumaremos el término $h/2\mathcal{N}(0,1)$ tal y como hicimos en la Sección §3.3 (ver el Código A.20).

  El intervalo que obtenemos para σ^2 es:

$$I^*(\sigma^2) = [1.51,\ 5.61],$$

  y el intervalo de confianza para σ es:

$$I^*(\sigma) = [1.23,\ 2.37].$$

Para finalizar el ejemplo del cálculo de intervalos de confianza bootstrap, vamos a hallar el intervalo de confianza para σ utilizando un método analítico y compararemos con los resultados anteriores. Para ello, vamos a hacer uso de que conocemos la distribución del estadístico:

$$T = \frac{n-1}{\sigma^2}\widehat{S}_X^2 \sim \chi^2_{n-1},$$

y entonces podemos calcular los cuantiles d_1, d_2 tales que:

$$P(T \leq d_1) = \frac{\alpha}{2}, \quad P(T \leq d_2) = 1 - \frac{\alpha}{2},$$

de donde obtenemos:

$$1-\alpha = P\left(d_1 \leq T \leq d_2\right) = P\left(\frac{n-1}{d_2}\widehat{S}_X^2 \leq \sigma^2 \leq \frac{n-1}{d_1}\widehat{S}_X^2\right),$$

y podemos concluir entonces que el intervalo de confianza para σ es:

$$I(\sigma) = \left[\sqrt{\frac{n-1}{d_2}\widehat{S}_X^2}, \sqrt{\frac{n-1}{d_1}\widehat{S}_X^2}\right]. \tag{4.5}$$

Para programar el cálculo hemos seguido el siguiente procedimiento (ver el Código A.21):

(1) Generamos una muestra aleatoria a partir de una normal de media 17.4 y desviación típica 2.1 (en R con `rnorm`), y calculamos su varianza (con el comando `var` en R).

(2) Hallamos los cuantiles d_1, d_2 de una χ^2_{n-1} asociados a las probabilidades $\alpha/2, 1-\alpha/2$ respectivamente. En R podemos hacer lo anterior con la instrucción `qchisq`.

(3) Aplicamos la Ecuación (4.5) para determinar el intervalo de confianza para σ.

El intervalo que hemos hallado para σ^2 es:

$$I(\sigma^2) = [2.39,\ 6.81],$$

y el intervalo de confianza para σ:

$$I(\sigma) = [1.55,\ 2.61].$$

Como podemos comprobar en el ejemplo anterior, todos los intervalos de confianza bootstrap que obtenemos para σ son similares entre sí y no podemos determinar cuál de los métodos nos proporciona un intervalo de confianza mejor que los demás. Por otro lado, los extremos del intervalo de confianza que hemos calculado con el método exacto son mayores que los extremos que obtenemos con los métodos bootstrap. Normalmente, buscamos que la longitud del intervalo de confianza sea relativamente pequeña y que contenga el verdadero valor del parámetro a estimar (en nuestro caso σ). En la Tabla 4.1 podemos observar los resultados obtenidos.

	Uniforme	Paramétrico	Simetrizado	Suavizado	Exacto
Extremo inferior	1.34	1.34	1.30	1.23	1.55
Extremo superior	2.40	2.35	2.38	2.37	2.61

Tabla 4.1: Valores de cada extremo del intervalo de confianza de σ de cada método (percentil básico).

En la siguiente sección investigaremos la longitud media de los intervalos bootstrap y exacto, y cuál de los métodos ofrece una mayor cobertura, esto es, cuál de los métodos ofrece una mayor proporción de intervalos que contengan el verdadero valor del parámetro a estimar, que debería ser cercano al $(1-\alpha)\times 100\,\%$.

4.1.2. Cobertura y longitud de los intervalos del método percentil básico

Para esta sección, vamos a generar un número grande de muestras de tamaño $n = 30$, por ejemplo 1000 muestras, y veremos cuántas veces los intervalos bootstrap que construimos contienen el verdadero parámetro que queremos estimar, que en este caso será σ. Además, calcularemos la longitud media de los intervalos producidos por cada método bootstrap.
Para ello, recuperaremos la variable aleatoria $\mathcal{N}(17.4, 2.1)$ y seguiremos los siguientes pasos (ver el Código A.22):

(1) Para facilitar el cálculo, vamos a crear una función que tome como parámetros la muestra que estemos considerando, el método por el que queremos hallar el intervalo de confianza, el número de réplicas bootstrap B, el valor α y el estimador $\widehat{\theta}$. Separando los casos por métodos bootstrap (podemos usar la sentencia `if`), vamos calculando los intervalos de confianza como vimos en el Ejemplo 4.1, para después hallar los cuantiles c_1^*, c_2^* (con la instrucción `quantile` en R). Finalmente, la función debe devolver los extremos del intervalo calculados utilizando la Ecuación (4.3).

(2) Ahora generamos una muestra a partir de una normal $\mathcal{N}(17.4, 2.1)$ (en R usamos `rnorm`), y de ella calculamos su cuasivarianza. Vamos a llamar a la función que habíamos creado en el punto (1) para cada uno de los métodos y tomando $\widehat{\theta} = \widehat{S}_X^2$ (además de sus correspondientes estimadores bootstrap), $B = 10^4$ (para evitar tiempos de computación elevados) y $\alpha = 0.05$. Además, calcularemos el intervalo de confianza exacto tal y como lo hicimos en el Ejemplo 4.1, así terminaremos con cinco intervalos de confianza.

(3) Repetimos el paso (2) tantas veces como muestras se quieran, en nuestro caso 1000 veces (en R con el comando `replicate`).

(4) Antes de hallar la cobertura, notemos en primer lugar que hemos calculado con la función definida en (1) un intervalo de confianza para σ^2, entonces tendremos que tomar

las raíces cuadradas de los extremos de los intervalos. Tras esto, podemos hallar la cobertura de los intervalos calculando la proporción de ellos que contienen el verdadero valor del parámetro, es decir, hallaremos la media de los intervalos que satisfacen que $\sigma \in I^*$ para cada método.

(5) Para la longitud de los intervalos hallaremos la media de la longitud de los intervalos, siendo esta la diferencia entre los extremos de dichos intervalos.

Los resultados que hemos obtenido para 1000 muestras están resumidos en la Tabla 4.2. Podemos observar que la cobertura del método exacto es superior al 95 % como deseábamos al inicio de este capítulo, sin embargo la cobertura de los métodos bootstrap es inferior al 95 % pero cercana al 90 %. Por otro lado, la longitud de los intervalos es mayor en el método exacto que la longitud de los intervalos que obtenemos con bootstrap. Esto está directamente relacionado con la cobertura de los intervalos: para asegurar que al menos un 95 % de los intervalos contengan el verdadero valor del parámetro es necesario que los intervalos tengan una longitud mayor. Como vimos en el Capítulo §2, aumentando el tamaño muestral obtendríamos mejores resultados pues la aproximación bootstrap tendría más información sobre la población.

	Uniforme	Paramétrico	Simetrizado	Suavizado	Exacto
Cobertura	0.894	0.909	0.890	0.891	0.956
Longitud	0.998	1.087	1.003	1.089	1.144

Tabla 4.2: Cobertura y longitud (media) de los intervalos de confianza de σ de cada método.

En las siguientes secciones estudiaremos métodos que intentan mejorar la cobertura de los intervalos de confianza bootstrap.

4.2. Método percentil *t*

4.2.1. Desarrollo teórico

El método percentil t se basa en construir un intervalo de confianza utilizando el siguiente estadístico:

$$T_t = \frac{\widehat{\theta} - \theta}{\sigma_{\widehat{\theta}}}, \tag{4.6}$$

donde $\widehat{\theta}$ es un estimador del parámetro θ y $\sigma_{\widehat{\theta}}$ un estimador de la desviación típica de $\widehat{\theta}$. Cuando estamos en el contexto bootstrap, la Ecuación (4.6) se reescribe de la siguiente manera:

$$T_t^* = \frac{\widehat{\theta}^* - \widehat{\theta}}{\sigma_{\widehat{\theta}^*}}, \tag{4.7}$$

donde ahora $\widehat{\theta}^*$ es el estimador $\widehat{\theta}$ asociado a una remuestra bootstrap y $\sigma_{\widehat{\theta}^*}$ es un estimador de la desviación típica de $\widehat{\theta}^*$.

Conociendo la distribución del estadístico de la Ecuación (4.7) calculamos los cuantiles c_1^*, c_2^* tales que:

$$P^*(T_t^* \leq c_1^*) = \frac{\alpha}{2}, \quad P^*(T_t^* \leq c_2^*) = 1 - \frac{\alpha}{2}.$$

De esta forma aseguramos un nivel de confianza $P^*(c_1^* \leq T_t^* \leq c_2^*) = 1 - \alpha$. Así, podemos aproximar lo siguiente:

$$1 - \alpha \simeq P(c_1^* \leq T_t \leq c_2^*) = P\left(\widehat{\theta} - c_2^*\,\sigma_{\widehat{\theta}} \leq \theta \leq \widehat{\theta} - c_1^*\,\sigma_{\widehat{\theta}}\right),$$

lo que nos deja el siguiente intervalo de confianza bootstrap:

$$I^* = \left[\widehat{\theta} - c_2^*\,\sigma_{\widehat{\theta}}, \widehat{\theta} - c_1^*\,\sigma_{\widehat{\theta}}\right]. \tag{4.8}$$

Debemos observar que esta expresión del intervalo de confianza solo es válida cuando la estimación $\sigma_{\widehat{\theta}}$ no depende de θ. En caso de que existiera una dependencia habría que buscar la forma de centrar el intervalo en θ.

La desventaja principal del método t es que es necesario conocer una expresión explícita del estimador $\sigma_{\widehat{\theta}}$, lo que en la mayoría de las ocasiones supone conocer la distribución de la población. En los ejemplos en los que usemos este método supondremos conocida la distribución de la población para hallar la expresión de $\sigma_{\widehat{\theta}}$. De este último podemos obtener directamente $\sigma_{\widehat{\theta}^*}$ (cambiando θ por $\widehat{\theta}$ y $\widehat{\theta}$ por $\widehat{\theta}^*$), o incluso podemos hallar $\sigma_{\widehat{\theta}^*}$ calculando numéricamente su desviación típica (ya que con las réplicas bootstrap conseguimos la distribución de $\widehat{\theta}^*$).

Recuperemos el Ejemplo 4.1.

Ejemplo 4.2. Se han registrado 30 observaciones de una variable aleatoria que sigue una distribución normal de media 17.4 (desconocido) y desviación típica 2.1 (desconocido). Halla un intervalo de confianza con nivel de confianza 0.95 para la desviación típica empleando el método percentil t.

Resolveremos este ejercicio utilizando primero el método bootstrap uniforme y después el resto de métodos que conocemos.

- Uniforme: supongamos que $\vec{X} = (X_1, \ldots, X_n)$ es una muestra aleatoria simple a partir de $X \sim \mathcal{N}(17.4, 2.1)$ y el estimador que vamos a usar para $\theta = \sigma^2$ es de nuevo:

$$\widehat{\theta} = \widehat{S}_X^2 = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \overline{X})^2.$$

En muchas ocasiones el problema del método percentil t es hallar $\sigma_{\widehat{\theta}}$, pero en este caso como la población es normal podemos usar el Teorema de Fisher sobre la cuasivarianza[1]:

$$\frac{n-1}{\sigma^2}\widehat{\theta} = \frac{n-1}{\sigma^2}\widehat{S}_X^2 \sim \chi^2_{n-1},$$

y es conocido que la varianza de una variable $Z \sim \chi^2_k$ es $\mathrm{Var}(Z) = 2k$, por lo tanto deducimos que:

$$2(n-1) = \mathrm{Var}\left(\frac{n-1}{\sigma^2}\widehat{S}_X^2\right) = \frac{(n-1)^2}{\sigma^4}\mathrm{Var}\left(\widehat{S}_X^2\right) \Rightarrow \mathrm{Var}\left(\widehat{S}_X^2\right) = \frac{2\sigma^4}{n-1},$$

y así concluimos que:

$$\sigma_{\widehat{\theta}} = \sqrt{\mathrm{Var}\left(\widehat{S}_X^2\right)} = \sqrt{\frac{2}{n-1}}\sigma^2 = \sqrt{\frac{2}{n-1}}\,\theta.$$

Como podemos ver, $\sigma_{\widehat{\theta}}$ depende del parámetro $\theta = \sigma^2$, por lo que no podremos usar la expresión de la Ecuación (4.8). En cualquier caso, el estadístico del método percentil t que nos queda es el siguiente:

$$T_t = \sqrt{\frac{n-1}{2}}\frac{\widehat{S}_X^2 - \sigma^2}{\sigma^2}, \tag{4.9}$$

y por lo tanto el estadístico bootstrap del método percentil t asociado a la b-ésima réplica es:

$$T_t^* = \sqrt{\frac{n-1}{2}}\frac{\widehat{S}_X^{2(b)} - \widehat{S}_X^2}{\widehat{S}_X^2}. \tag{4.10}$$

[1] Aquí supondremos conocida la distribución de la población para poder usar el método t.

donde $b \in \{1, \ldots, B\}$. A partir de aquí, el procedimiento es completamente análogo salvo la expresión cerrada del intervalo, la cual vamos a hallar en este momento: supongamos que c_1^*, c_2^* son los cuantiles asociados a $\alpha/2, 1-\alpha/2$ de T_t^* respectivamente, entonces:

$$\begin{aligned} 1-\alpha \simeq P\left(c_1^* \leq \sqrt{\frac{n-1}{2}}\frac{\widehat{S}_X^2 - \sigma^2}{\sigma^2} \leq c_2^*\right) = \\ = P\left(\sqrt{\frac{2}{n-1}}\, c_1^* \leq \frac{\widehat{S}_X^2}{\sigma^2} - 1 \leq \sqrt{\frac{2}{n-1}}\, c_2^*\right) = \\ = P\left(\frac{\widehat{S}_X^2}{1+\sqrt{\frac{2}{n-1}}\, c_2^*} \leq \sigma^2 \leq \frac{\widehat{S}_X^2}{1+\sqrt{\frac{2}{n-1}}\, c_1^*}\right), \end{aligned}$$

y así tenemos el intervalo de confianza para σ:

$$I^* = \left[\sqrt{\frac{\widehat{S}_X^2}{1+\sqrt{\frac{2}{n-1}}\, c_2^*}}, \sqrt{\frac{\widehat{S}_X^2}{1+\sqrt{\frac{2}{n-1}}\, c_1^*}}\right]. \tag{4.11}$$

Ahora vamos a programar el cálculo del intervalo de confianza hemos seguido los siguientes pasos (ver el Código A.23):

(1) Creamos una muestra aleatoria a partir de una variable aleatoria normal de media 17.4 y desviación típica 2.1 (en R con `rnorm`). De ella hallamos su cuasivarianza (con el comando `var` en R).

(2) Generamos una remuestra de tamaño n a partir de la muestra del punto anterior, en R utilizamos `sample` con reemplazamiento, y calculamos su cuasivarianza muestral (de nuevo, con `var`). De esta forma tendremos un valor $S_X^{2(b)}$.

(3) Reiteramos el punto (2) para $b \in \{1, \ldots, B\}$, con $B = 10^5$ en nuestro caso. Este proceso puede realizarse con la función `replicate` en R.

(4) A partir del vector que obtenemos del punto (3) y la cuasivarianza del punto (1), aplicamos la Ecuación (4.10) obteniendo de esta forma la distribución de T_t^*, y con ella calculamos los cuantiles c_1^*, c_2^* asociados a las probabilidades $\alpha/2$ y $1-\alpha/2$ respectivamente (notemos que como buscamos un nivel de confianza de 0.95, necesariamente se tiene que $\alpha = 0.05$).

(5) Con los cuantiles c_1^*, c_2^*, usamos la Ecuación (4.11) para obtener el intervalo de confianza buscado.

El intervalo para σ^2 es el siguiente:

$$I^*(\sigma^2) = [2.47,\ 7.90],$$

mientras que el intervalo de confianza para σ es:

$$I^*(\sigma) = [1.57,\ 2.81].$$

- Paramétrico: en este caso solo cambiaremos la generación de la remuestra respecto del método uniforme, ahora utilizaremos una variable aleatoria normal de media $\widehat{\mu}_{MV}$ y desviación típica $\widehat{\sigma}_{MV}$ definidos en la Ecuación (3.3) (ver el Código A.24).

  El intervalo para σ^2 nos queda:

  $$I^*(\sigma^2) = [2.47,\ 7.04],$$

  y el intervalo de confianza para σ es:

  $$I^*(\sigma) = [1.57,\ 2.65].$$

- Simétrico: en este caso creamos el vector que contiene la muestra simetrizada $\vec{Y}$ definido en la Ecuación (3.12) y de él hacemos el remuestreo. Así, solo tendríamos que cambiar en los pasos del método uniforme la forma en la que remuestreamos (ver el Código A.25).

  El intervalo que obtenemos para σ^2 es:

  $$I^*(\sigma^2) = [2.43,\ 7.54],$$

  mientras que el intervalo de confianza para σ es el siguiente:

  $$I^*(\sigma) = [1.56,\ 2.75].$$

- Suavizado: procedemos como en el caso uniforme, pero para generar la remuestra la perturbamos con una variable aleatoria $h/2\mathcal{N}(0,1)$, donde h se calcula usando la Ecuación (3.17) (ver el Código A.26).

  Así, el intervalo de confianza para σ^2 nos ha quedado:

  $$I^*(\sigma^2) = [2.36,\ 7.40],$$

  y el intervalo de confianza para σ es:

  $$I^*(\sigma) = [1.53,\ 2.72].$$

Para terminar el ejemplo, resumimos los resultados que hemos obtenido por el método percentil t en la Tabla 4.3, junto con el resultado exacto del Ejemplo 4.1. Como podemos comprobar, los resultados utilizando el método t producen unos extremos del intervalo mayores en todos los casos si los comparamos con los resultados de la Tabla 4.1, lo que en principio debería ser mejor pues se asemejan mucho más al intervalo que obteníamos con el método exacto.

	Uniforme	Paramétrico	Simetrizado	Suavizado	Exacto
Extremo inferior	1.57	1.56	1.56	1.53	1.55
Extremo superior	2.81	2.64	2.75	2.72	2.61

Tabla 4.3: Valores de cada extremo del intervalo de confianza de σ de cada método (percentil t).

Sin embargo, como habíamos mencionado previamente, también podemos usar el hecho de que conocemos la distribución bootstrap $\widehat{\theta}^*$ para calcular $\sigma_{\widehat{\theta}^*}$. Es por ello que vamos a repetir el procedimiento del Ejemplo 4.2 pero a la hora de usar la Ecuación (4.7) utilizaremos una estimación de $\sigma_{\widehat{\theta}^*}$ como puede ser la raíz cuadrada de la cuasivarianza de $\widehat{\theta}^*$ (en R usaremos la función `sd`). Es decir, solo cambiaremos la forma de hallar T_t^*.
También usaremos la Ecuación (4.9) para obtener un valor de $\sigma_{\widehat{\theta}}$. Así, los intervalos de confianza bootstrap serán los de la Ecuación (4.11). Los nuevos resultados pueden observarse en la Tabla 4.4. Como podemos ver, son muy similares a los que habíamos obtenido en la Tabla 4.3 dado que solo hemos cambiado la forma de calcular T_t^*.

	Uniforme	Paramétrico	Simetrizado	Suavizado	Exacto
Extremo inferior	1.59	1.56	1.57	1.55	1.55
Extremo superior	2.77	2.69	2.71	2.64	2.61

Tabla 4.4: Valores de cada extremo del intervalo de confianza de σ de cada método (percentil t) usando una estimación de $\sigma_{\widehat{\theta}^*}$.

En la siguiente sección vamos a estudiar la cobertura y la longitud de los intervalos que produce este método.

4.2.2. Cobertura y longitud de los intervalos del método percentil t

Repetiremos el procedimiento de la Sección §4.1.2: calcularemos la cobertura y la longitud media de los intervalos t bootstrap para 1000 muestras de tamaño $n = 30$ y $B = 10^4$. Puesto que las modificaciones que hemos hecho respecto del método percentil básico han sido escasas, solo vamos a realizar unos pequeños cambios con respecto a los pasos especificados en la Sección §4.1.2: como el desarrollo que hemos seguido en el Ejemplo 4.2 ha sido específico del problema, la función definida en el paso (1) no tiene el argumento $\widehat{\theta}$. Además, dentro de dicha función tendremos que cambiar el estadístico T^*_{bas} por el estadístico T^*_t que teníamos en la Ecuación (4.10) y también los extremos del intervalo de confianza por los que obtuvimos en la Ecuación (4.11) (ver el Código A.27).
Los resultados para la cobertura y la longitud media de los intervalos están recogidos en la Tabla 4.5. Como podemos ver, la cobertura ha mejorado con respecto de los resultados de la Tabla 4.2 del método básico, y de hecho las coberturas son todas iguales o superiores al 93 %. Con respecto de la longitud media, obtenemos unos resultados parecidos a los del método exacto, siendo los nuevos intervalos ligeramente más anchos que los del método básico.

	Uniforme	Paramétrico	Simetrizado	Suavizado	Exacto
Cobertura	0.930	0.954	0.931	0.945	0.956
Longitud	1.114	1.143	1.100	1.096	1.144

Tabla 4.5: Cobertura y longitud (media) de los intervalos de confianza de σ de cada método.

Realizando el estudio de simulación con el cambio que considerábamos al final de la sección anterior obtenemos los resultados de la Tabla 4.6. Podemos apreciar que ahora tanto la cobertura como la longitud media son superiores a los que hemos obtenido en la Tabla 4.5, y esto se debe a que la estimación $\sigma_{\widehat{\theta}^*}$ hecha a partir de la distribución de $\widehat{\theta}^*$ es una aproximación, resultando que los nuevos intervalos sean más anchos que los que obtenemos usando la expresión explícita de $\sigma_{\widehat{\theta}^*}$.

	Uniforme	Paramétrico	Simetrizado	Suavizado	Exacto
Cobertura	0.958	0.962	0.958	0.950	0.956
Longitud	1.284	1.216	1.268	1.160	1.144

Tabla 4.6: Cobertura y longitud (media) de los intervalos de confianza de σ de cada método usando una estimación de $\sigma_{\widehat{\theta}^*}$.

4.3. Método percentil *t* simetrizado

4.3.1. Desarrollo teórico

El método percentil t simetrizado es muy similar al método percentil t, la única diferencia entre ambos es la selección de los cuantiles de la distribución. En los métodos anteriores exigíamos la siguiente condición: $P(c_1 \leq T \leq c_2) \geq 1-\alpha$, y habíamos mencionado que esto deja que podamos elegir los valores α_1, α_2 tales que $P(T \leq c_1) \leq \alpha_1$, $P(T \leq c_2) \geq 1-\alpha_2$ y verifiquen $\alpha_1 + \alpha_2 = \alpha$. A la hora de realizar cálculos habíamos escogido $\alpha_1 = \alpha_2 = \alpha/2$, es decir, dejando igual probabilidad en las colas de la distribución. En el caso del bootstrap t simetrizado se eligen los cuantiles de forma que sean simétricos, esto es,

$$P(-c \leq T \leq c) = P(|T| \leq c) \geq 1-\alpha,$$

y su versión en bootstrap:

$$P^*(-c^* \leq T^* \leq c^*) = P^*(|T^*| \leq c^*) \geq 1-\alpha.$$

Particularizando para el caso del método t, el nuevo intervalo de confianza bootstrap que se obtiene es

$$I^* = \left[\widehat{\theta} - c^*\sigma_{\widehat{\theta}}, \widehat{\theta} + c^*\sigma_{\widehat{\theta}}\right].$$

De nuevo, esta expresión solo es válida en caso de que $\sigma_{\widehat{\theta}}$ no dependa de θ.

Volvamos al Ejemplo 4.2 y apliquemos lo que acabamos de introducir.

Ejemplo 4.3. Se han registrado 30 observaciones de una variable aleatoria que sigue una distribución normal de media 17.4 (desconocido) y desviación típica 2.1 (desconocido). Halla un intervalo de confianza con nivel de confianza 0.95 para la desviación típica usando el método percentil t simetrizado.

Resolveremos este ejercicio utilizando primero el método bootstrap uniforme y después el resto de métodos que conocemos.

- Uniforme: puesto que el método t simetrizado es una ligera modificación del método t a la hora de elegir los cuantiles, el desarrollo que seguiremos es totalmente análogo al del Ejemplo 4.2 sin más que elegir $c_1^* = -c^*$ y $c_2^* = c^*$. Para programar el cálculo hemos seguido los siguientes pasos (ver el Código A.28):

  (1) Creamos una muestra aleatoria a partir de una variable aleatoria normal de media 17.4 y desviación típica 2.1 (en R con `rnorm`). De ella hallamos su cuasivarianza (con el comando `var` en R).

  (2) Generamos una remuestra de tamaño n a partir de la muestra del punto anterior, en R utilizamos `sample` con reemplazamiento, y calculamos su cuasivarianza muestral (de nuevo, con `var`). De esta forma tendremos un valor $S_X^{2(b)}$.

  (3) Reiteramos el punto (2) para $b \in \{1, \ldots, B\}$, con $B = 10^5$ en nuestro caso. Este proceso puede realizarse con la función `replicate` en R.

  (4) A partir del vector que obtenemos del punto (3) y la cuasivarianza del punto (1), aplicamos la Ecuación (4.10) obteniendo de esta forma la distribución de T_t^*, y con ella calculamos el cuantil c^* de $|T_t^*|$ asociado a la probabilidad $1-\alpha$ (notemos que como buscamos un nivel de confianza de 0.95, necesariamente se tiene que $\alpha = 0.05$).

  (5) Con los cuantiles $c_1^* = -c^*$ y $c_2^* = c^*$, usamos la Ecuación (4.11) para obtener el intervalo de confianza buscado.

  El intervalo para σ^2 es:

  $$I^*(\sigma^2) = [2.47\,, 7.92],$$

  mientras que el intervalo de confianza para σ nos queda:

  $$I^*(\sigma) = [1.57,\ 2.81].$$

- Paramétrico: ahora cambiaremos la forma de realizar el remuestreo: usaremos una variable aleatoria normal de media $\widehat{\mu}_{MV}$ y desviación típica $\widehat{\sigma}_{MV}$ dados por la Ecuación (3.3) (ver el Código A.29).

El intervalo para σ^2 nos ha quedado:

$$I^*(\sigma^2) = [2.53\,, 7.34],$$

y por otro lado el intervalo de confianza para σ:

$$I^*(\sigma) = [1.59,\ 2.71].$$

- Simétrico: para esta modificación del bootstrap uniforme crearemos el vector de la muestra simetrizada de la Ecuación (3.12) y es de este último del que escogeremos remuestras (ver el Código A.30).

  El intervalo para σ^2 es el siguiente:

  $$I^*(\sigma^2) = [2.48,\ 7.84],$$

  y el intervalo de confianza para σ es:

  $$I^*(\sigma) = [1.57,\ 2.80].$$

- Suavizado: seguiremos los pasos del método uniforme, pero perturbaremos la remuestra añadiendo el término $h/2\mathcal{N}(0,1)$, con h el parámetro de suavizado que definimos en la Ecuación (3.17) (ver el Código A.31).

  Entonces el intervalo para σ^2 que obtenemos es:

  $$I^*(\sigma^2) = [2.44,\ 8.20],$$

  y el intervalo de confianza para σ nos queda:

  $$I^*(\sigma) = [1.56,\ 2.86].$$

Para finalizar el ejemplo, hemos resumido los resultados del método percentil t simetrizado en la Tabla 4.7 con el resultado exacto del Ejemplo 4.1. Podemos observar que los resultados son similares a los que habíamos obtenido en la Tabla 4.3 del método percentil t, aunque el extremo superior ahora es ligeramente superior.

	Uniforme	Paramétrico	Simetrizado	Suavizado	Exacto
Extremo inferior	1.57	1.59	1.57	1.56	1.55
Extremo superior	2.81	2.71	2.80	2.86	2.61

Tabla 4.7: Valores de cada extremo del intervalo de confianza de σ de cada método (percentil t simetrizado).

Si en vez de usar la expresión explícita de $\sigma_{\widehat{\theta}^*}$ la estimamos a partir de la distribución bootstrap, encontramos los valores de la Tabla 4.8.

	Uniforme	Paramétrico	Simetrizado	Suavizado	Exacto
Extremo inferior	1.58	1.58	1.58	1.58	1.55
Extremo superior	2.77	2.75	2.76	2.76	2.61

Tabla 4.8: Valores de cada extremo del intervalo de confianza de σ de cada método (percentil t simetrizado) usando una estimación de $\sigma_{\widehat{\theta}^*}$.

En la siguiente sección estudiaremos la cobertura y longitud media de los intervalos de confianza que genera el método percentil t simetrizado.

4.3.2. Cobertura y longitud de los intervalos del método percentil *t* simetrizado

El método que seguiremos para obtener la cobertura y longitud de los intervalos de confianza es completamente análogo al que seguimos en la Sección §4.1.2: hallaremos la cobertura y longitud media de los intervalos para 1000 muestras de tamaño $n = 30$ y $B = 10^4$ réplicas bootstrap. Respecto del procedimiento seguido en la Sección §4.2.2 no hay ninguna diferencia salvo la forma de obtener los cuantiles: en este caso calcularemos la probabilidad $P(|T_t^*| \leq c^*) \geq 1 - \alpha$ y haremos $c_1^* = -c^*$ y $c_2^* = c^*$ (ver el Código A.32).

Hemos recogido los resultados de la cobertura y longitud de los intervalos del método t simetrizado en la Tabla 4.9. Como podemos observar, los resultados son muy similares a los del método percentil t (ver Tabla 4.5) tanto en la cobertura como en la longitud, si bien es cierto que la longitud de los intervalos del método t simetrizado es ligeramente superior. Esta conclusión era lo que esperábamos del método percentil t simetrizado puesto que no hemos

introducido una mejora sustancial con respecto del método percentil t, tan solo la forma de hallar los cuantiles.

	Uniforme	Paramétrico	Simetrizado	Suavizado	Exacto
Cobertura	0.930	0.955	0.931	0.947	0.956
Longitud	1.130	1.206	1.127	1.216	1.144

Tabla 4.9: Cobertura y longitud (media) de los intervalos de confianza de σ de cada método.

Usando una estimación de $\sigma_{\widehat{\theta}^*}$ en vez de la expresión explícita de la Ecuación (4.10) tenemos los resultados de la Tabla 4.10. De nuevo, encontramos que la cobertura y la longitud media de los intervalos es mayor que usando la expresión explícita de $\sigma_{\widehat{\theta}^*}$.

	Uniforme	Paramétrico	Simetrizado	Suavizado	Exacto
Cobertura	0.957	0.959	0.958	0.958	0.956
Longitud	1.284	1.264	1.280	1.284	1.144

Tabla 4.10: Cobertura y longitud (media) de los intervalos de confianza de σ de cada método usando una estimación de $\sigma_{\widehat{\theta}^*}$.

4.4. Resumen de la cobertura y longitud de cada método

En esta sección resumimos en la Tabla 4.11 la cobertura y longitud esperada de los intervalos de confianza que hemos obtenido en las secciones anteriores. Podemos observar que la cobertura y la longitud esperada de los métodos t y t simetrizado son más altas que las del método básico, como podíamos esperar puesto que estamos incluyendo más información a la hora de realizar el remuestreo (conocemos la expresión explícita de $\sigma_{\widehat{\theta}}$).

		Uniforme	Paramétrico	Simetrizado	Suavizado	Exacto
Básico	Cobertura	0.894	0.909	0.890	0.891	0.956
	Longitud	0.998	1.087	1.003	1.089	1.144
t	Cobertura	0.930	0.954	0.931	0.945	0.956
	Longitud	1.114	1.143	1.100	1.096	1.144
t simetrizado	Cobertura	0.930	0.955	0.931	0.947	0.956
	Longitud	1.130	1.206	1.127	1.216	1.144

Tabla 4.11: Cobertura y longitud (media) de los intervalos de confianza de σ de cada método bootstrap para el método percentil básico, t y t simetrizado.

Capítulo 5

El método bootstrap para otras familias paramétricas

En esta sección estudiaremos otros ejemplos relacionados con el cálculo de intervalos de confianza bootstrap para otras variables poblacionales y estimadores.

5.1. Bootstrap para la media de una normal

Supongamos que se han registrado 50 observaciones de una variable aleatoria que sigue una distribución normal de media 0 (desconocido) y desviación típica 1 (desconocido), y queremos hallar un intervalo de confianza con nivel de confianza 0.9 para la media μ usando el método percentil t.

Vamos a resolver esta situación usando los cuatro tipos de bootstrap que conocemos. Supongamos entonces que $\vec{X} = (X_1, \ldots, X_n)$ es una muestra aleatoria a partir de una variable aleatoria $X \sim \mathcal{N}(\mu, \sigma)$, en este caso el estadístico que vamos a utilizar es:

$$\widehat{\theta} = \overline{X},$$

y como parámetro a estimar $\theta = \mu$, entonces el estadístico asociado al método percentil básico es:

$$T_t = \frac{\widehat{\theta} - \theta}{\sigma_{\widehat{\theta}}} = \frac{\overline{X} - \mu}{\sigma_{\overline{X}}}, \tag{5.1}$$

lo que nos deja que el estadístico bootstrap asociado a la b-ésima réplica sea:

$$T_t^* = \frac{\widehat{\theta}^* - \widehat{\theta}}{\sigma_{\widehat{\theta}^*}} = \frac{\overline{X}^{*(b)} - \overline{X}}{\sigma_{\overline{X}^{*(b)}}}, \tag{5.2}$$

con $b \in \{1, \ldots, B\}$.

Para hallar $\sigma_{\overline{X}}$ usaremos el hecho de que $X \sim \mathcal{N}(\mu, \sigma)$, y entonces:

$$\overline{X} \sim \mathcal{N}(\mu, \sigma/\sqrt{n}) \Rightarrow \sigma_{\overline{X}} = \frac{\sigma}{\sqrt{n}}.$$

Ya que σ es desconocido, vamos a usar el hecho de que por el Teorema de Fisher $\overline{X}$ y $\widehat{S}_X^2$ son independientes y satisfacen:

$$\sqrt{n}\frac{\overline{X} - \mu}{\sigma} \sim \mathcal{N}(0, 1), \quad \frac{n-1}{\sigma^2}\widehat{S}_X^2 \sim \chi_{n-1}^2,$$

debido a que $X \sim \mathcal{N}(\mu, \sigma)$. Entones podemos construir el siguiente estadístico:

$$\sqrt{n}\frac{\overline{X} - \mu}{\sigma}\sqrt{\frac{\sigma^2(n-1)}{(n-1)\widehat{S}_X^2}} = \sqrt{n}\frac{\overline{X} - \mu}{\widehat{S}_X} \sim t_{n-1}.$$

Por lo que podemos modificar la Ecuación (5.2) utilizando el estadístico que acabamos de encontrar:

$$T_t^* = \sqrt{n}\frac{\overline{X}^{*(b)} - \overline{X}}{\widehat{S}_X^{(b)}},$$

con $b \in \{1, \ldots, B\}$. Entonces podemos utilizar la Ecuación (4.3) para escribir el intervalo de confianza bootstrap para μ:

$$I^*(\mu) = \left[\overline{X} - \frac{c_2^*\,\widehat{S}_X}{\sqrt{n}}, \overline{X} - \frac{c_1^*\,\widehat{S}_X}{\sqrt{n}}\right], \tag{5.3}$$

siendo c_1^*, c_2^* los cuantiles de T_t^* asociados a las probabilidades $\alpha/2, 1-\alpha/2$ respectivamente. De igual forma, para calcular el intervalo de confianza para μ de forma teórica reescribimos el estadístico de la Ecuación (5.1):

$$T_t = \sqrt{n}\frac{\overline{X} - \mu}{\widehat{S}_X} \sim t_{n-1},$$

y así podemos hallar los cuantiles d_1, d_2 de una distribución t_{n-1} asociados a las probabilidades $\alpha/2, 1-\alpha/2$, de lo que podemos concluir:

$$I(\mu) = \left[\overline{X} - \frac{d_2\,\widehat{S}_X}{\sqrt{n}}, \overline{X} - \frac{d_1\,\widehat{S}_X}{\sqrt{n}}\right]. \tag{5.4}$$

Podemos seguir los siguientes pasos para hallar los intervalos de confianza deseados (ver el Código A.33).

(1) Generamos una muestra aleatoria a partir de una normal con los parámetros del enunciado (en R usamos `rnorm`), y de ella calculamos la media muestral (con el comando `mean` en R), que resulta ser $\overline{x} = 0.10$, y la raíz de la cuasivarianza muestral (con `sd` en R), que es $\widehat{s}_X = 0.83$. Además, el valor del estimador máximo verosímil de σ para la muestra es $\widehat{\sigma}_{MV}(\vec{x}) = 0.82$.

(2) Creamos el vector que contiene la muestra simetrizada $\vec{Y}$ que habíamos definido en la Ecuación (3.12), y también hallamos el parámetro de suavizado h con la Ecuación (3.17). Notemos que el estimador máximo verosímil de μ es $\overline{X}$ y el de σ es $\widehat{\sigma}_{MV}$, tal y como sabíamos por la Ecuación (3.3).

(3) Creamos cuatro remuestras de tamaño $n = 50$: la primera a partir de la muestra inicial del punto (1) (en R con la instrucción `sample` con reemplazamiento), la segunda a partir de una normal de media $\overline{x} = 0.1$ y desviación típica $\widehat{\sigma}_{MV}(\vec{x}) = 0.82$ (la función `rnorm` nos permite realizar este paso). En tercer lugar, extraemos una remuestra de tamaño n a partir de la muestra simetrizada que teníamos del punto (2) (de nuevo, con el comando `sample` con reemplazamiento), y por último construimos una remuestra de tamaño n de la muestra inicial y la perturbamos añadiendo el término $h/2\mathcal{N}(0,1)$, con el valor de h del punto (2) (usando `sample` y `rnorm` respectivamente).

(4) Calculamos el estadístico T_t^* que teníamos en la Ecuación (5.2) para cada remuestra y almacenamos estos valores.

(5) Repetimos los puntos (3) y (4) para $b \in \{1, \ldots, B\}$, siendo $B = 10^5$. En R usamos la función `replicate`.

(6) Hallamos los cuantiles c_1^*, c_2^* de cada estadístico que obtenemos del punto (5) (con la función `quantile` en R), y con la Ecuación (5.3) encontramos los intervalos de confianza bootstrap.

(7) Por último, el intervalo de confianza analítico lo calculamos a partir de la Ecuación (5.4) (los cuantiles d_1, d_2 se pueden hallar con la función `qt` en R).

Podemos observar los resultados en la Tabla 5.1. Una diferencia notable es que los intervalos son más pequeños que los que encontrábamos en secciones anteriores. Esto es debido a

que por una parte estamos estimando un parámetro totalmente distinto, y por otra hemos disminuido el nivel de confianza, por lo que ahora esperamos que haya menos intervalos que contengan el verdadero valor de μ. Además, hemos aumentado el tamaño muestral, lo que provoca que los intervalos se parezcan más al exacto (como vimos en la Figura 2.4 aumentar el tamaño muestral aumenta la precisión del método bootstrap).

	Uniforme	Paramétrico	Simetrizado	Suavizado	Exacto
Extremo inferior	-0.11	-0.10	-0.10	-0.11	-0.10
Extremo superior	0.29	0.30	0.30	0.29	0.30

Tabla 5.1: Valores de cada extremo del intervalo de confianza de μ de cada método (percentil t).

Hagamos ahora un pequeño estudio de la cobertura y longitud de los intervalos para μ. Este estudio va a consistir en generar 1000 muestras de tamaño $n = 50$ a partir de una normal $\mathcal{N}(0, 1)$ (supondremos desviación típica desconocida) y $B = 10^3$ réplicas bootstrap para cada muestra y comparar los resultados. Realizaremos los cálculos para un nivel de confianza $1 - \alpha = 0.9$.

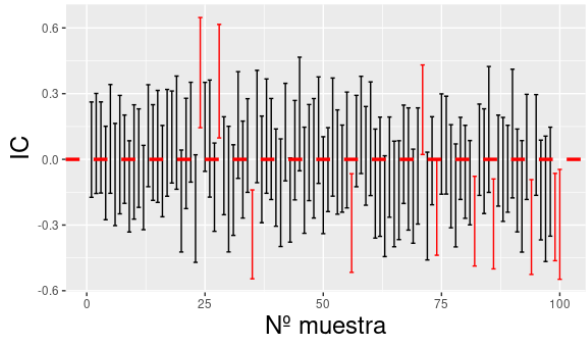
Los pasos que vamos a seguir son muy sencillos: reutilizando la función que habíamos usado en la Sección §4.2.2, vamos a cambiar el estadístico T_t^* definido en la función por el que tenemos en la Ecuación (5.2), así como también los intervalos de confianza ahora serán los de la Ecuación (5.3). Además, para hallar el intervalo de confianza teórico, vamos a utilizar la Ecuación (5.4) (ver el Código A.34).

Hemos resumido los resultados en la Tabla 5.2. Tal y como esperábamos, la cobertura de todos los métodos es muy alta debido a que ahora el tamaño muestral es mayor y el método bootstrap tiene más información. Es por ello que también la cobertura y la longitud son muy similares a las del método exacto ($n = 50$ es un tamaño relativamente grande).

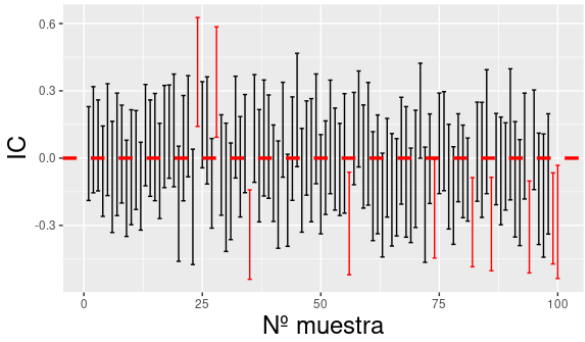
	Uniforme	Paramétrico	Simetrizado	Suavizado	Exacto
Cobertura	0.900	0.902	0.904	0.900	0.901
Longitud	0.469	0.469	0.469	0.470	0.471

Tabla 5.2: Cobertura y longitud (media) de los intervalos de confianza de μ de cada método.

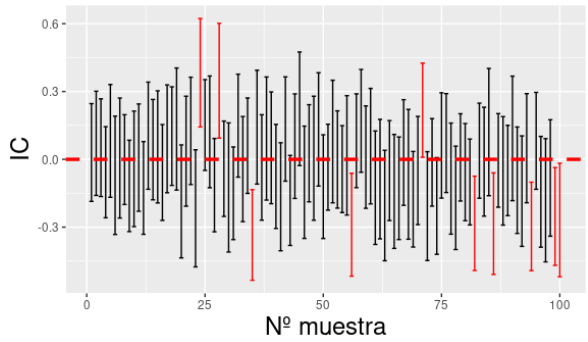
En la Figura 5.1 podemos observar la representación de los 100 primeros intervalos de confianza que hemos obtenido con cada método. En cada gráfica hemos representado con una línea discontinua roja el verdadero valor de μ y con una barra roja aquellos intervalos que no cubren dicho verdadero valor (ver el Código A.35).



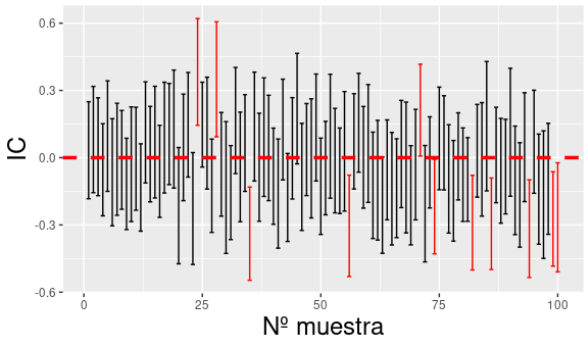
(a) Intervalos de confianza para uniforme.



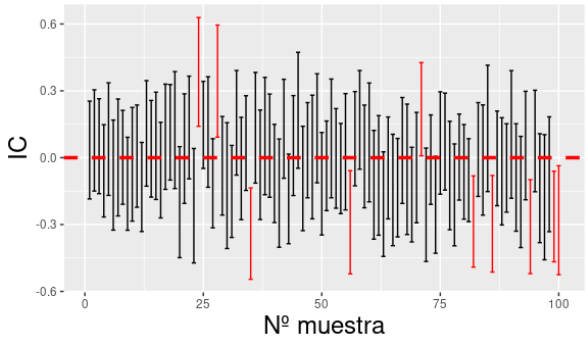
(b) Intervalos de confianza para paramétrico.



(c) Intervalos de confianza para simetrizado.



(d) Intervalos de confianza para suavizado.



(e) Intervalos de confianza para analítico.

Figura 5.1: Intervalos de confianza de μ para las 100 primeras muestras.

Para finalizar, vamos a repetir el estudio de la cobertura y longitud de 1000 muestras variando el tamaño muestral para observar las diferencias que se producen en la cobertura y

longitud de los intervalos de confianza bootstrap obtenidos por el método percentil t. En este caso hemos elegido $n \in \{10, 30, 50, 100, 150\}$. Los resultados pueden verse en la Tabla 5.3. Observamos que la longitud media de los intervalos va disminuyendo al aumentar el tamaño muestral (es decir, los intervalos se hacen más estrechos y por lo tanto el valor de μ se encuentra más localizado), mientras que la cobertura de dichos intervalos se mantiene en torno a 0.9 al aumentar n. Además, para valores pequeños de n vemos que los resultados de la cobertura son ligeramente más pequeños que para valores superiores de n.

		Uniforme	Paramétrico	Simetrizado	Suavizado	Exacto
$n = 10$	Cobertura	0.881	0.888	0.892	0.885	0.889
	Longitud	1.177	1.124	1.126	1.171	1.127
$n = 30$	Cobertura	0.909	0.908	0.903	0.907	0.909
	Longitud	0.614	0.611	0.611	0.613	0.613
$n = 50$	Cobertura	0.900	0.902	0.904	0.900	0.901
	Longitud	0.469	0.469	0.469	0.470	0.471
$n = 100$	Cobertura	0.913	0.909	0.908	0.910	0.912
	Longitud	0.330	0.330	0.329	0.330	0.330
$n = 150$	Cobertura	0.910	0.915	0.910	0.910	0.913
	Longitud	0.270	0.270	0.270	0.270	0.270

Tabla 5.3: Cobertura y longitud (media) de los intervalos de confianza de μ de cada método bootstrap para $n \in \{10, 30, 50, 100, 150\}$.

5.2. Bootstrap para el parámetro de una exponencial

Supongamos que se han registrado 30 observaciones de una variable aleatoria que sigue una distribución exponencial de parámetro $\lambda = 0.1$ (desconocido), y queremos hallar un intervalo de confianza con nivel 0.9 para λ usando el método percentil básico.

Notemos que la distribución exponencial es paramétrica y continua, pero no es simétrica y por lo tanto al aplicar el método bootstrap simetrizado debería dar resultados erróneos (o por lo menos peores que el resto de métodos).

Puesto que vamos a calcular el intervalo de confianza con el método percentil básico, ne-

cesitamos en primer lugar un estimador de λ, así que calculemos por ejemplo el estimador máximo verosímil de λ. Sea $\vec{X} = (X_1, \ldots, X_n)$ una muestra aleatoria a partir de una variable aleatoria $X \sim \exp(\lambda)$, entonces su función de verosimilitud está dada por:

$$L(\vec{x}, \lambda) = \lambda^n \exp\left(-\lambda \sum_{i=1}^{n} x_i\right) \mathbb{1}_{(0,\infty)^n}(x_1, \ldots, x_n). \tag{5.5}$$

Asumiendo que los valores $(x_1, \ldots, x_n) \in (0, \infty)^n$ (notemos que por ser $\vec{X}$ m.a.s. a partir de X siempre tendremos la condición anterior), podemos tomar la función indicatriz $\mathbb{1}$ igual a 1. Así, tomando el logaritmo natural de la Ecuación (5.5) obtenemos:

$$\ln(L(\vec{x}, \lambda)) = n \ln(\lambda) - \lambda \sum_{i=1}^{n} x_i. \tag{5.6}$$

Derivando con respecto de λ la Ecuación (5.6) e igualando a cero obtenemos el estimador máximo verosímil:

$$\frac{\partial \ln(L)}{\partial \lambda} = \frac{n}{\lambda} - \sum_{i=1}^{n} x_i = 0 \Rightarrow \widehat{\lambda}_{MV} = \frac{1}{\overline{X}}. \tag{5.7}$$

Derivando dos veces con respecto de λ la Ecuación (5.6) y evaluando en el estimador máximo verosímil encontramos que $\widehat{\lambda}_{MV}$ es efectivamente un máximo de L:

$$\left.\frac{\partial^2 \ln(L)}{\partial \lambda^2}\right|_{\lambda=\widehat{\lambda}_{MV}} = \frac{-n}{\widehat{\lambda}_{MV}^2} < 0.$$

Ahora que ya tenemos nuestro estimador $\widehat{\theta} = \widehat{\lambda}_{MV}$ el siguiente paso es construir el estadístico T_{bas} de la Ecuación (4.1):

$$T_{\text{bas}} = \widehat{\theta} - \theta = \frac{1}{\overline{X}} - \lambda,$$

y su versión bootstrap:

$$T^*_{\text{bas}} = \widehat{\theta}^* - \widehat{\theta} = \frac{1}{\overline{X}^{*(b)}} - \frac{1}{\overline{X}}, \tag{5.8}$$

donde $b \in \{1, \ldots, B\}$. Podemos aplicar directamente la Ecuación (4.3) para obtener el intervalo de confianza bootstrap:

$$I^*(\lambda) = \left[\frac{1}{\overline{X}} - c_2^*, \frac{1}{\overline{X}} - c_1^*\right], \tag{5.9}$$

siendo c_1^*, c_2^* los cuantiles de T^*_{bas} asociados a las probabilidades $\alpha/2, 1-\alpha/2$ respectivamente.

El intervalo teórico con el que compararemos se puede hallar usando diferentes propiedades de las distribuciones exponencial y γ. Sea $(Y_1, \ldots, Y_n)$ una muestra aleatoria simple a partir de $Y \sim \exp(\nu)$, se verifican dos propiedades:

$$\sum_{i=1}^{n} Y_i \sim \gamma(n, \nu), \quad \nu\overline{Y} = \frac{\nu}{n}\sum_{i=1}^{n} Y_i \sim \gamma(n, n). \tag{5.10}$$

Aplicando la Ecuación (5.10) a nuestra muestra $\vec{X}$, podemos hallar los cuantiles d_1, d_2 de $\lambda\overline{X} \sim \gamma(n, n)$ asociados a las probabilidades $\alpha/2, 1-\alpha/2$ respectivamente y construir un intervalo de confianza centrado en λ:

$$I(\lambda) = \left[\frac{d_1}{\overline{X}}, \frac{d_2}{\overline{X}}\right]. \tag{5.11}$$

Podemos seguir los siguientes pasos para hallar los intervalos de confianza deseados (ver el Código A.36).

(1) Generamos una muestra aleatoria a partir de una exponencial con los parámetros del enunciado (en R usamos `rexp`), y de ella calculamos el inverso la media muestral (con el comando `mean` en R), que resulta ser $\widehat{\lambda}_{MV}(\vec{x}) = 1/\overline{x} = 0.09$.

(2) Creamos el vector que contiene la muestra simetrizada $\vec{Y}$ que habíamos definido en la Ecuación (3.12), y también hallamos el parámetro de suavizado h con la Ecuación (3.17). Notemos que el estimador máximo verosímil de λ es $1/\overline{X}$, tal y como sabíamos por la Ecuación (5.7).

(3) Creamos cuatro remuestras de tamaño $n = 30$: la primera a partir de la muestra inicial del punto (1) (en R con la instrucción `sample` con reemplazamiento), la segunda a partir de una exponencial de parámetro $1/\overline{x} = 0.09$ (con la función `rexp` en R). En tercer lugar, extraemos una remuestra de tamaño n a partir de la muestra simetrizada que teníamos del punto (2) (de nuevo, con el comando `sample` con reemplazamiento), y por último creamos una remuestra de tamaño n de la muestra inicial y la perturbamos añadiendo el término $h/2\mathcal{N}(0, 1)$, con el valor de h del punto (2) (usando `sample` y `rnorm` respectivamente).

(4) Calculamos el estadístico T^*_{bas} que teníamos en la Ecuación (5.8) para cada remuestra y almacenamos estos valores.

(5) Repetimos los puntos (3) y (4) para $b \in \{1, \ldots, B\}$, siendo $B = 10^5$. En R usamos la función `replicate`.

(6) Hallamos los cuantiles c_1^*, c_2^* de cada estadístico que obtenemos del punto (5) (con la función `quantile` en R), y con la Ecuación (5.9) encontramos los intervalos de confianza bootstrap.

(7) Por último, el intervalo de confianza analítico lo calculamos a partir de la Ecuación (5.11) (los cuantiles d_1, d_2 a partir de una distribución $\gamma(n, n)$ se pueden hallar con la función `qgamma` en R).

Hemos expuesto los resultados en la Tabla 5.4. En principio, no vemos que haya mucha diferencia entre el método simetrizado y el resto, aunque esto puede deberse a la elección la muestra inicial.

	Uniforme	Paramétrico	Simetrizado	Suavizado	Exacto
Extremo inferior	0.058	0.056	0.054	0.057	0.066
Extremo superior	0.114	0.114	0.113	0.114	0.121

Tabla 5.4: Valores de cada extremo del intervalo de confianza de λ de cada método (percentil básico).

Para ver si el método simetrizado rinde peor que el resto vamos a realizar ahora un estudio de la cobertura y longitud de los intervalos para λ. Va a consistir en generar 1000 muestras de tamaño $n = 30$ a partir de una exponencial $\exp(\lambda = 0.1)$ y $B = 10^3$ réplicas bootstrap para cada muestra y comparar los resultados. Elegiremos un nivel de confianza $1 - \alpha = 0.9$.

Los pasos que vamos a seguir son los siguientes: de acuerdo a cómo programamos la función del punto (1) en la Sección §4.1.2 solo tendremos que pasar como argumento $\widehat{\theta} = 1/\overline{X}$ en vez de la cuasivarianza, haciendo que los pasos a seguir sean completamente análogos cambiando $\widehat{S}_X^2$ por $1/\overline{X}$ (así como sus respectivos estadísticos bootstrap) y modificando el paramétrico de tal forma que las remuestras sean a partir de una exponencial de parámetro $\widehat{\lambda}_{MV} = 1/\overline{X}$ (ver el Código A.37).

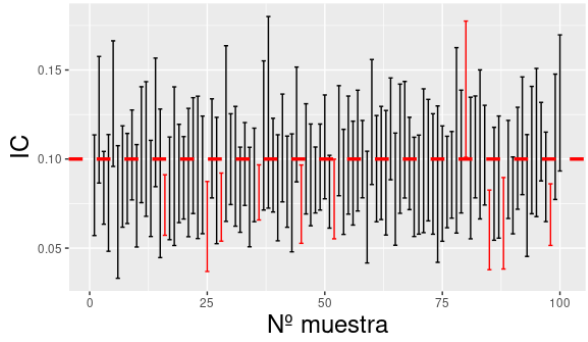
Hemos resumido los resultados en la Tabla 5.5. Ahora podemos ver que el método que peores resultados proporciona es el método simetrizado, puesto que es el que menos cobertura

tiene y la longitud media de los intervalos es superior a la del resto (salvo el paramétrico). Este comportamiento era el esperado puesto que la distribución de partida no es simétrica.

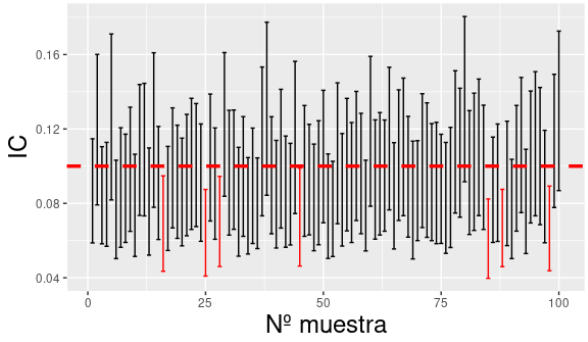
	Uniforme	Paramétrico	Simetrizado	Suavizado	Exacto
Cobertura	0.853	0.892	0.850	0.863	0.891
Longitud	0.062	0.065	0.064	0.063	0.062

Tabla 5.5: Cobertura y longitud (media) de los intervalos de confianza de λ de cada método.

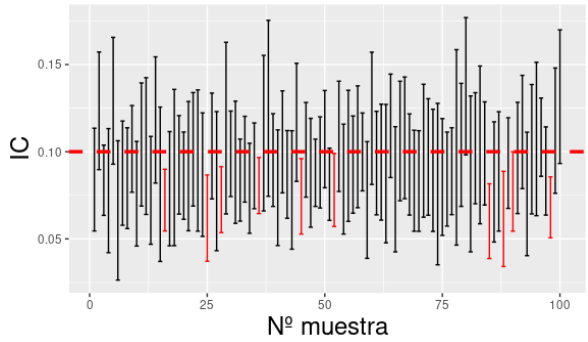
Vamos a representar los intervalos de confianza bootstrap y teóricos en la Figura 5.2 para las 100 primeras muestras. La línea roja discontinua representa el verdadero valor de λ y con una barra roja aquellos intervalos que no cubren dicho verdadero valor. Estos gráficos nos dan una idea de la cobertura de los métodos así como de la longitud esperada (ver el Código A.38).



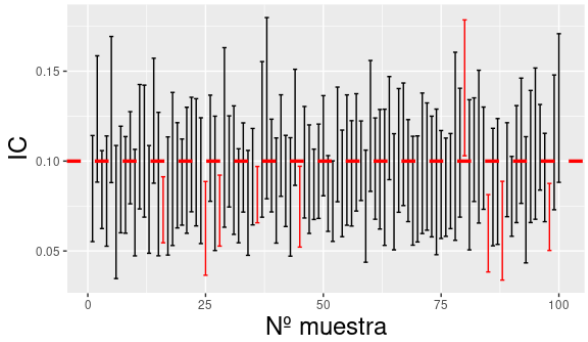
(a) Intervalos de confianza para uniforme.



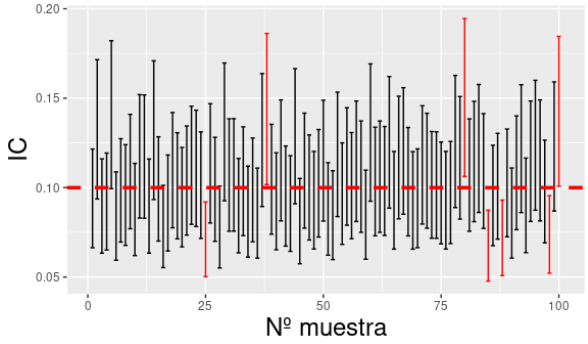
(b) Intervalos de confianza para paramétrico.



(c) Intervalos de confianza para simetrizado.



(d) Intervalos de confianza para suavizado.



(e) Intervalos de confianza para analítico.

Figura 5.2: Intervalos de confianza de λ para las 100 primeras muestras.

Para finalizar, vamos a repetir el estudio de la cobertura y longitud de 1000 muestras variando el tamaño muestral como hicimos al final de la sección anterior. Hemos elegido $n \in \{10, 30, 50, 100, 150\}$ de nuevo. Hemos resumido los resultados en la Tabla 5.6.

		Uniforme	Paramétrico	Simetrizado	Suavizado	Exacto
$n = 10$	Cobertura	0.797	0.847	0.799	0.808	0.902
	Longitud	0.121	0.131	0.132	0.126	0.112
$n = 30$	Cobertura	0.853	0.892	0.850	0.863	0.891
	Longitud	0.062	0.065	0.064	0.063	0.062
$n = 50$	Cobertura	0.895	0.908	0.891	0.896	0.919
	Longitud	0.047	0.049	0.048	0.048	0.047
$n = 100$	Cobertura	0.880	0.889	0.879	0.889	0.896
	Longitud	0.033	0.034	0.033	0.033	0.033
$n = 150$	Cobertura	0.881	0.890	0.881	0.887	0.895
	Longitud	0.027	0.027	0.027	0.027	0.027

Tabla 5.6: Cobertura y longitud (media) de los intervalos de confianza de λ de cada método bootstrap para $n \in \{10, 30, 50, 100, 150\}$.

De forma similar a los resultados de la Tabla 5.3, la longitud de los intervalos se va acortando a medida que aumenta n y llega a ser igual que la longitud esperada que proporcionan los intervalos de confianza analíticos. En este caso, las coberturas de todos los métodos se encuentran por debajo de 0.9 aunque aumentemos n, pero puede deberse a que algunas de

las muestras seleccionadas no representan bien a la población. Un apunte interesante es que los intervalos de confianza bootstrap que obtenemos tienen una longitud mayor o igual al método exacto pero, aun así, los intervalos bootstrap tienen menos cobertura.

5.3. Bootstrap para el parámetro de una Poisson

Supongamos que se han registrado 35 observaciones de una variable aleatoria que sigue una distribución de Poisson de parámetro $\lambda = 2$ (desconocido), y buscamos hallar un intervalo de confianza con nivel 0.9 para λ usando el método percentil básico.

Debemos recordar que la distribución de Poisson es paramétrica, discreta y no es simétrica, por lo tanto al aplicar el método bootstrap simetrizado y suavizado deberíamos obtener resultados peores que el resto de métodos.

Para calcular el intervalo de confianza con el método percentil básico vamos a usar el estimador máximo verosímil de λ. Sea $\vec{X} = (X_1, \ldots, X_n)$ una muestra aleatoria a partir de una variable aleatoria $X \sim \mathcal{P}(\lambda)$, entonces su función de verosimilitud está dada por:

$$L(\vec{x}, \lambda) = \frac{\lambda^{x_1+\ldots+x_n} e^{-n\lambda}}{x_1! \ldots x_n!}. \tag{5.12}$$

Tomando el logaritmo natural de la Ecuación (5.12) obtenemos:

$$\ln(L(\vec{x}, \lambda)) = -n\lambda + \ln(\lambda) \sum_{i=1}^{n} x_i - \sum_{i=1}^{n} \ln(x_i!). \tag{5.13}$$

Derivando con respecto de λ la Ecuación (5.13) e igualando a cero obtenemos el estimador máximo verosímil de λ:

$$\frac{\partial \ln(L)}{\partial \lambda} = -n + \frac{1}{\lambda} \sum_{i=1}^{n} x_i = 0 \Rightarrow \widehat{\lambda}_{MV} = \overline{X}. \tag{5.14}$$

Para comprobar que $\widehat{\lambda}_{MV}$ es un máximo de L, derivamos dos veces con respecto de λ la Ecuación (5.13) y evaluamos en el estimador máximo verosímil:

$$\left.\frac{\partial^2 \ln(L)}{\partial \lambda^2}\right|_{\lambda=\widehat{\lambda}_{MV}} = \frac{-1}{\widehat{\lambda}_{MV}^2} \sum_{i=1}^{n} x_i = \frac{-1}{\widehat{\lambda}_{MV}} < 0.$$

Encontramos entonces nuestro estimador $\widehat{\theta} = \widehat{\lambda}_{MV}$, el siguiente paso es construir el estadístico T_{bas} de la Ecuación (4.1):

$$T_{\text{bas}} = \widehat{\theta} - \theta = \overline{X} - \lambda,$$

y su versión bootstrap:

$$T^*_{\text{bas}} = \widehat{\theta}^* - \widehat{\theta} = \overline{X}^{*(b)} - \overline{X}, \tag{5.15}$$

donde $b \in \{1, \ldots, B\}$. Utilizamos la Ecuación (4.3) para obtener el intervalo de confianza bootstrap:

$$I^*(\lambda) = \left[\overline{X} - c^*_2, \overline{X} - c^*_1\right], \tag{5.16}$$

siendo c^*_1, c^*_2 los cuantiles de T^*_{bas} asociados a las probabilidades $\alpha/2, 1-\alpha/2$ respectivamente. Puesto que la distribución de Poisson no tiene ningún estadístico pivote sencillo, vamos a usar una aproximación asintótica con el Teorema central del límite para comparar con los intervalos de confianza bootstrap.

Teorema 5.1 (Teorema central del límite)**.** *Sean $X_1, \ldots, X_n$ variables aleatorias independientes e idénticamente distribuidas tales que $\mathrm{E}(X_i) = \mu, \mathrm{Var}(X_i) = \sigma^2 < \infty, \forall i \in \{1, \ldots, n\}$, entonces se verifica:*

$$\sqrt{n}\frac{\overline{X} - \mu}{\sigma} \xrightarrow{\mathcal{L}} \mathcal{N}(0, 1).$$

Ya que $(X_1, \ldots, X_n)$ es una m.a.s. a partir de $X \sim \mathcal{P}(\lambda)$, se tiene que $\mathrm{E}(X_i) = \lambda$, $\mathrm{Var}(X_i) = \lambda < \infty, \forall i \in \{1, \ldots, n\}$, aplicando el Teorema central del límite encontramos:

$$\sqrt{n}\frac{\overline{X} - \lambda}{\sqrt{\lambda}} \xrightarrow{\mathcal{L}} \mathcal{N}(0, 1). \tag{5.17}$$

Para calcular un intervalo de confianza para λ tendremos que hacer uso también del denominado método delta.

Teorema 5.2 (Método delta)**.** *Sea $\{T_n\}_{n\in\mathbb{N}}$ una sucesión de estadísticos construidos a partir del muestreo de una variable aleatoria X cuya distribución depende de $\theta \in \mathbb{R}$ y tales que:*

$$\sqrt{n}(T_n - \theta) \xrightarrow{\mathcal{L}} \mathcal{N}(0, \sigma).$$

Si consideramos una función g derivable y con derivada no nula en el espacio paramétrico, entonces:

$$\sqrt{n}(g(T_n) - g(\theta)) \xrightarrow{\mathcal{L}} \mathcal{N}(0, \sigma|g'(\theta)|).$$

En este caso, el espacio paramétrico es $\Theta = (0, \infty)$ y los estadísticos son $T_n = \overline{X}$ con $n \in \mathbb{N}$. Con la Ecuación (5.17) tenemos la convergencia en ley y proponemos la función $g(x) = \sqrt{x}$ derivable y que cumple $g'(x) = 1/(2\sqrt{x}) \neq 0, \forall x \in \Theta$.

Así, a la luz del método delta:

$$\sqrt{n}\left(\sqrt{\overline{X}}-\sqrt{\lambda}\right)\xrightarrow{\mathcal{L}}\mathcal{N}\left(0,\sqrt{\lambda}\frac{1}{2\sqrt{\lambda}}\right)\Rightarrow 2\sqrt{n}\left(\sqrt{\overline{X}}-\sqrt{\lambda}\right)\xrightarrow{\mathcal{L}}\mathcal{N}\left(0,1\right). \tag{5.18}$$

Con la Ecuación (5.18) hemos encontrado un estadístico pivote asintótico para λ con el que podemos construir un intervalo de confianza:

$$I(\lambda)=\left[\left(\sqrt{\overline{X}}-\frac{d_2}{2\sqrt{n}}\right)^2,\left(\sqrt{\overline{X}}-\frac{d_1}{2\sqrt{n}}\right)^2\right], \tag{5.19}$$

siendo d_1, d_2 los cuantiles de una normal $\mathcal{N}(0,1)$ asociados a las probabilidades $\alpha/2, 1-\alpha/2$ respectivamente.

Podemos seguir los siguientes pasos para hallar los intervalos de confianza bootstrap y asintótico (ver el Código A.39).

(1) Generamos una muestra aleatoria a partir de una Poisson con los parámetros del enunciado (en R usamos `rpois`), y de ella calculamos su media muestral (con el comando `mean` en R), que resulta ser $\widehat{\lambda}_{MV}(\vec{x})=\overline{x}=2$.

(2) Creamos el vector que contiene la muestra simetrizada $\vec{Y}$ que habíamos definido en la Ecuación (3.12), y también hallamos el parámetro de suavizado h con la Ecuación (3.17). Observemos que el estimador máximo verosímil de λ es $\overline{X}$, como sabíamos por la Ecuación (5.14).

(3) Creamos cuatro remuestras de tamaño $n=35$: la primera a partir de la muestra inicial del punto (1) (en R con la instrucción `sample` con reemplazamiento), la segunda a partir de una Poisson de parámetro $\overline{x}=2$ (con la función `rpois` en R). En tercer lugar, extraemos una remuestra de tamaño n a partir de la muestra simetrizada que teníamos del punto (2) (de nuevo, con el comando `sample` con reemplazamiento), y por último construimos una remuestra de tamaño n de la muestra inicial y la perturbamos añadiendo el término $h/2\mathcal{N}(0,1)$, con el valor de h del punto (2) (usando `sample` y `rnorm` respectivamente).

(4) Calculamos el estadístico T^*_{bas} que teníamos en la Ecuación (5.15) para cada remuestra y almacenamos estos valores.

(5) Repetimos los puntos (3) y (4) para $b\in\{1,\ldots,B\}$, siendo $B=10^5$. En R usamos la función `replicate`.

(6) Hallamos los cuantiles c_1^*, c_2^* de cada estadístico que obtenemos del punto (5) (con la función `quantile` en R), y con la Ecuación (5.16) encontramos los intervalos de confianza bootstrap.

(7) Por último, el intervalo de confianza analítico lo calculamos a partir de la Ecuación (5.19) (los cuantiles d_1, d_2 a partir de una distribución $\mathcal{N}(0,1)$ se pueden hallar con la función `qnorm` en R).

Hemos expuesto los resultados en la Tabla 5.7. Para ver si realmente los métodos simetrizado y suavizado funcionan peor que el resto tendremos que hacer un estudio de cobertura y longitud, puesto que para esta muestra no hay gran diferencia en los intervalos de confianza.

	Uniforme	Paramétrico	Simetrizado	Suavizado	Asintótico
Extremo inferior	1.600	1.600	1.629	1.601	1.626
Extremo superior	2.371	2.371	2.371	2.385	2.412

Tabla 5.7: Valores de cada extremo del intervalo de confianza de λ de cada método (percentil básico).

Para el estudio de la cobertura y longitud generaremos 1000 muestras de tamaño $n = 35$ a partir de una Poisson $\mathcal{P}(\lambda = 2)$ y $B = 10^3$ réplicas bootstrap. Tomaremos un nivel de confianza $1 - \alpha = 0.9$.

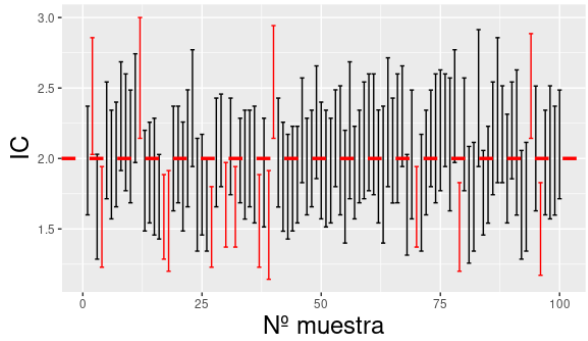
Los pasos que vamos a seguir son completamente análogos a los que seguimos en la Sección §5.2, teniendo en cuenta que $\widehat{\theta} = \overline{X}$ y que los intervalos con los que vamos a comparar son de la forma que teníamos en la Ecuación (5.19) (ver el Código A.40).

Hemos resumido los resultados en la Tabla 5.8. Podemos observar que, aunque la cobertura es elevada, los métodos simetrizado y suavizado son los que peor funcionan, siendo el suavizado el peor ya que produce intervalos tan largos como el paramétrico pero cubre menos veces el valor real de λ.

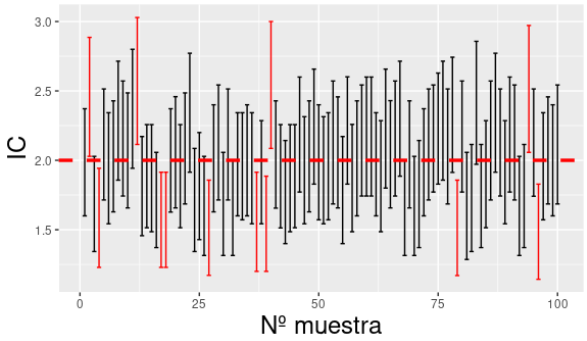
	Uniforme	Paramétrico	Simetrizado	Suavizado	Asintótico
Cobertura	0.884	0.905	0.871	0.875	0.899
Longitud	0.766	0.783	0.767	0.782	0.785

Tabla 5.8: Cobertura y longitud (media) de los intervalos de confianza de λ de cada método.

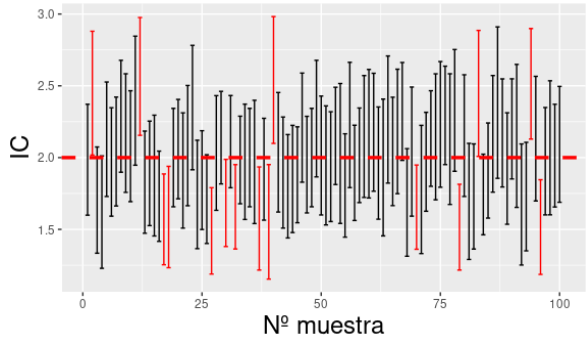
En la Figura 5.3 vemos los intervalos de confianza bootstrap y asintóticos para las 100 primeras muestras. Hemos representado con una línea roja el verdadero valor de λ y con una barra roja aquellos intervalos que no cubren dicho verdadero valor (ver el Código A.41).



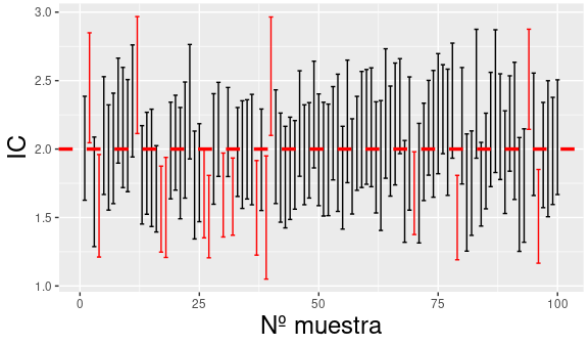
(a) Intervalos de confianza para uniforme.



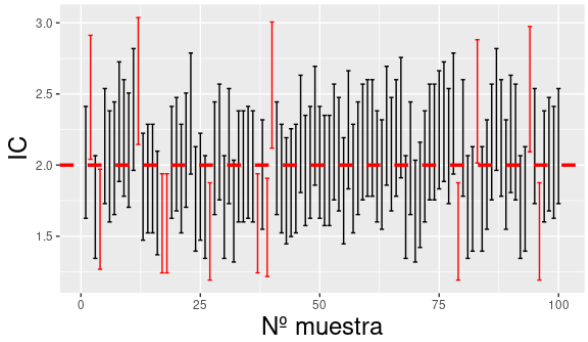
(b) Intervalos de confianza para paramétrico.



(c) Intervalos de confianza para simetrizado.



(d) Intervalos de confianza para suavizado.



(e) Intervalos de confianza para asintótico.

Figura 5.3: Intervalos de confianza de λ para las 100 primeras muestras.

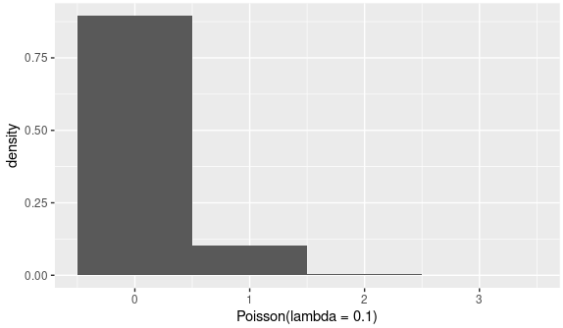
Repitamos el estudio de la cobertura y longitud de 1000 muestras variando el tamaño muestral como hicimos al final de las secciones anteriores. Hemos elegido otra vez los siguientes valores $n \in \{10, 30, 50, 100, 150\}$. Los resultados están recogidos en la Tabla 5.9.

Observamos que los resultados de cobertura son peores cuando n es bajo. Además, la longitud media de los intervalos va decreciendo al aumentar n. En general, los métodos bootstrap y el asintótico tienen resultados parecidos en cobertura y longitud, salvo el método simetrizado que tiene una cobertura ligeramente más baja a igualdad de longitud, como cabía esperar puesto que la distribución de Poisson no es simétrica. Sin embargo, el método suavizado parece tener un rendimiento similar al método uniforme en la cobertura a pesar de que la distribución de Poisson no es continua. Aun así, los intervalos que produce el método suavizado son ligeramente más largos en media que los del uniforme, lo que nos llevaría a elegir el bootstrap uniforme por encima del suavizado en este caso.

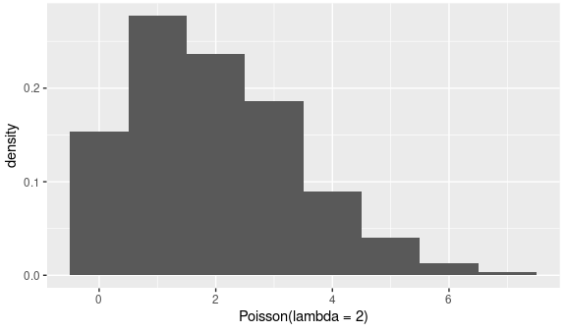
		Uniforme	Paramétrico	Simetrizado	Suavizado	Asintótico
$n = 10$	Cobertura	0.851	0.876	0.854	0.854	0.903
	Longitud	1.348	1.454	1.356	1.400	1.463
$n = 30$	Cobertura	0.895	0.909	0.894	0.893	0.913
	Longitud	0.825	0.841	0.826	0.845	0.844
$n = 50$	Cobertura	0.885	0.903	0.877	0.883	0.894
	Longitud	0.645	0.654	0.644	0.656	0.656
$n = 100$	Cobertura	0.897	0.899	0.889	0.899	0.892
	Longitud	0.460	0.464	0.460	0.467	0.465
$n = 150$	Cobertura	0.888	0.893	0.888	0.893	0.894
	Longitud	0.377	0.378	0.377	0.382	0.380

Tabla 5.9: Cobertura y longitud (media) de los intervalos de confianza de λ de cada método bootstrap para $n \in \{10, 30, 50, 100, 150\}$.

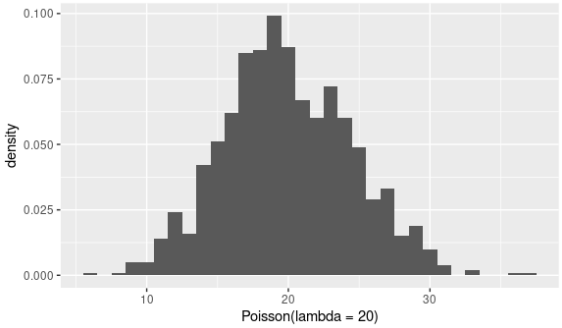
Para terminar, hagamos un estudio de la cobertura variando λ. Notemos que la forma de la distribución de Poisson depende del parámetro λ: cuando λ es pequeño la distribución es totalmente discreta, mientras que al aumentar λ se aproxima a una distribución continua, tal y como podemos ver en la Figura 5.4.



(a) Distribución de Poisson para $\lambda = 0.1$.



(b) Distribución de Poisson para $\lambda = 2$.



(c) Distribución de Poisson para $\lambda = 20$.

Figura 5.4: Distribución $\mathcal{P}(\lambda)$ variando λ.

Para la cobertura y longitud de los intervalos de confianza tomaremos 1000 muestras de tamaño $n = 35$, $B = 1000$ réplicas bootstrap y $1 - \alpha = 0.9$. Para el parámetro, escogeremos $\lambda \in \{0.1, 2, 20\}$. Los resultados pueden observarse en la Tabla 5.10. Podemos concluir que al aumentar el valor del parámetro λ tanto los métodos bootstrap como el método asintótico mejoran su cobertura (hasta estabilizarse en torno a 0.9). Vemos que la longitud también aumenta al aumentar λ debido a que la varianza de una Poisson también es λ, provocando que la dispersión de los datos crezca y que los intervalos tengan que ser más anchos para asegurar el nivel de confianza establecido.

		Uniforme	Paramétrico	Simetrizado	Suavizado	Asintótico
$\lambda = 0.1$	Cobertura	0.842	0.848	0.840	0.844	0.828
	Longitud	0.160	0.163	0.165	0.170	0.167
$\lambda = 2$	Cobertura	0.884	0.905	0.871	0.875	0.899
	Longitud	0.766	0.783	0.767	0.782	0.785
$\lambda = 20$	Cobertura	0.889	0.896	0.890	0.890	0.897
	Longitud	2.415	2.479	2.413	2.470	2.485

Tabla 5.10: Cobertura y longitud (media) de los intervalos de confianza de λ de cada método bootstrap para $\lambda \in \{0.1, 2, 20\}$.

Capítulo 6

Conclusiones

A lo largo de esta memoria hemos ido recopilando resultados y conclusiones acerca del bootstrap. Inicialmente, hemos visto que el método bootstrap depende de dos factores: lo representativa que sea la muestra y, en menor medida, del número de réplicas bootstrap que se usen. Comprobamos que cuanto más representativa es la muestra mejores son los resultados que devuelve el método bootstrap (en particular las muestras serán más representativas al aumentar el tamaño muestral n), mientras que un número elevado de réplicas bootstrap no mejora en gran medida los resultados (y el tiempo de cómputo se eleva considerablemente) pero sí es importante que sean suficientes (a partir de 1000 réplicas el método funciona relativamente bien). Además, verificamos que el método bootstrap se aproxima al método exacto y que además nos permite abordar problemas y obtener conclusiones incluso cuando no podemos resolver dichos problemas analíticamente.

También hemos comparado el método bootstrap uniforme con tres modificaciones del mismo: paramétrico, simetrizado y suavizado. Estas extensiones del bootstrap uniforme asumían distintas hipótesis sobre la población con el fin de mejorar el rendimiento del método bootstrap. En particular, en el paramétrico suponíamos conocida la familia paramétrica a la que la población pertenecía, en el simetrizado tomábamos como hipótesis que la población procedía de una variable aleatoria simétrica y en el suavizado que la población provenía de una variable continua. Vimos que el tipo de bootstrap que mejor funcionaba para aproximar los momentos muestrales de la cuasivarianza fue el paramétrico debido a que las hipótesis que hacemos sobre la población son muy fuertes. Seguido del paramétrico, el bootstrap suavizado lograba buenas aproximaciones con hipótesis mucho más débiles. Los métodos que peor

rendimiento tenían eran el simetrizado y el uniforme, de lo que concluíamos que el método bootstrap simetrizado no introducía una mejora considerable respecto del método uniforme.

Dentro de los tipos de bootstrap, dimos especial importancia al método suavizado. Esta técnica se basaba en estimar la función de densidad de la población de forma no paramétrica mediante el uso de un parámetro de suavizado h y un núcleo K. Con el fin de conseguir los mejores resultados posibles para el método suavizado, variamos los valores del parámetro h (dado por la Ecuación (3.17)) escalándolo por $1, 1/2, 1/3, 1/4$, dejando el mismo núcleo (gaussiano en este caso). Obtuvimos que el factor $1/2$ era el que menor error relativo producía a la hora de calcular los momentos de la cuasivarianza. Por otro lado, investigamos si realmente la elección del núcleo era poco relevante a igualdad de h. Para ello elegimos tres núcleos: gaussiano, rectangular y de Epanechnikov. Obtuvimos unos resultados que al principio nos resultaron sorprendentes: el núcleo gaussiano tenía un rendimiento mucho peor que los otros dos núcleos. Sin embargo, esto era debido a que el núcleo gaussiano tiene una varianza mayor que la de los otros dos núcleos. Utilizando el valor $h/2$ como parámetro de suavizado para el núcleo gaussiano obteníamos unos resultados muy similares a los de los núcleos rectangular y de Epanechnikov.

Asimismo, introdujimos tres formas de calcular intervalos de confianza bootstrap: percentil básico, t y t simetrizado, siendo el método t simetrizado una ligera variación del t a la hora de escoger los cuantiles. Utilizamos estos métodos para hallar intervalos de confianza para σ. De estos tres, el t y t simetrizado daban los mejores resultados de cobertura y longitud, asemejándose mucho a los que devolvía el método exacto. No obstante, para usar el método t necesitamos una expresión para la desviación típica del estimador que usamos al construir el intervalo de confianza, lo cual en la práctica no siempre es posible. En cualquier caso, el método básico daba unos resultados de cobertura y longitud aceptables, por lo que, en caso de no tener información sobre la desviación típica del estimador, es una buena opción para calcular intervalos de confianza.

Finalmente, usamos el método bootstrap para hallar intervalos de confianza para otras familias paramétricas y otros estimadores, como pueden ser la exponencial o la Poisson y la media muestral o su inverso. En estas situaciones, con el bootstrap hemos obtenido rendimientos de cobertura y longitud similares a los métodos analíticos y asintóticos, sobre todo cuando aumentábamos el tamaño muestral n. Estas nuevas distribuciones tenían propieda-

des distintas (no son simétricas y la Poisson no es continua) a la normal (que es simétrica y continua) lo que provocaba que los métodos simetrizado y suavizado funcionasen peor, especialmente el método simetrizado.

De este estudio que hemos realizado sobre el método bootstrap podemos concluir que es una buena herramienta para aproximar el comportamiento de estimadores cuando la distribución de la población subyacente es desconocida. Con dicha aproximación, podemos hallar sesgos, varianzas e incluso intervalos de confianza cuando no podemos aplicar el método exacto. Además, el método bootstrap no se reduce solo a lo que hemos recogido en esta memoria. El bootstrap se utiliza también en contrastes de hipótesis, como tests de simetría [15] o de dominancia estocástica [1]. Igualmente, el bootstrap tiene aplicaciones fuera del ámbito teórico de la estadística, por ejemplo en econometría [7] (en el contexto del análisis de las series temporales) o en el análisis de procesamiento de señales [16]. De hecho, el propio Efron usó el método bootstrap para el estudio del leptón τ en física de partículas [8].
En definitiva, el método bootstrap es una técnica útil no solo en el contexto de la matemática, sino que se extiende a cualquier ámbito de estudio en el que se requiera de herramientas estadísticas para obtener resultados confiables.

Referencias

[1] Álvarez-Esteban, P. C., del Barrio, E., Cuesta-Albertos, J. A., y Matrán, C. Models for the assessment of treatment improvement: The ideal and the feasible. *Statistical Science 32*, 3 (2017), 469 – 485.

[2] Cao Abad, R., y Fernández Casal, R. *Técnicas de Remuestreo.* https://rubenfcasal. github. io/book_remuestreo, Madrid, España, 2020.

[3] Efron, B. Bootstrap methods: Another look at the jackknife. *The Annals of Statistics 7*, 1 (1979), 1 – 26.

[4] Efron, B. Better bootstrap confidence intervals. *Journal of the American Statistical Association 82*, 397 (1987), 171–185.

[5] Efron, B., y Tibshirani, R. J. *An introduction to the bootstrap.* CRC press, Boca Ratón, EEUU, 1994.

[6] Epanechnikov, V. A. Non-parametric estimation of a multivariate probability density. *Theory of Probability & Its Applications 14*, 1 (1969), 153 – 158.

[7] Hacker, S., y Hatemi-J, A. A bootstrap test for causality with endogenous lag length choice: theory and application in finance. *Journal of Economic Studies 39*, 2 (2012), 144–160.

[8] Hayes, K. G., Perl, M. L., y Efron, B. Application of the bootstrap statistical method to the tau-decay-mode problem. *Physical Review D 39* (1989), 274–279.

[9] Parzen, E. On estimation of a probability density function and mode. *The Annals of Mathematical Statistics 33*, 3 (1962), 1065 – 1076.

[10] Quenouille, M. H. Approximate tests of correlation in time-series. *Mathematical Proceedings of the Cambridge Philosophical Society 45*, 3 (1949), 483–484.

[11] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Viena, Austria, 2021.

[12] Rosenblatt, M. Remarks on some nonparametric estimates of a density function. *The Annals of Mathematical Statistics 27*, 3 (1956), 832 – 837.

[13] Rubin, D. B. The bayesian bootstrap. *The Annals of Statistics 9*, 1 (1981), 130–134.

[14] Silverman, B. W. *Density estimation for statistics and data analysis*, vol. 26. CRC press, Boca Ratón, EEUU, 1986.

[15] Zheng, T., y Gastwirth, J. L. On bootstrap tests of symmetry about an unknown median. *Journal of Data Science 8*, 3 (2010), 413–427.

[16] Zoubir, A., y Boashash, B. The bootstrap and its application in signal processing. *IEEE Signal Processing Magazine 15*, 1 (1998), 56–76.

Apéndice A

Códigos utilizados

En este apéndice podemos encontrar los códigos que hemos ido realizando a lo largo de este trabajo. En caso de querer utilizarlos, recomendamos usar el visor de documentos Adobe Acrobat Reader y seleccionar el texto pulsando la tecla `Alt` (se debe prestar especial atención cuando se copien líneas que contengan comillas `""`).

- El siguiente código fue empleado aquí:

```
# Bootstrap para aproximar la función de distribución (dos muestras)
library(tidyverse)

set.seed(1)
muestra1 <- rnorm(25,17.4,2.1)
muestra2 <- rnorm(25,17.4,2.1)

n <- length(muestra1)
S2_1 <- var(muestra1)
S2_2 <- var(muestra2)

# Aplicamos el método bootstrap
B <- 100000 # Número de réplicas bootstrap

T_boot <- replicate(B,{
  remuestra1 <- sample(muestra1,n,replace = TRUE)
  S2_boot1 <- var(remuestra1)

  remuestra2 <- sample(muestra2,n,replace = TRUE)
```

```
  S2_boot2 <- var(remuestra2)
  c((n - 1)/S2_1*S2_boot1,(n - 1)/S2_2*S2_boot2)
})

p1 <- ggplot(data = tibble(T_boot[1,]),aes(x = T_boot[1,])) +
  geom_histogram(aes(y = ..density..),binwidth = 1) +
  stat_function(fun = dchisq,args = n - 1,colour = "red",size = 1.2,aes(color =
    "dchisq")) +
  labs(title = "Distribución de T* (muestra 1)") + xlab("") + ylab("T*") +
  scale_color_manual("",values = c(dchisq = "red")) +
  theme(legend.position = c(0.85,0.8),
        legend.background = element_rect(fill = "transparent"),
        legend.text = element_text(face = "bold",size = 10),
        plot.title = element_text(size = 18,face = "bold"),
        axis.title.y = element_text(size = 18))
p1

p2 <- ggplot(data = tibble(T_boot[2,]),aes(x = T_boot[2,])) +
  geom_histogram(aes(y = ..density..),binwidth = 1) +
  stat_function(fun = dchisq,args = n - 1,colour = "red",size = 1.2,aes(color =
    "dchisq")) +
  labs(title = "Distribución de T* (muestra 2)") + xlab("") + ylab("T*") +
  scale_color_manual("",values = c(dchisq = "red")) +
  theme(legend.position = c(0.85,0.8),
        legend.background = element_rect(fill = "transparent"),
        legend.text = element_text(face = "bold",size = 10),
        plot.title = element_text(size = 18,face = "bold"),
        axis.title.y = element_text(size = 18))
p2
```

Código A.1: Código utilizado para la representación de la Figura 2.1.

- El siguiente código fue empleado aquí:

```
for (i in 1:2){
  for (j in 1:4){
    momento_analitico <- 2**j*gamma(j + (n - 1)/2)/gamma((n - 1)/2)
    print(paste0("Momento de orden ",j," de T* para la muestra ",i," : ",
    mean(T_boot[i,]**j)))
```

```
    print(paste0("Error relativo (%): ",
                 abs(mean(T_boot[i,]**j) -
    momento_analitico)/momento_analitico*100))
  }
}
```

Código A.2: Código utilizado para la obtención de los momentos (uniforme).

- El siguiente código fue empleado aquí:

```
# Bootstrap para la cuasivarianza de una muestra procedente de una exponencial
library(tidyverse)

set.seed(1)
lambda <- 2
n <- 30
muestra <- rexp(n,lambda)

S2 <- var(muestra)

# Distribución bootstrap
B <- 1e5 # Número de réplicas bootstrap
T_boot <- replicate(B,{
  remuestra <- sample(muestra,n,replace = TRUE)
  var(remuestra)
})

# Plot de la distribución
p <- ggplot(data = tibble(T_boot),aes(x = T_boot)) +
  geom_histogram(aes(y = ..density..)) +
  labs(title = "Distribución de T*") + xlab("") + ylab("T*") +
  scale_color_manual("") +
  theme(legend.position = c(0.85,0.8),
        legend.background = element_rect(fill = "transparent"),
        legend.text = element_text(face = "bold",size = 10),
        plot.title = element_text(size = 18,face = "bold"),
        axis.title.y = element_text(size = 18))
p
```

Código A.3: Código empleado para la representación de la Figura 2.6.

- El siguiente código fue empleado aquí:

```
for (j in 1:4){
  print(paste0("Momento de orden ",j," de T* para la muestra : ",mean(T_boot**j)))
}
```

Código A.4: Código utilizado para la obtención de los momentos (exponencial).

- El siguiente código fue empleado aquí:

```
# Bootstrap paramétrico para la cuasivarianza
library(tidyverse)

set.seed(1)
n <- 25
mu <- 17.4
muestra <- rnorm(n,mu,2.1)

mu_EMV <- mean(muestra)
sigma2_EMV <- 1/n*sum((muestra - mu_EMV)**2) # Estimador MV de la varianza

B <- 1e5 # Número de réplicas bootstrap

T_boot <- replicate(B,{
  remuestra <- rnorm(n,mu_EMV,sqrt(sigma2_EMV))
  S2_boot <- var(remuestra)
  (n - 1)/sigma2_EMV*S2_boot
})

p1 <- ggplot(data = tibble(T_boot),aes(x = T_boot)) +
  geom_histogram(aes(y = ..density..),binwidth = 1)+
  stat_function(fun = dchisq,args = n - 1,colour = "red",size = 1.2,aes(color =
    "dchisq")) +
  xlab("") + ylab("T*") +
  scale_color_manual("",values = c(dchisq = "red")) +
  theme(legend.position = c(0.85,0.8),
        legend.background = element_rect(fill = "transparent"),
        legend.text = element_text(face = "bold",size = 10),
        plot.title = element_text(size = 18,face = "bold"),
        axis.title.y = element_text(size = 18))
```

```
p1
```

Código A.5: Código utilizado para la representación de la Figura 3.1.

- El siguiente código fue empleado aquí:

```
for (j in 1:4){
  momento_analitico <- 2**j*gamma(j + (n - 1)/2)/gamma((n - 1)/2)
  print(paste0("Momento de orden ",j," de T* : ",mean(T_boot**j)))
  print(paste0("Error relativo (%): ",
               abs(mean(T_boot**j) - momento_analitico)/momento_analitico*100))
}
```

Código A.6: Código utilizado para la obtención de los momentos (paramétrico).

- El siguiente código fue empleado aquí:

```
# Bootstrap simétrico para la cuasivarianza
library(tidyverse)

set.seed(1)
n <- 25
mu <- 17.4
muestra <- rnorm(n,mu,2.1)

S2 <- var(muestra)

muestra_sim <- c(muestra,2*mean(muestra) - muestra)

B <- 1e5 # Número de réplicas bootstrap

T_boot <- replicate(B,{
  remuestra <- sample(muestra_sim,n,replace = TRUE)
  S2_boot <- var(remuestra)
  (n - 1)/S2*S2_boot
})

p1 <- ggplot(data = tibble(T_boot),aes(x = T_boot)) +
  geom_histogram(aes(y = ..density..),binwidth = 1)+
```

```
  stat_function(fun = dchisq,args = n - 1,colour = "red",size = 1.2,aes(color =
    "dchisq")) +
  labs(title = "Distribución de T*") + xlab("") + ylab("T*") +
  scale_color_manual("",values = c(dchisq = "red")) +
  theme(legend.position = c(0.85,0.8),
        legend.background = element_rect(fill = "transparent"),
        legend.text = element_text(face = "bold",size = 10),
        plot.title = element_text(size = 18,face = "bold"),
        axis.title.y = element_text(size = 18))
p1
```

Código A.7: Código utilizado para la representación de la Figura 3.2.

- El siguiente código fue empleado aquí:

```
for (j in 1:4){
  momento_analitico <- 2**j*gamma(j + (n - 1)/2)/gamma((n - 1)/2)
  print(paste0("Momento de orden ",j," de T* : ",mean(T_boot**j)))
  print(paste0("Error relativo (%): ",
               abs(mean(T_boot**j) - momento_analitico)/momento_analitico*100))
}
```

Código A.8: Código utilizado para la obtención de los momentos (simetrizado).

- El siguiente código fue empleado aquí:

```
# Ejemplo estimación de la función de densidad a partir de una muestra
library(tidyverse)

set.seed(1)
muestra <- rnorm(50)
anchos <- c(0.1,0.4,1,8)
for (i in anchos){
  p <- ggplot(data = tibble(muestra),aes(x = muestra)) +
    geom_histogram(aes(y = ..density..),bins = 20,alpha = 0.8) +
    geom_density(bw = i,kernel = "gaussian",color = "red",size = 0.8,linetype =
    "dashed") +
    ylim(0,0.7) + xlim(min(muestra),max(muestra)) +
    stat_function(fun = dnorm,color = "blue",linetype = "dotted",size = 0.8)

```

```
  print(p)
}
```

Código A.9: Código utilizado para la representación de la Figura 3.3.

- El siguiente código fue empleado aquí:

```
# Bootstrap suavizado para la cuasivarianza
library(tidyverse)

set.seed(1)
n <- 25
mu <- 17.4
muestra <- rnorm(n,mu,2.1)

S2 <- var(muestra)

# Parámetro de suavizado h
fh <- density(muestra)
h <- fh$bw

B <- 1e5 # Número de réplicas bootstrap

T_boot <- replicate(B,{
  remuestra <- sample(muestra,n,replace = TRUE) + h*rnorm(n)
  S2_boot <- var(remuestra)
  (n - 1)/S2*S2_boot
})

p1 <- ggplot(data = tibble(T_boot),aes(x = T_boot)) +
  geom_histogram(aes(y = after_stat(density)),binwidth = 1)+
  stat_function(fun = dchisq,args = n - 1,colour = "red",size = 1.2,aes(color =
    "dchisq")) +
  labs(title = "Distribución de T*") + xlab("") + ylab("T*") +
  scale_color_manual("",values = c(dchisq = "red")) +
  theme(legend.position = c(0.85,0.8),
        legend.background = element_rect(fill = "transparent"),
        legend.text = element_text(face = "bold",size = 10),
        plot.title = element_text(size = 18,face = "bold"))
```

```
p1
```

Código A.10: Código utilizado para la representación de la Figura 3.4.

- El siguiente código fue empleado aquí:

```
for (j in 1:4){
  momento_analitico <- 2**j*gamma(j + (n - 1)/2)/gamma((n - 1)/2)
  print(paste0("Momento de orden ",j," de T* : ",mean(T_boot**j)))
  print(paste0("Error relativo (%): ",
               abs(mean(T_boot**j) - momento_analitico)/momento_analitico*100))
}
```

Código A.11: Código utilizado para la obtención de los momentos (suavizado).

- El siguiente código fue empleado aquí:

```
# Estudio del error relativo variando el parámetro h
n_muestra <- 100
n <- 25
mu <- 17.4

set.seed(1)
muestras <- matrix(0,nrow = n,ncol = n_muestra) # Guardamos en una matriz las
                                                # muestras
for (i in 1:n_muestra){
  muestras[,i] <- rnorm(n,mu,2.1)
}

S2_vector <- apply(muestras,2,var)

fh_vector <- apply(muestras,2,density)
h_vector <- numeric(n_muestra)
for (i in 1:n_muestra){
  h_vector[i] <- fh_vector[[i]]$bw
}

B <- 1e5

T_boot1 <- matrix(0,nrow = B,ncol = n_muestra)
```

```r
T_boot2 <- matrix(0,nrow = B,ncol = n_muestra)
T_boot3 <- matrix(0,nrow = B,ncol = n_muestra)
T_boot4 <- matrix(0,nrow = B,ncol = n_muestra)

for (i in 1:n_muestra){
  T_boot <- replicate(B,{
    remuestra <- sample(muestras[,i],n,replace = TRUE)
    remuestra1 <- remuestra + h_vector[i]*rnorm(n)
    remuestra2 <- remuestra + h_vector[i]*rnorm(n)/2
    remuestra3 <- remuestra + h_vector[i]*rnorm(n)/3
    remuestra4 <- remuestra + h_vector[i]*rnorm(n)/4

    S2_boot_all <- apply(rbind(remuestra1,remuestra2,remuestra3,remuestra4),1,var)
    (n - 1)/S2_vector[i]*S2_boot_all
  })
  T_boot1[,i] <- T_boot[1,]
  T_boot2[,i] <- T_boot[2,]
  T_boot3[,i] <- T_boot[3,]
  T_boot4[,i] <- T_boot[4,]
}

error_rel <- function(x,y) {abs(x - y)/y*100}
for (j in 1:4){
  momento_analitico <- 2**j*gamma(j + (n - 1)/2)/gamma((n - 1)/2)
  print(paste0("Error relativo (medio) sin modificar para el momento de orden
    ",j," (%): ",mean(error_rel(apply(T_boot1**j,2,mean),momento_analitico))))
  print(paste0("Error relativo (medio) con factor 0.5 para el momento de orden
    ",j," (%): ",mean(error_rel(apply(T_boot2**j,2,mean),momento_analitico))))
  print(paste0("Error relativo (medio) con factor 0.33 para el momento de orden
    ",j," (%): ",mean(error_rel(apply(T_boot3**j,2,mean),momento_analitico))))
  print(paste0("Error relativo (medio) con factor 0.25 para el momento de orden
    ",j," (%): ",mean(error_rel(apply(T_boot4**j,2,mean),momento_analitico))))
}
```

Código A.12: Código utilizado para obtener los resultados de la Tabla 3.1.

- El siguiente código fue empleado aquí:

```r
# Gráfica de una de las distribuciones bootstrap
```

```r
imax <- which.max(error_rel(apply(T_boot2,2,mean),n - 1))
imin <- which.min(error_rel(apply(T_boot2,2,mean),n - 1))

p1 <- ggplot(data = tibble(T_boot2[,imax]),aes(x = T_boot2[,imax])) +
  geom_histogram(aes(y = ..density..),binwidth = 1)+
  stat_function(fun = dchisq,args = n - 1,colour = "red",size = 1.2,aes(color =
    "dchisq")) +
  labs(title = "Distribución de T* (max. error rel.)") + xlab("") + ylab("T*") +
  scale_color_manual("",values = c(dchisq = "red")) +
  theme(legend.position = c(0.85,0.8),
        legend.background = element_rect(fill = "transparent"),
        legend.text = element_text(face = "bold",size = 10),
        plot.title = element_text(size = 18,face = "bold"),
        axis.title.y = element_text(size = 18))
p1

p2 <- ggplot(data = tibble(T_boot2[,imin]),aes(x = T_boot2[,imin])) +
  geom_histogram(aes(y = ..density..),binwidth = 1)+
  stat_function(fun = dchisq,args = n - 1,colour = "red",size = 1.2,aes(color =
    "dchisq")) +
  labs(title = "Distribución de T* (max. error rel.)") + xlab("") + ylab("T*") +
  scale_color_manual("",values = c(dchisq = "red")) +
  theme(legend.position = c(0.85,0.8),
        legend.background = element_rect(fill = "transparent"),
        legend.text = element_text(face = "bold",size = 10),
        plot.title = element_text(size = 18,face = "bold"),
        axis.title.y = element_text(size = 18))
p2
```

Código A.13: Código utilizado en la representación de la Figura 3.5.

- El siguiente código fue empleado aquí:

```r
# Generación de números aleatorios a partir de Epanechnikov
k <- 1e5
epanechnikov_den <- function(u){3/4*(1 - u^2)*(-1 <= u & u <= 1)}
g <- function(u){3/4*(-1 <= u & u <= 1)}
C <- integrate(g,-Inf,Inf)$value # Valor de G(x) cuando x tiende a infinito

```

```r
set.seed(1)
epanechnikov_val <- replicate(k,{
  while (TRUE){
    u <- runif(1,0,C)
    t <- 4/3*u - 1
    if (runif(1) < epanechnikov_den(t)/g(t)){return(t)}
  }
})

p1 <- ggplot(data = tibble(epanechnikov_val),aes(x = epanechnikov_val)) +
  geom_histogram(aes(y = ..density..),binwidth = 0.05)+
  stat_function(fun = epanechnikov_den,color = "red",size = 1.2) +
  stat_function(fun = g,color = "blue",size = 1.2) +
  labs(title = "Distribución de Epanechnikov") + xlab("") + ylab(TeX("$K^*_E$")) +
  theme(legend.text = element_text(face = "bold",size = 10),
        plot.title = element_text(size = 18,face = "bold"),
        axis.title.y = element_text(size = 18))
p1
```

Código A.14: Código utilizado para la representación de la Figura 3.6.

- El siguiente código fue empleado aquí:

```r
# Estudio del error relativo variando el kernel
n_muestra <- 100
n <- 25
mu <- 17.4

set.seed(1)
muestras <- matrix(0,nrow = n,ncol = n_muestra) # Guardamos en una matriz las
                                                # muestras
for (i in 1:n_muestra){
  muestras[,i] <- rnorm(n,mu,2.1)
}

S2_vector <- apply(muestras,2,var)

density_g <- function(x){density(x,kernel = "gaussian")}
fh_vector <- apply(muestras,2,density_g)
```

```
17
18 h_vector <- numeric(n_muestra)
19
20 for (i in 1:n_muestra){
21   h_vector[i] <- fh_vector[[i]]$bw
22 }
23
24 # Funciones para la generación de números aleatorios a partir de Epanechnikov
25 epanechnikov_den <- function(u){3/4*(1 - u^2)*(-1 <= u & u <= 1)}
26 g <- function(u){3/4*(-1 <= u & u <= 1)}
27 C <- integrate(g,-Inf,Inf)$value # Valor de G(x) cuando x tiende a infinito
28
29 B <- 1e5
30
31 T_boot1 <- matrix(0,nrow = B,ncol = n_muestra)
32 T_boot2 <- matrix(0,nrow = B,ncol = n_muestra)
33 T_boot3 <- matrix(0,nrow = B,ncol = n_muestra)
34
35 for (i in 1:n_muestra){
36   T_boot <- replicate(B,{
37     random_epanechnikov <- replicate(n,{
38       while (TRUE){
39         u <- runif(1,0,C)
40         t <- 4/3*u - 1
41         if (runif(1) < epanechnikov_den(t)/g(t)){return(t)}
42       }
43     })
44     remuestra <- sample(muestras[,i],n,replace = TRUE)
45     remuestra1 <- remuestra + h_vector[i]*rnorm(n)
46     remuestra2 <- remuestra + h_vector[i]*runif(n,-1,1)
47     remuestra3 <- remuestra + h_vector[i]*random_epanechnikov
48
49     S2_boot_all <- apply(rbind(remuestra1,remuestra2,remuestra3),1,var)
50     (n - 1)/S2_vector[i]*S2_boot_all
51   })
52   T_boot1[,i] <- T_boot[1,]
53   T_boot2[,i] <- T_boot[2,]
```

```
  T_boot3[,i] <- T_boot[3,]
}
error_rel <- function(x,y) {abs(x - y)/y*100}
for (j in 1:4){
  momento_analitico <- 2**j*gamma(j + (n - 1)/2)/gamma((n - 1)/2)
  print(paste0("Error relativo (medio) con kernel gaussiano para el momento de
    orden ",j," (%): ",
                mean(error_rel(apply(T_boot1**j,2,mean),momento_analitico))))
  print(paste0("Error relativo (medio) con kernel rectangular para el momento de
    orden ",j," (%): ",
                mean(error_rel(apply(T_boot2**j,2,mean),momento_analitico))))
  print(paste0("Error relativo (medio) con kernel epanechnikov para el momento de
    orden ",j," (%): ",
                mean(error_rel(apply(T_boot3**j,2,mean),momento_analitico))))
}
```

Código A.15: Código utilizado para la obtención de los resultados de la Tabla 3.2.

- El siguiente código fue empleado aquí:

```
# Comparación métodos bootstrap (uniforme, paramétrico, simetrizado, suavizado)
# Librerías
library(tidyverse)
library(progress)

# Creamos la muestras y elegimos los parámetros que van a tener
n_muestra <- 100 # Calcularemos 100 muestras para los métodos (las mismas)
n <- 25 # Tamaño de cada muestra
mu <- 17.4 # Media conocida

set.seed(1)
muestras <- matrix(0,nrow = n,ncol = n_muestra) # Guardamos en una matriz las
                                                # muestras
for (i in 1:n_muestra){
  muestras[,i] <- rnorm(n,mu,2.1)
}

# Configuramos la barra de progreso
pb <- progress_bar$new(format = "(:spin) [:bar] :percent [Tiempo transcurrido:
```

```
   :elapsedfull || Tiempo restante estimado: :eta]",
                      total = n_muestra, # Número iteraciones for
                      complete = "=",    # Caracteres de las iteraciones
                                         #finalizadas
                      incomplete = "-", # Caracteres de las iteraciones no
                                         #finalizadas
                      current = ">",     # Caracter actual
                      clear = FALSE,     # Si TRUE, borra la barra cuando
                                         #termine
                      width = 100)

# Uniforme y paramétrico
S2_vector <- apply(muestras,2,var)

mu_EMV_vector <- apply(muestras,2,mean)
sigma2_EMV <- function(x){1/length(x)*sum((x - mean(x))**2)}
sigma2_EMV_vector <- apply(muestras,2,sigma2_EMV)

# Simétrico
# Hacemos broadcasting para operar un vector de longitud 100 con una matriz 25x100
# con la función sweep. Después concatenamos las matrices para obtener la
# "matriz simetrizada"
muestras_sim <- rbind(muestras,-sweep(muestras,2,2*apply(muestras,2,mean)))

# Suavizado
# Parámetro de suavizado
fh_vector <- apply(muestras,2,density)
h_vector <- numeric(n_muestra)
for (i in 1:n_muestra){
  h_vector[i] <- fh_vector[[i]]$bw
}

# Bootstrap
B <- 1e5 # Número de réplicas bootstrap
T_boot_u <- matrix(0,nrow = B,ncol = n_muestra)
T_boot_p <- matrix(0,nrow = B,ncol = n_muestra)
T_boot_sim <- matrix(0,nrow = B,ncol = n_muestra)
```

```r
T_boot_su <- matrix(0,nrow = B,ncol = n_muestra)

for (i in 1:n_muestra){
  pb$tick() # Barra de progreso

  T_boot <- replicate(B,{
    remuestra_u <- sample(muestras[,i],n,replace = TRUE)
    remuestra_p <- rnorm(n,mu_EMV_vector[i],sqrt(sigma2_EMV_vector[i]))
    remuestra_sim <- sample(muestras_sim[,i],n,replace = TRUE)
    remuestra_su <- sample(muestras[,i],n,replace = TRUE) + h_vector[i]*rnorm(n)/2

    t1 <- (n - 1)/S2_vector[i]*var(remuestra_u)
    t2 <- (n - 1)/sigma2_EMV_vector[i]*var(remuestra_p)
    t3 <- (n - 1)/S2_vector[i]*var(remuestra_sim)
    t4 <- (n - 1)/S2_vector[i]*var(remuestra_su)
    c(t1,t2,t3,t4)
  })
  T_boot_u[,i] <- T_boot[1,]
  T_boot_p[,i] <- T_boot[2,]
  T_boot_sim[,i] <- T_boot[3,]
  T_boot_su[,i] <- T_boot[4,]
}

# Calculamos el error relativo con respecto a los momentos poblacionales
# para cada método
error_rel <- function(x,y) {abs(x - y)/y*100}
for (j in 1:4){
  momento_analitico <- 2**j*gamma(j + (n - 1)/2)/gamma((n - 1)/2)
  print(paste0("Error relativo (medio) unif para el momento de orden ",j," (%): ",
               mean(error_rel(apply(T_boot_u**j,2,mean),momento_analitico))))
  print(paste0("Error relativo (medio) param para el momento de orden ",j," (%): ",
               mean(error_rel(apply(T_boot_p**j,2,mean),momento_analitico))))
  print(paste0("Error relativo (medio) sim para el momento de orden ",j," (%): ",
               mean(error_rel(apply(T_boot_sim**j,2,mean),momento_analitico))))
  print(paste0("Error relativo (medio) suav para el momento de orden ",j," (%): ",
               mean(error_rel(apply(T_boot_su**j,2,mean),momento_analitico))))
```

```
}
```

Código A.16: Código utilizado para la obtención de los resultados de la Tabla 3.3.

- El siguiente código fue empleado aquí:

```
# Intervalo de confianza bootstrap uniforme para la desviación típica
set.seed(1)
n <- 30
muestra <- rnorm(n,17.4,2.1)
alfa <- 0.05

S2 <- sd(muestra)

B <- 1e5
T_boot <- replicate(B,{
  remuestra <- sample(muestra,n,replace = TRUE)
  sd(remuestra)
})
T_boot <- T_boot - S2
c1 <- quantile(T_boot,alfa/2)
c2 <- quantile(T_boot,1 - alfa/2)

print(paste0("IC(sigma^2) = [",S2 - c2,",",S2 - c1,"]"))
print(paste0("IC(sigma) = [",sqrt(S2 - c2),",",sqrt(S2 - c1),"]"))
```

Código A.17: Código utilizado para la obtención del intervalo de confianza básico (uniforme).

- El siguiente código fue empleado aquí:

```
# Intervalo de confianza bootstrap paramétrico para la desviación típica
set.seed(1)
n <- 30
muestra <- rnorm(n,17.4,2.1)
alfa <- 0.05

sigma2_EMV <- 1/n*sum((muestra - mean(muestra))**2)
S2 <- var(muestra)

B <- 1e5
```

```r
T_boot <- replicate(B,{
  remuestra <- rnorm(n,mean(muestra),sqrt(sigma2_EMV))
  var(remuestra)
})
T_boot <- T_boot - S2
c1 <- quantile(T_boot,alfa/2)
c2 <- quantile(T_boot,1 - alfa/2)

print(paste0("IC(sigma^2) = [",S2 - c2,",",S2 - c1,"]"))
print(paste0("IC(sigma) = [",sqrt(S2 - c2),",",sqrt(S2 - c1),"]"))
```

Código A.18: Código utilizado para la obtención del intervalo de confianza básico (paramétrico).

- El siguiente código fue empleado aquí:

```r
# Intervalo de confianza bootstrap simetrizado para la desviación típica
set.seed(1)
n <- 30
muestra <- rnorm(n,17.4,2.1)
alfa <- 0.05

muestra_sim <- c(muestra,2*mean(muestra) - muestra)
S2 <- var(muestra)

B <- 1e5
T_boot <- replicate(B,{
  remuestra <- sample(muestra_sim,n,replace = TRUE)
  var(remuestra)
})
T_boot <- T_boot - S2
c1 <- quantile(T_boot,alfa/2)
c2 <- quantile(T_boot,1 - alfa/2)

print(paste0("IC(sigma^2) = [",S2 - c2,",",S2 - c1,"]"))
print(paste0("IC(sigma) = [",sqrt(S2 - c2),",",sqrt(S2 - c1),"]"))
```

Código A.19: Código utilizado para la obtención del intervalo de confianza básico (simetrizado).

- El siguiente código fue empleado aquí:

```
# Intervalo de confianza bootstrap suavizado para la desviación típica
set.seed(1)
n <- 30
muestra <- rnorm(n,17.4,2.1)
alfa <- 0.05

S2 <- var(muestra)

fh <- density(muestra)
h <- fh$bw

B <- 1e5
T_boot <- replicate(B,{
  remuestra <- sample(muestra,n,replace = TRUE) + h/2*rnorm(n)
  var(remuestra)
})
T_boot <- T_boot - S2
c1 <- quantile(T_boot,alfa/2)
c2 <- quantile(T_boot,1 - alfa/2)

print(paste0("IC(sigma^2) = [",S2 - c2,",",S2 - c1,"]"))
print(paste0("IC(sigma) = [",sqrt(S2 - c2),",",sqrt(S2 - c1),"]"))
```

Código A.20: Código utilizado para la obtención del intervalo de confianza básico (suavizado).

- El siguiente código fue empleado aquí:

```
# Intervalo de confianza para la desviación estándar
set.seed(1)
n <- 30
muestra <- rnorm(n,17.4,2.1)
alfa <- 0.05

S2 <- var(muestra)

d1 <- qchisq(alfa/2,n - 1)
d2 <- qchisq(1 - alfa/2,n - 1)

```

```r
print(paste0("IC(sigma^2) = [",(n - 1)/d2*S2,",",(n - 1)/d1*S2,"]"))
print(paste0("IC(sigma) = [",sqrt((n - 1)/d2*S2),",",sqrt((n - 1)/d1*S2),"]"))
```

Código A.21: Código utilizado para la obtención del intervalo de confianza exacto.

- El siguiente código fue empleado aquí:

```r
# Cobertura y longitud de intervalos de confianza bootstrap y exacto método
# percentil básico
ICboot <- function(muestra,metodo,B,alfa,T_muestra,mu = 17.4){
  n <- length(muestra)
  if (metodo == "unif"){
    T_boot <- replicate(B,{
      remuestra <- sample(muestra,n,replace = TRUE)
      var(remuestra)
    })
  }
  if (metodo == "param"){
    T_boot <- replicate(B,{
      remuestra <- rnorm(n,mu,sqrt(T_muestra))
      1/n*sum((remuestra - mu)**2)
    })
  }
  if (metodo == "sim"){
    muestra_sim <- c(muestra,2*mean(muestra) - muestra)
    T_boot <- replicate(B,{
      remuestra <- sample(muestra_sim,n,replace = TRUE)
      var(remuestra)
    })
  }
  if (metodo == "suav"){
    h <- density(muestra)$bw
    T_boot <- replicate(B,{
      remuestra <- sample(muestra,n,replace = TRUE) + h/2*rnorm(n)
      var(remuestra)
    })
  }

  T_boot <- T_boot - T_muestra
```

```
  c1 <- quantile(T_boot,alfa/2,names = FALSE)
  c2 <- quantile(T_boot,1 - alfa/2,names = FALSE)

  c(T_muestra - c2,T_muestra - c1)
}

set.seed(1)
n_muestra <- 1000
n <- 30
mu <- 17.4

alfa <- 0.05
B <- 1e4
ICs <- replicate(n_muestra,{
  muestra <- rnorm(n,17.4,2.1)
  S2 <- var(muestra)
  EMV <- 1/n*sum((muestra - mu)**2)

  IC1 <- ICboot(muestra,"unif",B,alfa,S2)
  IC2 <- ICboot(muestra,"param",B,alfa,EMV)
  IC3 <- ICboot(muestra,"sim",B,alfa,S2)
  IC4 <- ICboot(muestra,"suav",B,alfa,S2)

  d1 <- qchisq(alfa/2,n - 1)
  d2 <- qchisq(1 - alfa/2,n - 1)

  c(c((n - 1)/d2*S2,(n - 1)/d1*S2),IC1,IC2,IC3,IC4)
})

# Cobertura de los intervalos
sigma <- 2.1
metodo <- c("exacto","unif","param","sim","suav")
for (i in 1:length(metodo)){
  print(paste0("Cobertura de ",metodo[i]," : ",mean(sqrt(ICs[2*i - 1,]) <= sigma &
    sigma <= sqrt(ICs[2*i,]))))
}

```

```r
# Longitud de los intervalos
for (i in 1:length(metodo)){
  print(paste0("Longitud de ",metodo[i]," : ",mean(sqrt(ICs[2*i,]) - sqrt(ICs[2*i
    - 1,]))))
}
```

Código A.22: Código utilizado para la obtención de los resultados de la Tabla 4.2.

- El siguiente código fue empleado aquí:

```r
# Intervalo de confianza bootstrap uniforme para la desviación típica
set.seed(1)
n <- 30
muestra <- rnorm(n,17.4,2.1)
alfa <- 0.05

S2 <- var(muestra)

B <- 1e5
T_boot <- replicate(B,{
  remuestra <- sample(muestra,n,replace = TRUE)
  var(remuestra)
})
T_boot <- sqrt((n - 1)/2)*(T_boot - S2)/S2
c1 <- quantile(T_boot,alfa/2)
c2 <- quantile(T_boot,1 - alfa/2)

print(paste0("IC(sigma^2) = [",S2/(1 + sqrt(2/(n - 1))*c2),",",S2/(1 + sqrt(2/(n -
    1))*c1),"]"))
print(paste0("IC(sigma) = [",sqrt(S2/(1 + sqrt(2/(n - 1))*c2)),",",sqrt(S2/(1 +
    sqrt(2/(n - 1))*c1)),"]"))
```

Código A.23: Código utilizado para la obtención del intervalo de confianza t (uniforme).

- El siguiente código fue empleado aquí:

```r
# Intervalo de confianza bootstrap paramétrico para la desviación típica
set.seed(1)
n <- 30
muestra <- rnorm(n,17.4,2.1)
```

```r
alfa <- 0.05
mu <- 17.4

EMV <- 1/n*sum((muestra - mu)**2)
S2 <- var(muestra)

B <- 1e5
T_boot <- replicate(B,{
  remuestra <- rnorm(n,mu,sqrt(EMV))
  var(remuestra)
})
T_boot <- sqrt((n - 1)/2)*(T_boot - S2)/S2
c1 <- quantile(T_boot,alfa/2)
c2 <- quantile(T_boot,1 - alfa/2)

print(paste0("IC(sigma^2) = [",S2/(1 + sqrt(2/(n - 1))*c2),",",S2/(1 + sqrt(2/(n - 1))*c1),"]"))
print(paste0("IC(sigma) = [",sqrt(S2/(1 + sqrt(2/(n - 1))*c2)),",",sqrt(S2/(1 + sqrt(2/(n - 1))*c1)),"]"))
```

Código A.24: Código utilizado para la obtención del intervalo de confianza t (paramétrico).

- El siguiente código fue empleado aquí:

```r
# Intervalo de confianza bootstrap simetrizado para la desviación típica
set.seed(1)
n <- 30
muestra <- rnorm(n,17.4,2.1)
alfa <- 0.05
mu <- 17.4

muestra_sim <- c(muestra,2*mean(muestra) - muestra)
S2 <- var(muestra)

B <- 1e5
T_boot <- replicate(B,{
  remuestra <- sample(muestra_sim,n,replace = TRUE)
  var(remuestra)
})
```

```r
T_boot <- sqrt((n - 1)/2)*(T_boot - S2)/S2
c1 <- quantile(T_boot,alfa/2)
c2 <- quantile(T_boot,1 - alfa/2)

print(paste0("IC(sigma^2) = [",S2/(1 + sqrt(2/(n - 1))*c2),",",S2/(1 + sqrt(2/(n - 1))*c1),"]"))
print(paste0("IC(sigma) = [",sqrt(S2/(1 + sqrt(2/(n - 1))*c2)),",",sqrt(S2/(1 + sqrt(2/(n - 1))*c1)),"]"))
```

Código A.25: Código utilizado para la obtención del intervalo de confianza t (simetrizado).

- El siguiente código fue empleado aquí:

```r
# Intervalo de confianza bootstrap suavizadp para la desviación típica
set.seed(1)
n <- 30
muestra <- rnorm(n,17.4,2.1)
alfa <- 0.05

S2 <- var(muestra)
h <- density(muestra)$bw

B <- 1e5
T_boot <- replicate(B,{
  remuestra <- sample(muestra,n,replace = TRUE) + h/2*rnorm(n,0,1)
  var(remuestra)
})
T_boot <- sqrt((n - 1)/2)*(T_boot - S2)/S2
c1 <- quantile(T_boot,alfa/2)
c2 <- quantile(T_boot,1 - alfa/2)

print(paste0("IC(sigma^2) = [",S2/(1 + sqrt(2/(n - 1))*c2),",",S2/(1 + sqrt(2/(n - 1))*c1),"]"))
print(paste0("IC(sigma) = [",sqrt(S2/(1 + sqrt(2/(n - 1))*c2)),",",sqrt(S2/(1 + sqrt(2/(n - 1))*c1)),"]"))
```

Código A.26: Código utilizado para la obtención del intervalo de confianza t (suavizado).

- El siguiente código fue empleado aquí:

```r
# Cobertura y longitud de intervalos de confianza bootstrap y exacto método
# percentil t
ICboot_t <- function(muestra,metodo,B,alfa){
  n <- length(muestra)
  if (metodo == "unif"){
    T_boot <- replicate(B,{
      remuestra <- sample(muestra,n,replace = TRUE)
      var(remuestra)
    })
  }
  if (metodo == "param"){
    EMV <- 1/n*sum((muestra - mean(muestra))**2)
    T_boot <- replicate(B,{
      remuestra <- rnorm(n,mean(muestra),sqrt(EMV))
      var(remuestra)
    })
  }
  if (metodo == "sim"){
    muestra_sim <- c(muestra,2*mean(muestra) - muestra)
    T_boot <- replicate(B,{
      remuestra <- sample(muestra_sim,n,replace = TRUE)
      var(remuestra)
    })
  }
  if (metodo == "suav"){
    h <- density(muestra)$bw
    T_boot <- replicate(B,{
      remuestra <- sample(muestra,n,replace = TRUE) + h/2*rnorm(n)
      var(remuestra)
    })
  }
  S2 <- var(muestra)
  T_boot <- sqrt((n - 1)/2)*(T_boot - S2)/S2
  c1 <- quantile(T_boot,alfa/2,names = FALSE)
  c2 <- quantile(T_boot,1 - alfa/2,names = FALSE)

  c(S2/(1 + sqrt(2/(n - 1))*c2),S2/(1 + sqrt(2/(n - 1))*c1))
```

```r
}

set.seed(1)
n_muestra <- 1000
n <- 30

alfa <- 0.05
B <- 1e4
ICs <- replicate(n_muestra,{
  muestra <- rnorm(n,17.4,2.1)
  S2 <- var(muestra)

  IC1 <- ICboot_t(muestra,"unif",B,alfa)
  IC2 <- ICboot_t(muestra,"param",B,alfa)
  IC3 <- ICboot_t(muestra,"sim",B,alfa)
  IC4 <- ICboot_t(muestra,"suav",B,alfa)

  d1 <- qchisq(alfa/2,n - 1)
  d2 <- qchisq(1 - alfa/2,n - 1)

  c(c((n - 1)/d2*S2,(n - 1)/d1*S2),IC1,IC2,IC3,IC4)
})

# Cobertura de los intervalos
sigma <- 2.1
metodo <- c("exacto","unif","param","sim","suav")
for (i in 1:length(metodo)){
  print(paste0("Cobertura de ",metodo[i]," : ",mean(sqrt(ICs[2*i - 1,]) <= sigma &
    sigma <= sqrt(ICs[2*i,]))))
}

# Longitud de los intervalos
for (i in 1:length(metodo)){
  print(paste0("Longitud de ",metodo[i]," : ",mean(sqrt(ICs[2*i,]) - sqrt(ICs[2*i
    - 1,]))))
}
```

Código A.27: Código utilizado para la obtención de los resultados de la Tabla 4.5.

- El siguiente código fue utilizado aquí:

```
# Intervalo de confianza bootstrap uniforme para la desviación típica
set.seed(1)
n <- 30
muestra <- rnorm(n,17.4,2.1)
alfa <- 0.05

S2 <- var(muestra)

B <- 1e5
T_boot <- replicate(B,{
  remuestra <- sample(muestra,n,replace = TRUE)
  var(remuestra)
})
T_boot <- sqrt((n - 1)/2)*(T_boot - S2)/S2

c_sim <- quantile(abs(T_boot),1 - alfa)

print(paste0("IC(sigma^2) = [",S2/(1 + sqrt(2/(n - 1))*c_sim),",",S2/(1 -
    sqrt(2/(n - 1))*c_sim),"]"))
print(paste0("IC(sigma) = [",sqrt(S2/(1 + sqrt(2/(n - 1))*c_sim)),",",sqrt(S2/(1 -
    sqrt(2/(n - 1))*c_sim)),"]"))
```

Código A.28: Código utilizado para la obtención del intervalo de confianza t simetrizado (uniforme).

- El siguiente código fue utilizado aquí:

```
# Intervalo de confianza bootstrap paramétrico para la desviación típica
set.seed(1)
n <- 30
muestra <- rnorm(n,17.4,2.1)
alfa <- 0.05

S2 <- var(muestra)
EMV <- 1/n*sum((muestra - mean(muestra))**2)

B <- 1e5
```

```r
T_boot <- replicate(B,{
  remuestra <- rnorm(n,mean(muestra),sqrt(EMV))
  var(remuestra)
})
T_boot <- sqrt((n - 1)/2)*(T_boot - S2)/S2

c_sim <- quantile(abs(T_boot),1 - alfa)

print(paste0("IC(sigma^2) = [",S2/(1 + sqrt(2/(n - 1))*c_sim),",",S2/(1 -
    sqrt(2/(n - 1))*c_sim),"]"))
print(paste0("IC(sigma) = [",sqrt(S2/(1 + sqrt(2/(n - 1))*c_sim)),",",sqrt(S2/(1 -
    sqrt(2/(n - 1))*c_sim)),"]"))
```

Código A.29: Código utilizado para la obtención del intervalo de confianza t simetrizado (paramétrico).

- El siguiente código fue utilizado aquí:

```r
# Intervalo de confianza bootstrap simetrizado para la desviación típica
set.seed(1)
n <- 30
muestra <- rnorm(n,17.4,2.1)
alfa <- 0.05

S2 <- var(muestra)
muestra_sim <- c(muestra,2*mean(muestra) - muestra)

B <- 1e5
T_boot <- replicate(B,{
  remuestra <- sample(muestra_sim,n,replace = TRUE)
  var(remuestra)
})
T_boot <- sqrt((n - 1)/2)*(T_boot - S2)/S2

c_sim <- quantile(abs(T_boot),1 - alfa)

print(paste0("IC(sigma^2) = [",S2/(1 + sqrt(2/(n - 1))*c_sim),",",S2/(1 -
    sqrt(2/(n - 1))*c_sim),"]"))
```

```
print(paste0("IC(sigma) = [",sqrt(S2/(1 + sqrt(2/(n - 1))*c_sim)),",",sqrt(S2/(1 -
    sqrt(2/(n - 1))*c_sim)),"]"))
```

Código A.30: Código utilizado para la obtención del intervalo de confianza t simetrizado (simetrizado).

- El siguiente código fue utilizado aquí:

```
# Intervalo de confianza bootstrap suavizado para la desviación típica
set.seed(1)
n <- 30
muestra <- rnorm(n,17.4,2.1)
alfa <- 0.05

S2 <- var(muestra)
h <- density(muestra)$bw

B <- 1e5
T_boot <- replicate(B,{
  remuestra <- sample(muestra,n,replace = TRUE) + h/2*rnorm(n)
  var(remuestra)
})
T_boot <- sqrt((n - 1)/2)*(T_boot - S2)/S2

c_sim <- quantile(abs(T_boot),1 - alfa)

print(paste0("IC(sigma^2) = [",S2/(1 + sqrt(2/(n - 1))*c_sim),",",S2/(1 -
    sqrt(2/(n - 1))*c_sim),"]"))
print(paste0("IC(sigma) = [",sqrt(S2/(1 + sqrt(2/(n - 1))*c_sim)),",",sqrt(S2/(1 -
    sqrt(2/(n - 1))*c_sim)),"]"))
```

Código A.31: Código utilizado para la obtención del intervalo de confianza t simetrizado (suavizado).

- El siguiente código fue utilizado aquí:

```
# Cobertura y longitud de intervalos de confianza bootstrap y exacto método
# percentil t simetrizado
ICboot_t_sim <- function(muestra,metodo,B,alfa){
  n <- length(muestra)
```

```r
  if (metodo == "unif"){
    T_boot <- replicate(B,{
      remuestra <- sample(muestra,n,replace = TRUE)
      var(remuestra)
    })
  }
  if (metodo == "param"){
    EMV <- 1/n*sum((muestra - mean(muestra))**2)
    T_boot <- replicate(B,{
      remuestra <- rnorm(n,mean(muestra),sqrt(EMV))
      var(remuestra)
    })
  }
  if (metodo == "sim"){
    muestra_sim <- c(muestra,2*mean(muestra) - muestra)
    T_boot <- replicate(B,{
      remuestra <- sample(muestra_sim,n,replace = TRUE)
      var(remuestra)
    })
  }
  if (metodo == "suav"){
    h <- density(muestra)$bw
    T_boot <- replicate(B,{
      remuestra <- sample(muestra,n,replace = TRUE) + h/2*rnorm(n)
      var(remuestra)
    })
  }
  S2 <- var(muestra)
  T_boot <- sqrt((n - 1)/2)*(T_boot - S2)/S2
  c_sim <- quantile(abs(T_boot),1 - alfa)

  c(S2/(1 + sqrt(2/(n - 1))*c_sim),S2/(1 - sqrt(2/(n - 1))*c_sim))
}

set.seed(1)
n_muestra <- 1000
n <- 30
```

```r

alfa <- 0.05
B <- 1e4
ICs <- replicate(n_muestra,{
  muestra <- rnorm(n,17.4,2.1)
  S2 <- var(muestra)

  IC1 <- ICboot_t_sim(muestra,"unif",B,alfa)
  IC2 <- ICboot_t_sim(muestra,"param",B,alfa)
  IC3 <- ICboot_t_sim(muestra,"sim",B,alfa)
  IC4 <- ICboot_t_sim(muestra,"suav",B,alfa)

  d1 <- qchisq(alfa/2,n - 1)
  d2 <- qchisq(1 - alfa/2,n - 1)

  c(c((n - 1)/d2*S2,(n - 1)/d1*S2),IC1,IC2,IC3,IC4)
})

# Cobertura de los intervalos
sigma <- 2.1
metodo <- c("exacto","unif","param","sim","suav")
for (i in 1:length(metodo)){
  print(paste0("Cobertura de ",metodo[i]," : ",mean(sqrt(ICs[2*i - 1,]) <= sigma &
    sigma <= sqrt(ICs[2*i,]))))
}

# Longitud de los intervalos
for (i in 1:length(metodo)){
  print(paste0("Longitud de ",metodo[i]," : ",mean(sqrt(ICs[2*i,]) - sqrt(ICs[2*i
    - 1,]))))
}
```

Código A.32: Código utilizado para la obtención de los resultados de la Tabla 4.9.

- El siguiente código fue utilizado aquí:

```r
# Intervalo de confianza bootstrap para la media
set.seed(1)
n <- 50
```

```r
muestra <- rnorm(n)
alfa <- 0.1

x_barra <- mean(muestra)
cuasi_dt <- sd(muestra)

sigma_EMV <- sqrt(1/n*sum((muestra - mean(muestra))**2))
muestra_sim <- c(muestra,2*x_barra - muestra)
h <- density(muestra)$bw

B <- 1e5
T_boot <- replicate(B,{
  remuestra_u <- sample(muestra,n,replace = TRUE)
  remuestra_p <- rnorm(n,x_barra,sigma_EMV)
  remuestra_sim <- sample(muestra_sim,n,replace = TRUE)
  remuestra_su <- sample(muestra,n,replace = TRUE) + h/2*rnorm(n)
  x_barra_boot <-
    c(mean(remuestra_u),mean(remuestra_p),mean(remuestra_sim),mean(remuestra_su))
  cuasi_dt_boot <-
    c(sd(remuestra_u),sd(remuestra_p),sd(remuestra_sim),sd(remuestra_su))
  sqrt(n)*(x_barra_boot - x_barra)/cuasi_dt_boot
})

metodo <- c("unif","param","sim","suav")
for (i in 1:length(metodo)){
  c1 <- quantile(T_boot[i,],alfa/2)
  c2 <- quantile(T_boot[i,],1 - alfa/2)
  print(paste0(metodo[i],": IC(mu) = [",x_barra - c2*cuasi_dt/sqrt(n),",",x_barra
    - c1*cuasi_dt/sqrt(n),"]"))
}
d1 <- qt(alfa/2,n - 1)
d2 <- qt(1 - alfa/2,n - 1)
print(paste0("exacto: IC(mu) = [",x_barra - d2*cuasi_dt/sqrt(n),",",x_barra -
    d1*cuasi_dt/sqrt(n),"]"))
```

Código A.33: Código utilizado para la obtención de los resultados de la Tabla 5.1.

- El siguiente código fue utilizado aquí:

```r
# Cobertura y longitud de intervalos de confianza bootstrap y exacto método
# percentil t
ICboot_t <- function(muestra,metodo,B,alfa){
  n <- length(muestra)
  x_barra <- mean(muestra)
  cuasi_dt <- sd(muestra)

  if (metodo == "unif"){
    T_boot <- replicate(B,{
      remuestra <- sample(muestra,n,replace = TRUE)
      sqrt(n)*(mean(remuestra) - x_barra)/sd(remuestra)
    })
  }
  if (metodo == "param"){
    sigma_EMV <- sqrt(1/n*sum((muestra - x_barra)**2))
    T_boot <- replicate(B,{
      remuestra <- rnorm(n,x_barra,sigma_EMV)
      sqrt(n)*(mean(remuestra) - x_barra)/sd(remuestra)
    })
  }
  if (metodo == "sim"){
    muestra_sim <- c(muestra,2*x_barra - muestra)
    T_boot <- replicate(B,{
      remuestra <- sample(muestra_sim,n,replace = TRUE)
      sqrt(n)*(mean(remuestra) - x_barra)/sd(remuestra)
    })
  }
  if (metodo == "suav"){
    h <- density(muestra)$bw
    T_boot <- replicate(B,{
      remuestra <- sample(muestra,n,replace = TRUE) + h/2*rnorm(n)
      sqrt(n)*(mean(remuestra) - x_barra)/sd(remuestra)
    })
  }

  c1 <- quantile(T_boot,alfa/2,names = FALSE)
  c2 <- quantile(T_boot,1 - alfa/2,names = FALSE)
```

```
38
39    c(x_barra - c2*cuasi_dt/sqrt(n),x_barra - c1*cuasi_dt/sqrt(n))
40  }
41
42  set.seed(123)
43  n_muestra <- 1000
44  n <- 50
45
46  alfa <- 0.1
47  B <- 1e3
48  ICs <- replicate(n_muestra,{
49    muestra <- rnorm(n)
50    x_barra <- mean(muestra)
51    cuasi_dt <- sd(muestra)
52
53    IC1 <- ICboot_t(muestra,"unif",B,alfa)
54    IC2 <- ICboot_t(muestra,"param",B,alfa)
55    IC3 <- ICboot_t(muestra,"sim",B,alfa)
56    IC4 <- ICboot_t(muestra,"suav",B,alfa)
57
58    d1 <- qt(alfa/2,n - 1)
59    d2 <- qt(1 - alfa/2,n - 1)
60
61    c(c(x_barra - d2*cuasi_dt/sqrt(n),x_barra - d1*cuasi_dt/sqrt(n)),IC1,IC2,IC3,IC4)
62  })
63
64  # Cobertura de los intervalos
65  mu <- 0
66  metodo <- c("exacto","unif","param","sim","suav")
67  for (i in 1:length(metodo)){
68    print(paste0("Cobertura de ",metodo[i]," : ",mean(ICs[2*i - 1,] <= mu & mu <=
        ICs[2*i,])))
69  }
70
71  # Longitud de los intervalos
72  for (i in 1:length(metodo)){
73    print(paste0("Longitud de ",metodo[i]," : ",mean(ICs[2*i,] - ICs[2*i - 1,])))
```

```
}
```

Código A.34: Código utilizado para la obtención de los resultados de la Tabla 5.2.

- El siguiente código fue utilizado aquí:

```
# Plot de los intervalos de confianza
library(tidyverse)
for (i in 1:length(metodo)){
  df <- data.frame(x = 1:100,
                   L = ICs[2*i - 1,1:100],
                   U = ICs[2*i,1:100],
                   L_not_in = ICs[2*i - 1,1:100] > mu | mu > ICs[2*i,1:100],
                   U_not_in = ICs[2*i - 1,1:100] > mu | mu > ICs[2*i,1:100])

  p <- ggplot(data = df,aes(x = x)) +
    geom_errorbar(aes(ymax = U, ymin = L,color = L_not_in)) +
    geom_hline(yintercept = mu,color = "red",linetype = "dashed",size = 1.2) +
    xlab("Nº muestra") + ylab("IC") + scale_color_manual(values =
    c("black","red")) +
    theme(axis.title.y = element_text(size = 18),
          axis.title.x = element_text(size = 18),
          legend.position = "none")

  print(p)
}
```

Código A.35: Código utilizado para la obtención de la Figura 5.1.

- El siguiente código fue utilizado aquí:

```
# Intervalo de confianza bootstrap para lambda (exponencial)
set.seed(1)
n <- 30
muestra <- rexp(n,0.1)
alfa <- 0.1

lambda_MV <- 1/mean(muestra)
muestra_sim <- c(muestra,2*mean(muestra) - muestra)
h <- density(muestra)$bw
```

```r

B <- 1e5
T_boot <- replicate(B,{
  remuestra_u <- sample(muestra,n,replace = TRUE)
  remuestra_p <- rexp(n,lambda_MV)
  remuestra_sim <- sample(muestra_sim,n,replace = TRUE)
  remuestra_su <- sample(muestra,n,replace = TRUE) + h/2*rnorm(n)

    c(1/mean(remuestra_u),1/mean(remuestra_p),1/mean(remuestra_sim),1/mean(remuestra_su))
})
T_boot <- T_boot - lambda_MV

metodo <- c("unif","param","sim","suav")
for (i in 1:length(metodo)){
  c1 <- quantile(T_boot[i,],alfa/2)
  c2 <- quantile(T_boot[i,],1 - alfa/2)
  print(paste0(metodo[i],": IC(lambda) = [",lambda_MV - c2,",",lambda_MV - c1,"]"))
}
d1 <- qgamma(alfa/2,n,n)
d2 <- qgamma(1 - alfa/2,n,n)
print(paste0("exacto: IC(lambda) = [",d1*lambda_MV,",",d2*lambda_MV,"]"))
```

Código A.36: Código utilizado para la obtención de los resultados de la Tabla 5.4.

- El siguiente código fue utilizado aquí:

```r
# Cobertura y longitud de intervalos de confianza bootstrap y exacto método
# percentil básico
ICboot <- function(muestra,metodo,B,alfa,T_muestra){
  n <- length(muestra)
  if (metodo == "unif"){
    T_boot <- replicate(B,{
      remuestra <- sample(muestra,n,replace = TRUE)
      1/mean(remuestra)
    })
  }
  if (metodo == "param"){
    T_boot <- replicate(B,{
      remuestra <- rexp(n,T_muestra)
```

```
      1/mean(remuestra)
    })
  }
  if (metodo == "sim"){
    muestra_sim <- c(muestra,2*mean(muestra) - muestra)
    T_boot <- replicate(B,{
      remuestra <- sample(muestra_sim,n,replace = TRUE)
      1/mean(remuestra)
    })
  }
  if (metodo == "suav"){
    h <- density(muestra)$bw
    T_boot <- replicate(B,{
      remuestra <- sample(muestra,n,replace = TRUE) + h/2*rnorm(n)
      1/mean(remuestra)
    })
  }

  T_boot <- T_boot - T_muestra
  c1 <- quantile(T_boot,alfa/2,names = FALSE)
  c2 <- quantile(T_boot,1 - alfa/2,names = FALSE)

  c(T_muestra - c2,T_muestra - c1)
}

set.seed(1)
n_muestra <- 1000
n <- 30

alfa <- 0.1
B <- 1e3
ICs <- replicate(n_muestra,{
  muestra <- rexp(n,0.1)
  lambda_MV <- 1/mean(muestra)

  IC1 <- ICboot(muestra,"unif",B,alfa,lambda_MV)
  IC2 <- ICboot(muestra,"param",B,alfa,lambda_MV)
```

```
  IC3 <- ICboot(muestra,"sim",B,alfa,lambda_MV)
  IC4 <- ICboot(muestra,"suav",B,alfa,lambda_MV)

  d1 <- qgamma(alfa/2,n,n)
  d2 <- qgamma(1 - alfa/2,n,n)

  c(c(d1*lambda_MV,d2*lambda_MV),IC1,IC2,IC3,IC4)
})

# Cobertura de los intervalos
lambda <- 0.1
metodo <- c("exacto","unif","param","sim","suav")
for (i in 1:length(metodo)){
  print(paste0("Cobertura de ",metodo[i]," : ",mean(ICs[2*i - 1,] <= lambda &
    lambda <= ICs[2*i,])))
}

# Longitud de los intervalos
for (i in 1:length(metodo)){
  print(paste0("Longitud de ",metodo[i]," : ",mean(ICs[2*i,] - ICs[2*i - 1,])))
}
```

Código A.37: Código utilizado para la obtención de los resultados de la Tabla 5.5.

- El siguiente código fue utilizado aquí:

```
# Plot de los intervalos de confianza
library(tidyverse)
for (i in 1:length(metodo)){
  df <- data.frame(x = 1:100,
                   L = ICs[2*i - 1,1:100],
                   U = ICs[2*i,1:100],
                   L_not_in = ICs[2*i - 1,1:100] > lambda | lambda >
    ICs[2*i,1:100],
                   U_not_in = ICs[2*i - 1,1:100] > lambda | lambda >
    ICs[2*i,1:100])

  p <- ggplot(data = df,aes(x = x)) +
    geom_errorbar(aes(ymax = U, ymin = L,color = L_not_in)) +
```

```
    geom_hline(yintercept = lambda,color = "red",linetype = "dashed",size = 1.2) +
    xlab("Nº muestra") + ylab("IC") + scale_color_manual(values =
    c("black","red")) +
    theme(axis.title.y = element_text(size = 18),
          axis.title.x = element_text(size = 18),
          legend.position = "none")

  print(p)
}
```

Código A.38: Código utilizado para la obtención de la Figura 5.2.

- El siguiente código fue utilizado aquí:

```
# Intervalo de confianza bootstrap para lambda (Poisson)
set.seed(1)
n <- 35
muestra <- rpois(n,2)
alfa <- 0.1

lambda_MV <- mean(muestra)
muestra_sim <- c(muestra,2*mean(muestra) - muestra)
h <- density(muestra)$bw

B <- 1e5
T_boot <- replicate(B,{
  remuestra_u <- sample(muestra,n,replace = TRUE)
  remuestra_p <- rpois(n,lambda_MV)
  remuestra_sim <- sample(muestra_sim,n,replace = TRUE)
  remuestra_su <- sample(muestra,n,replace = TRUE) + h/2*rnorm(n)
  c(mean(remuestra_u),mean(remuestra_p),mean(remuestra_sim),mean(remuestra_su))
})
T_boot <- T_boot - lambda_MV

metodo <- c("unif","param","sim","suav")
for (i in 1:length(metodo)){
  c1 <- quantile(T_boot[i,],alfa/2)
  c2 <- quantile(T_boot[i,],1 - alfa/2)
  print(paste0(metodo[i],": IC(lambda) = [",lambda_MV - c2,",",lambda_MV - c1,"]"))
```

```
}
d1 <- qnorm(alfa/2)
d2 <- qnorm(1 - alfa/2)
print(paste0("asintotico: IC(lambda) = [",(sqrt(lambda_MV) -
    d2/(2*sqrt(n)))**2,",",(sqrt(lambda_MV) - d1/(2*sqrt(n)))**2,"]"))
```

Código A.39: Código utilizado para la obtención de la Tabla 5.7.

- El siguiente código fue utilizado aquí:

```
# Cobertura y longitud de intervalos de confianza bootstrap y exacto método
# percentil básico
ICboot <- function(muestra,metodo,B,alfa,T_muestra){
  n <- length(muestra)
  if (metodo == "unif"){
    T_boot <- replicate(B,{
      remuestra <- sample(muestra,n,replace = TRUE)
      mean(remuestra)
    })
  }
  if (metodo == "param"){
    T_boot <- replicate(B,{
      remuestra <- rpois(n,T_muestra)
      mean(remuestra)
    })
  }
  if (metodo == "sim"){
    muestra_sim <- c(muestra,2*mean(muestra) - muestra)
    T_boot <- replicate(B,{
      remuestra <- sample(muestra_sim,n,replace = TRUE)
      mean(remuestra)
    })
  }
  if (metodo == "suav"){
    h <- density(muestra)$bw
    T_boot <- replicate(B,{
      remuestra <- sample(muestra,n,replace = TRUE) + h/2*rnorm(n)
      mean(remuestra)
    })
```

```
  }

  T_boot <- T_boot - T_muestra
  c1 <- quantile(T_boot,alfa/2,names = FALSE)
  c2 <- quantile(T_boot,1 - alfa/2,names = FALSE)

  c(T_muestra - c2,T_muestra - c1)
}

set.seed(1)
n_muestra <- 1000
n <- 35

alfa <- 0.1
B <- 1e3
ICs <- replicate(n_muestra,{
  muestra <- rpois(n,2)
  lambda_MV <- mean(muestra)

  IC1 <- ICboot(muestra,"unif",B,alfa,lambda_MV)
  IC2 <- ICboot(muestra,"param",B,alfa,lambda_MV)
  IC3 <- ICboot(muestra,"sim",B,alfa,lambda_MV)
  IC4 <- ICboot(muestra,"suav",B,alfa,lambda_MV)

  d1 <- qnorm(alfa/2)
  d2 <- qnorm(1 - alfa/2)

  c(c((sqrt(lambda_MV) - d2/(2*sqrt(n)))**2,(sqrt(lambda_MV) -
    d1/(2*sqrt(n)))**2),IC1,IC2,IC3,IC4)
})

# Cobertura de los intervalos
lambda <- 2
metodo <- c("asintotico","unif","param","sim","suav")
for (i in 1:length(metodo)){
  print(paste0("Cobertura de ",metodo[i]," : ",mean(ICs[2*i - 1,] <= lambda &
    lambda <= ICs[2*i,])))
```

```
}

# Longitud de los intervalos
for (i in 1:length(metodo)){
  print(paste0("Longitud de ",metodo[i]," : ",mean(ICs[2*i,] - ICs[2*i - 1,])))
}
```

Código A.40: Código utilizado para la obtención de la Tabla 5.8.

- El siguiente código fue utilizado aquí:

```
# Plot de los intervalos de confianza
library(tidyverse)
for (i in 1:length(metodo)){
  df <- data.frame(x = 1:100,
                   L = ICs[2*i - 1,1:100],
                   U = ICs[2*i,1:100],
                   L_not_in = ICs[2*i - 1,1:100] > lambda | lambda > ICs[2*i,1:100],
                   U_not_in = ICs[2*i - 1,1:100] > lambda | lambda > ICs[2*i,1:100])

  p <- ggplot(data = df,aes(x = x)) +
    geom_errorbar(aes(ymax = U, ymin = L,color = L_not_in)) +
    geom_hline(yintercept = lambda,color = "red",linetype = "dashed",size = 1.2) +
    xlab("Nº muestra") + ylab("IC") + scale_color_manual(values = c("black","red")) +
    theme(axis.title.y = element_text(size = 18),
          axis.title.x = element_text(size = 18),
          legend.position = "none")

  print(p)
}
```

Código A.41: Código utilizado para la obtención de la Figura 5.3.