Índice

Escuela Politécnica Nacional

Estadística Matemática

Simulación Montecarlo y Bootstrap: Indicador de masa corporal (IMC)

Realizado por:

Echanique Muñoz Debbie Elizabeth

Molina Morales Geoconda Dennisse

Encarnación Apolo Milton Fabián

Febrero 2024

1 Introducción:

La Organización Panamericana de la Salud (OPS) y la Organización para la Alimentación y la Agricultura de Naciones Unidas (FAO) publicaron que el 58% de los habitantes de Latinoamérica sufren de sobrepeso,lo cual se debe a la modificación de los patrones alimenticios y el aumento excesivo del consumo de alimentos procesados en los últimos años. La ONU estima que en el año 2025, 167 millones de personas se verán afectadas por sobrepeso. Por otra parte,cabe recalcar que se ha visto una disminución del 13% en la tasa de desnutrición entre los años 1990 y 2016.

La obesidad al igual que la malnutrición son dos trastornos alimenticios que afectan a la población. Para determinar si una persona sufre de alguno de estos trastornos uno de lo indicadores más utilizados es el indicador de masa corporal,por lo que en el presente trabajo estudiaremos la distribución del mismo y estableceremos intervalos que nos permitan determinar si una persona tiene un peso adecuado de acuerdo a su altura.

2 Planteamiento del Problema

El indicador de masa corporal es una medida internacional que nos permite determinar el estado nutricional de una persona, por lo que es una herramienta muy útil en la rama de la investigación epidemiológica, ya que niveles altos o bajos del IMC se asocian a enfermedades tales como:obesidad, desnutrición, enfermedades cardíacas, diabetes de tipo 2,entre otras. Sin embargo, en las últimas décadas se ha visto un descenso en la malnutrición de la población y un aumento de sobrepeso en algunas regiones, lo cual afectaría la distribución global del IMC, por esta razón se desea establecer intervalos de confianza para la media, el mínimo y el máximo del IMC, de tal forma que nos permitan determinar si una persona se encuentra dentro del rango normal de peso de acuerdo a su altura.

La metodología que se utilizará es simulación por montecarlo para poder encontrar la distribución del indicador de masa corporal a partir de las alturas y el peso.Luego, se encontrará la distribución por remuestreo bootstrap del indicador, para poder obtener la estimación del sesgo, precisión y los intervalos de confianza.

3 Objetivos

3.1 Objetivo General

1. Determinar si el peso de una persona es el adecuado de acuerdo a su altura.

3.2 Objetivos Específicos:

1. Encontrar la distribución del indicador de masa corporal a partir de los pesos y alturas usando la simulación de montecarlo.
2. Encontrar la distribución del indicador por remuestreo bootstrap.
3. Determinar los intervalos para el mínimo, máximo y la media del indicador de masa corporal.

4 Metodología

El indicador IMC es una medida que se utiliza para evaluar el peso corporal en relación con la estatura. Así, para determinar el valor esperados del IMC, es decir, el promedio esperado del IMC en la base de datos seleccionada, se porcedió a utilizar el método Monte Carlo Paramétrico y No Paramétrico.

4.1 Montecarlo

Montecarlo es una técnica estadística que se utiliza para aproximar cantidades desconocidas mediante la generación de números/muestras aleatorios.

4.1.1 Montecarlo Paramétrico

Para el caso particular del presente proyecto se realizarón los siguientees pasos, que son los pasos a seguir al utilizar el Método de Montecarlo Paramétrico.

Se ajustó un modelo multinormal bivariado a los datos del IMC selecionados, para después generar muestras aleatorias usando las medias y las varianzas del modelo ajustado anteriormente.

Con la obtención de las muestras aleatorias se procedió a calcular el IMC para cada muestra aleatoria construida para así calcular el promedio de estos IMC como la estimación del valor esperado del IMC.

4.1.2 Monte Carlo NO Paramétrico

Para este caso del método de Monte Carlo, se generaron muestras aleatorias de los datos de IMC existentes sin hacer suposiciones sobre su distribución, para ello, se utilizó la función *sample* implementada en el software R, el cual permite extraer una muestra aleatoria de elementos de un conjunto de datos, para después calcular el IMC para cada muestra aleatoria generada en el paso anterior, así, se calculó el promedio de los IMC anteriores como el valor esperado del IMC.

Matemáticamente se puede ver el método de la siguiente forma:

$$E[f(X)] \approx \frac{1}{N} \sum_{i=1}^{N} f(x_i)$$

4.2 Bootstrap

En lugar de hacer suposiciones sobre la distribución subyacente de los datos, Bootstrap utiliza los datos observados para aproximar la distribución de un estadístico de interés o para realizar inferencias sobre los parámetros.

Ell proceso de remuestreo en Bootstrap implica tomar muestras aleatorias, con reemplazo, del conjunto de datos original. Es decir, en cada iteración del proceso de bootstrap, se seleccionan aleatoriamente observaciones del conjunto de datos original y estas observaciones se incluyen en la muestra de bootstrap.

Por lo que permite estimar intervalos de confianza, realizar pruebas de hipótesis y realizar otras inferencias estadísticas sin depender de suposiciones sobre la distribución de los datos.

4.2.1 Bootstrap Paramétrico

El bootstrap paramétrico es una técnica de remuestreo que asume que los datos provienen de una distribución paramétrica conocida. Supongamos que tenemos una muestra $X_1, X_2, \ldots, X_n$ de una distribución con parámetros desconocidos θ.

Ajuste del modelo paramétrico:

Sea $f(X;\theta)$ la función de densidad de probabilidad (o función de masa de probabilidad) que describe la distribución de los datos observados X, donde θ es el vector de parámetros a estimar. El ajuste del modelo paramétrico implica encontrar los estimadores de máxima verosimilitud (MLE) o algún otro método de estimación de los parámetros $\hat{\theta}$.

$$\hat{\theta} = \arg\max_{\theta} \prod_{i=1}^{n} f(x_i; \theta)$$

Remuestreo con reemplazo:

$$X^{*(b)} = \{X^*_{1(b)}, X^*_{2(b)}, \ldots, X^*_{n(b)}\}, \quad b = 1, 2, \ldots, B$$

Cálculo de la estadística de interés:

$$\theta^{*(b)} = g(X^{*(b)}), \quad b = 1, 2, \ldots, B$$

donde $g(\cdot)$ representa la función que calcula el estadístico de interés a partir de la muestra bootstrap.

Aproximación de la distribución de remuestreo:

$$\hat{F}_{\theta^*}(\theta) = \frac{1}{B} \sum_{b=1}^{B} I(\theta^{*(b)} \leq \theta)$$

donde $\hat{F}_{\theta^*}(\theta)$ es la función de distribución empírica de las estadísticas bootstrap, $\theta^{*(b)}$ es la estadística de interés calculada a partir de la muestra bootstrap b, y $I(\cdot)$ es la función indicadora que toma el valor 1 si la condición es verdadera y 0 en caso contrario.

4.2.2 Bootstrap No Paramétrico

El bootstrap no paramétrico es una técnica de remuestreo que no hace suposiciones sobre la distribución subyacente de los datos. Para estimar el error estándar de un estimador $\hat{\theta}$ sin suponer una forma específica de distribución.

Remuestreo con reemplazo:

$$X^{*(b)} = \{X^*_{1(b)}, X^*_{2(b)}, \ldots, X^*_{n(b)}\}, \quad b = 1, 2, \ldots, B$$

Cálculo de la estadística de interés:

$$\theta^{*(b)} = g(X^{*(b)}), \quad b = 1, 2, \ldots, B$$

donde $g(\cdot)$ representa la función que calcula la estadística de interés a partir de la muestra bootstrap.

Aproximación de la distribución de remuestreo:

$$\hat{F}_{\theta^*}(\theta) = \frac{1}{B} \sum_{b=1}^{B} I(\theta^{*(b)} \leq \theta)$$

donde $\hat{F}_{\theta^*}(\theta)$ es la función de distribución empírica de las estadísticas bootstrap, $\theta^{*(b)}$ es la estadística de interés calculada a partir de la muestra bootstrap b, y $I(\cdot)$ es la función indicadora que toma el valor 1 si la condición es verdadera y 0 en caso contrario.

Debido a lo mencionado anteriormente se puede comparar los resultados obtenidos mediante Bootstrap con los obtenidos mediante Monte Carlo calculando el sesgo y la precisión de la estimación utilizando Bootstrap, para luego, compara estos resultados con los obtenidos mediante Monte Carlo, se calcula intervalos de confianza utilizando los dos métodos y se compara su tamaño y cobertura. Esto permite evaluar la concordancia entre los métodos en la estimación de la incertidumbre asociada con el estadístico de interés.

4.3 Sesgo y varianza

Consideremos $\hat{\theta} = T(L)$, y F la distribución conocida antes mencionada, se define el estadístico:

$$R(L, F) = \hat{\theta} - \theta$$

Así, matemáticamente se desea encontrar

$$Sesgo(\hat{\theta}) = E(\hat{\theta} - \theta) = E(R)Var(\theta) = Var(\hat{\theta} - \theta)$$

Para la implementación del presente proyecto se realizó, la técnica Bootstrap para aproximar el sesgo $Sesgo^*(\hat{\theta^*})$ y $Var^*(\hat{\theta^*})$ de la manera previamente mencionada.

4.4 Intervalos de confianza

Una parte fundamental de cualquier estudio es la creación de los intervalos de confianza para los parámetros de interés, para ello se utilizó el método percentil.

4.4.1 Método percentil

El método percentil-t se utiliza para construir intervalos de confianza cuando la distribución de los datos no es normal o cuando el tamaño de la muestra es pequeño, lo cual, es justamente lo que los datos seleccionados requieren pues se ajustó una mixtura de 3 normales.

Se basa en la distribución t de Student y utiliza los percentiles de esta distribución para determinar los límites del intervalo de confianza.

Así, la fórmula para calcular el intervalo de confianza utilizando el método percentil-t es:

$$IC = \bar{x} \pm t_{\frac{\alpha}{2}, n-1} \times \frac{s}{\sqrt{n}}$$

donde $\bar{x}$ es la media muestral, s es la desviación estándar muestral, n es el tamaño de la muestra y $t_{\frac{\alpha}{2}, n-1}$ es el valor crítico de la distribución t de Student con $n-1$ grados de libertad y un nivel de significancia α.

4.5 Contraste de hipótesis

Un contraste de hipótesis es una prueba estadística utilizada para tomar decisiones sobre una afirmación basada en datos observados.

Involucra la formulación de una hipótesis nula (H_0) y una hipótesis alternativa (H_1) , y la recolección de datos para determinar si hay suficiente evidencia para rechazar la hipótesis nula.

Como idea para contrastar pruebas de hipótesis, el estadístico de contraste se calcula a partir de los datos y se compara con un valor crítico o un p-valor para tomar una decisión sobre la hipótesis nula.

Para el presente proyecto se realizó un contraste de hipótesis para la media del indicador IMC.

Hipótesis nula y alternativa

- La hipótesis nula (H_0) establece que no hay diferencia significativa entre la media poblacional y un valor específico. Por ejemplo, $H_0 : \mu = \mu_0$, donde μ es la media poblacional y μ_0 es el valor hipotético que estamos contrastando.
- La hipótesis alternativa (H_1) puede ser de dos tipos:

- Si estamos interesados en detectar si la media poblacional es mayor que un valor específico, entonces $H_1 : \mu > \mu_0$.
- Si estamos interesados en detectar si la media poblacional es menor que un valor específico, entonces $H_1 : \mu < \mu_0$.
- Si estamos interesados en detectar si la media poblacional es diferente de un valor específico, entonces $H_1 : \mu \neq \mu_0$.

Estadística de contraste

- La estadística de contraste se basa en la muestra de datos y se utiliza para evaluar qué tan probable es obtener los datos observados bajo la hipótesis nula.
- Para contrastar sobre la media, una estadística común es la t de Student, que se calcula como:

$$t = \frac{\bar{x} - \mu_0}{s/\sqrt{n}}$$

donde $\bar{x}$ es la media muestral, s es la desviación estándar muestral y n es el tamaño de la muestra.

Decisión

- Utilizamos la estadística de contraste para calcular el valor p o para compararlo con un valor crítico.
- Si el valor p es menor que el nivel de significancia α, rechazamos la hipótesis nula y concluimos que hay evidencia suficiente para afirmar la hipótesis alternativa.
- Si la estadística de contraste cae dentro de la región de rechazo determinada por el valor crítico, también rechazamos la hipótesis nula.

4.6 Algoritmo Expectation-Maximization

(EM)

El **algoritmo EM** es una técnica iterativa utilizada para estimar los parámetros de un modelo estadístico cuando los datos están incompletos o involucran variables latentes no observadas. A menudo se aplica en situaciones donde se trabaja con **distribuciones mixtas**, como las que provienen de la mezcla de distribuciones normales.

4.6.1 Pasos Fundamentales:

1. **Expectativa (E-Step)**:
   - Se inicia con una **estimación inicial** de los parámetros del modelo.
   - Se crea una **distribución de probabilidad** basada en estos parámetros.
   - Esta etapa se conoce como el **paso E** o "expectativa".
2. **Maximización (M-Step)**:
   - Se introducen los **datos observados** en el modelo.
   - La distribución de probabilidad del paso E se **ajusta** para incorporar los nuevos datos.
   - Esta etapa se llama el **paso M**.
3. **Iteración**:
   - Los pasos 1 y 2 se repiten hasta que se alcance la **convergencia** (es decir, cuando la distribución deja de cambiar significativamente).
4. **Convergencia**:
   - El algoritmo EM mejora iterativamente la estimación de los parámetros.
   - A veces, se requieren **inicios aleatorios** para encontrar el mejor modelo, ya que el algoritmo puede converger hacia un máximo local en lugar del máximo global óptimo.

4.6.2 Base Matemática:

1. **Función de Verosimilitud Completa (Full Likelihood)**:

   - La función de verosimilitud completa representa la probabilidad conjunta de los datos observados y las características no observadas.

   - Se denota como $L(\theta)$, donde θ representa los parámetros del modelo.

   - La expresión general es:

$$L(\theta) = P(\text{datos observados}, \text{características no observadas}|\theta)$$

2. **Función de Verosimilitud Marginal (Marginal Likelihood)**: - La función de verosimilitud marginal se obtiene al marginalizar sobre las características no observadas. - Representa la probabilidad de observar solo los datos observados, sin tener en cuenta las características no observadas. - Se denota como $Q(\theta|\theta^{(t)})$, donde $\theta^{(t)}$ es la estimación actual de los parámetros en la iteración t. - La expresión es:

$$Q(\theta|\theta^{(t)}) = \sum_{\text{características no observadas}} P(\text{datos observados}, \text{características no observadas}|\theta)$$

3. **Actualización de Parámetros**: - En cada iteración, se actualizan los parámetros del modelo utilizando la función de verosimilitud marginal. - La actualización se realiza mediante la maximización de $Q(\theta|\theta^{(t)})$. - La expresión para actualizar los parámetros θ es:

$$\theta^{(t+1)} = \arg\max_{\theta} Q(\theta|\theta^{(t)})$$

4. **Convergencia**: - El algoritmo EM continúa iterando hasta que los parámetros convergen a un valor estable. - La convergencia se alcanza cuando los parámetros no cambian significativamente entre las iteraciones.

5 Simulación

\subsection Considerando las personas que tienen un índice de masa corporal (IMC) dentro de un rango normal, es decir dentro de 18.5 y 24.9. Por tanto se tiene una base de 342 individuos, los cuales se pueden representar en los grupos etarios mostrados en el Cuadro 1.

Grupo etario	**Porcentaje**
12-18 años	13.16%
19-30 años	54.39%
31-60 años	32.45%

Table 1: Porcentaje de grupo etario para IMC Normal

Se ajustará una distribución multivariante para las variables **peso** y **altura**. Para esto se grafica la distribución multivariante en la Figura 1

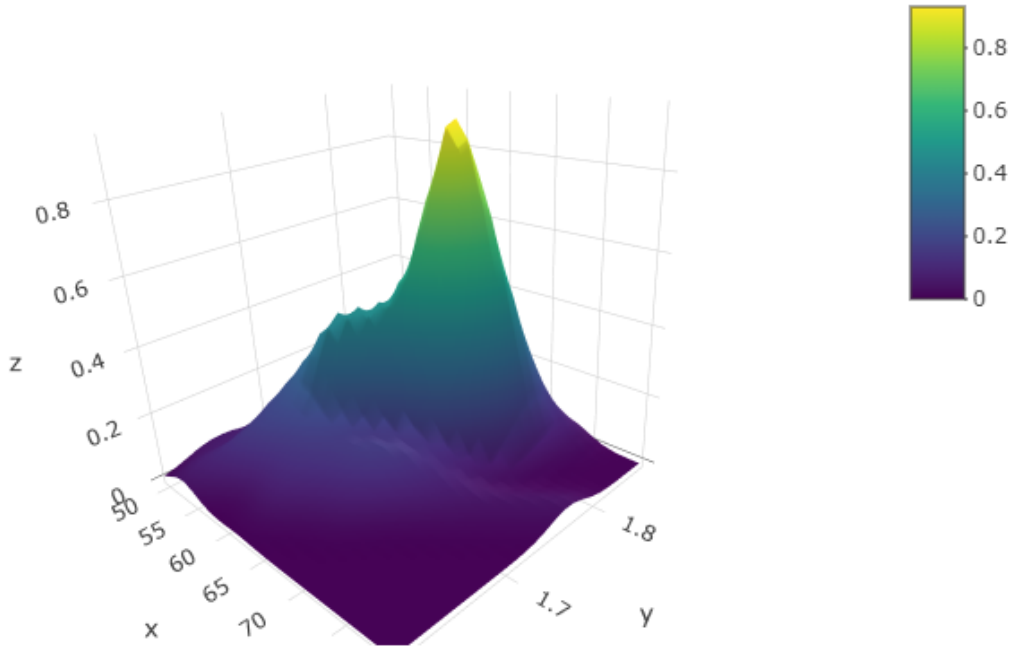

Figure 1: Distribución Multivariante del peso y altura

Mediante la gráfica se puede ver que los datos parecen seguir una mixtura de normales multivariantes, para tenerlo más claro se graficará en 2 dimensiones la distribución del indicador, como se tiene en la Figura 2.

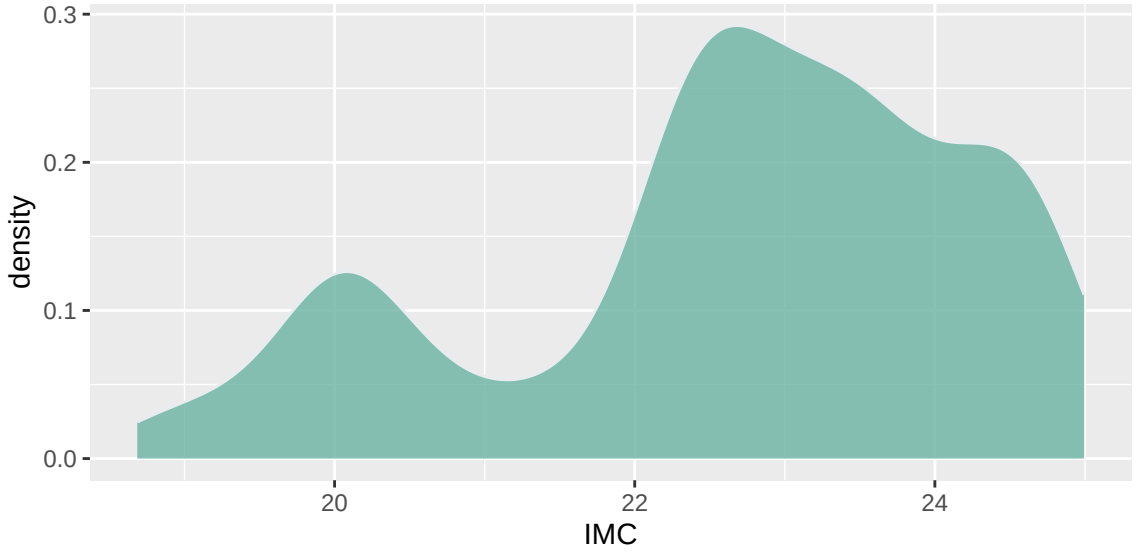

Figure 2: Distribución del peso y altura

Vemos que probablemente puede existir una mixtura de 3 normales multivariantes. Mediante el algoritmo EM se encuentran los parámetros de esta distribución. Se presentan las medias de las subpoblaciones encontradas:

$$\mu_1 = [59.48, 1.73] \quad \mu_2 = [66.58, 1.70] \quad \mu_3 = [72.63, 1.75]$$

Con las siguientes matrices de covarianzas:

$$\Sigma_1 = \begin{bmatrix} 20.997 & 0.256 \\ 0.256 & 0.003 \end{bmatrix}$$

$$\Sigma_2 = \begin{bmatrix} 29.536 & 0.218 \\ 0.218 & 0.002 \end{bmatrix}$$

$$\Sigma_3 = \begin{bmatrix} 4.85 & 0.009 \\ 0.009 & 0.0003 \end{bmatrix}$$

Y los pesos de la subpoblaciones $\lambda_1 = 0.174$, $\lambda_2 = 0.509$ y $\lambda_3 = 0.315$.

Podemos visualizar la densidad encontrada mediante la Figura 3.

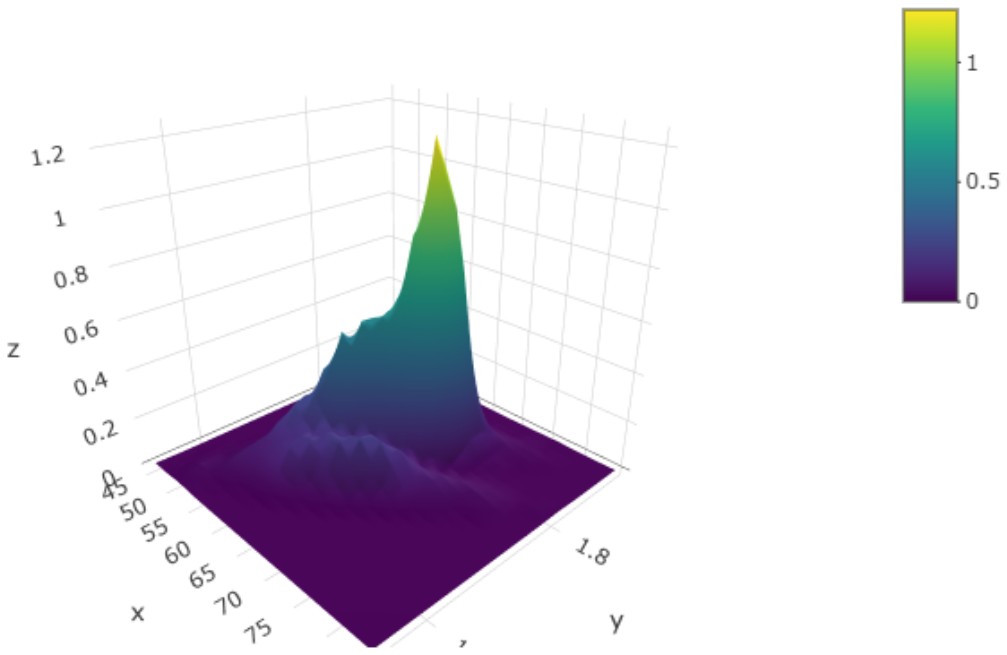

Figure 3: Distribución mixtura normal Multivariante encontrada

Ahora, mediante el Test Anderson-Darling, que fue adaptado para la mixtura de distribuciones normales multivariantes se tiene un valor p de 0.7483, por lo tanto no se rechaza la hipótesis nula. Entonces los datos provienen de una distribución de mixturas de normales multivariantes con los parámetros estimados.

Con la distribución exacta encontrada, se simula por Montecarlo los valores de las variables y se opera para obtener el indicador IMC y posteriormente su distribución. También se encontrará la distribución del indicador mediante bootstrap y se realiza una comparación. La Figura 4 muestra la comparación respectiva, además de comparar con la de los datos.

Las estimaciones de la media y el sesgo para los dos métodos están dados por el siguiente Cuadro:

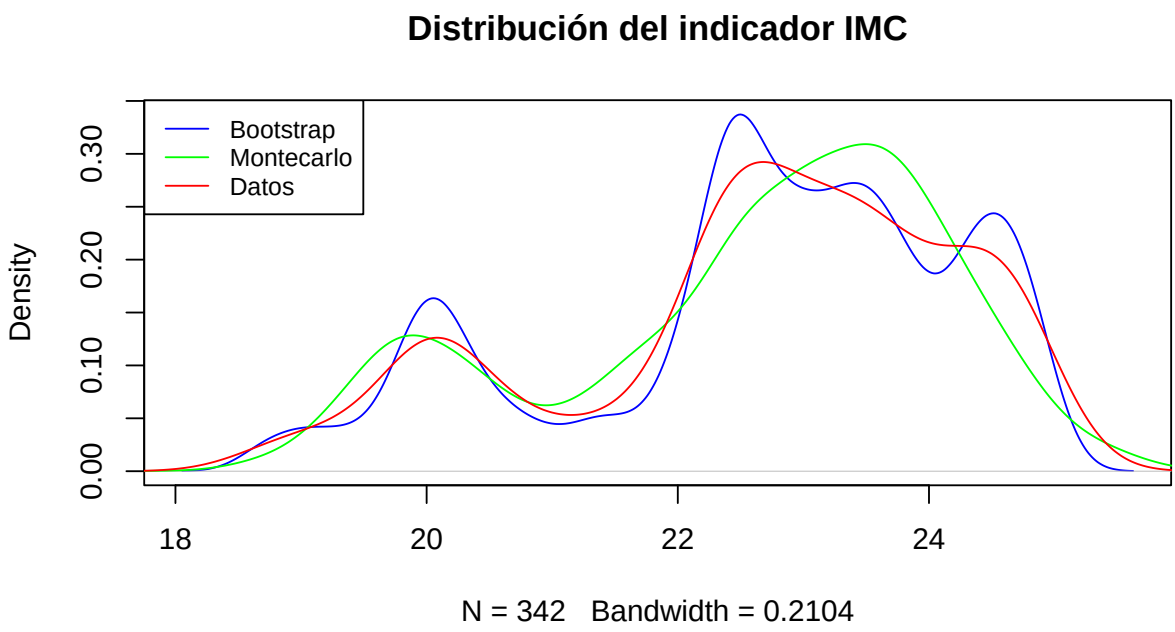

Figure 4: Distribución indicador IMC

Método	Estimación	Sesgo
Montecarlo	22.64392	-0.0017
Bootstrap	22.6483	0.0026

Table 2: Estimaciones para la Media

Las estimaciones para la varianza se pueden ver en el Cuadro siguiente:

Método	Estimación	Sesgo
Montecarlo	2.4965	0.0013
Bootstrap	2.4917	-0.003

Table 3: Estimaciones para la Varianza

Y por último las estimaciones para la desviación estándar están dadas por:

Método	Estimación	Sesgo
Montecarlo	1.5768	-0.0028
Bootstrap	1.5776	-0.0020

Table 4: Estimaciones para la Desviación estándar

Ahora se construyen intervalos de confianza al 95% de confiabilidad para la media por Montecarlo y por el Método percentil-t (bootstrap). Los resultados se muestran en el siguiente Cuadro:

Método	Inferior	Superior
Montecarlo	22.33871	22.93689
Bootstrap	22.46823	22.80890

Table 5: Intervalos de confianza para la media

Se tiene además que la precisión por Montecarlo es de 0.0021, y la precisión por Bootstrap es de 0.0854. Con esto se puede concluir que el 95% de las veces la media del indicador de IMC, debe encontrarse entre 22.47 y 22.81 Por último se estudiará el siguiente contraste de hipótesis:

$$\begin{cases} H_0: & \mu = 22.64 \\ H_1: & \mu \neq 22.64 \end{cases}$$

Para ello, se realizará 5000 pruebas para obtener la proporción de valores p, y además se graficará el tamaño del contrastre.

Para Montecarlo se tienen los resultados:

- Proporción de rechazos al 1%: 0.0656
- Proporción de rechazos al 5%: 0.1936
- Proporción de rechazos al 10%: 0.2926

Y el tamaño del contraste está dado por:

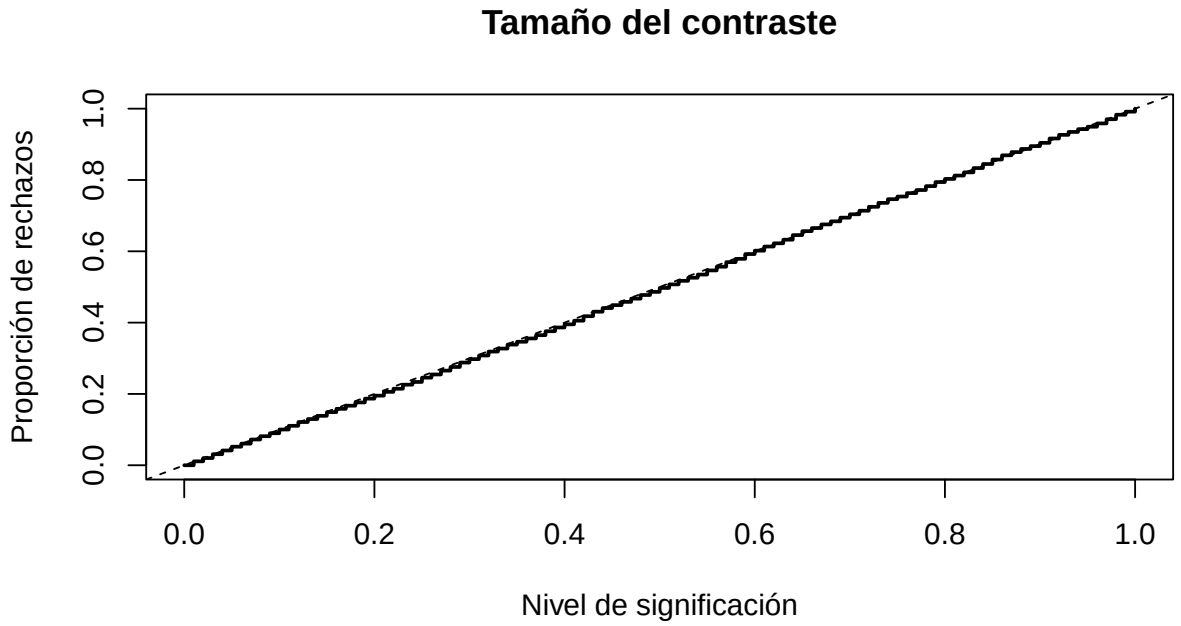

Figure 5: Tamaño del Contraste por Montecarlo

Por lo tanto no rechazamos la hipótesis nula planteada para la media.

Para Bootstrap se tienen los resultados:

- Proporción de rechazos al 1%: 0.0202
- Proporción de rechazos al 5%: 0.093
- Proporción de rechazos al 10%: 0.1634

Y el tamaño del constrate se muestra en la siguiente figura:

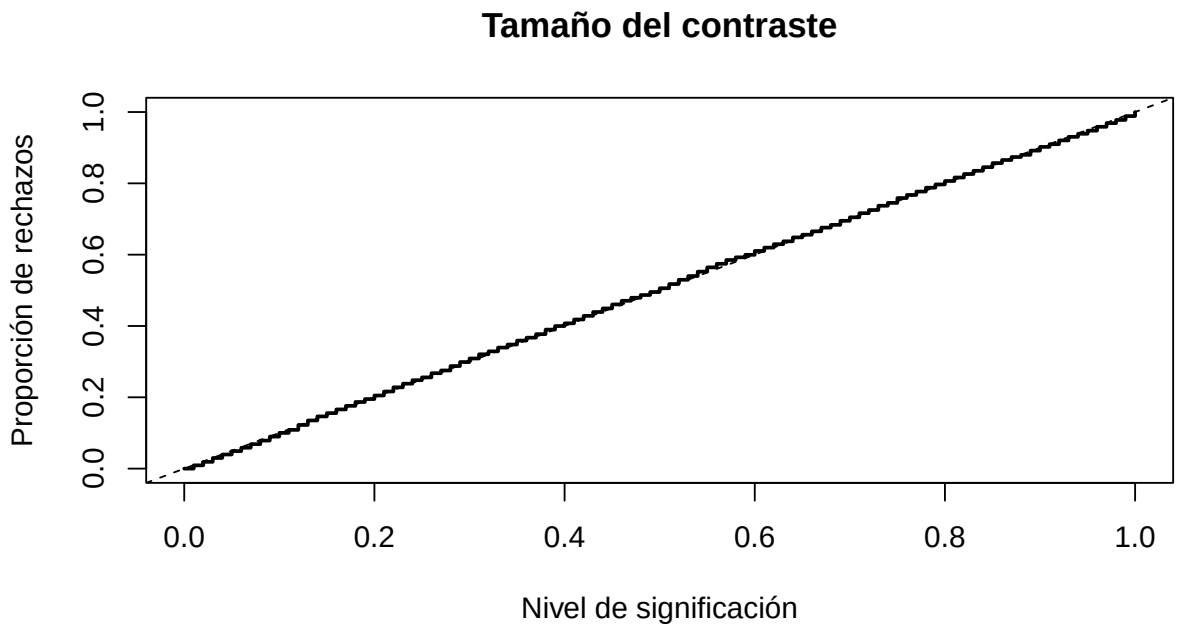

Figure 6: Tamaño del Contraste por Boostrap

De la misma forma, no rechazamos la hipótesis nula.

5.1 Mínimo del indicador IMC

Ahora, para analizar los valores mínimos que puede tomar el indicador IMC se realiza el procedimiento anterior. Se tienen los siguientes intervalos de confianza en la siguiente tabla:

Método	**Inferior**	**Superior**	**Precisión**
Montecarlo	18.19738	19.46157	0.004515656
Bootstrap	18.68512	18.92915	0.05310323

Table 6: Intervalos de confianza para el mínimo IMC

5.2 Máximo del indicador IMC

Análogo a lo anterior, se encontrará los intervalos de confianza para el máximo valor que puede tomar el indicador IMC. Los resultados se muestran en la siguiente tabla:

Método	Inferior	Superior	Precisión
Montecarlo	24.83709	26.35412	0.005388757
Bootstrap	24.91436	24.99417	0.02193493

Table 7: Intervalos de confianza para el máximo IMC

6 Bibliografía

Errandonea, U. (2012). Obesidad y trastornos de alimentación. Revista Médica Clínica las Condes, 23(2), 165-171. https://doi.org/10.1016/s0716-8640(12)70294-8 Gloria, N. –. (2023, 15 julio). ¿Qué es el índice de masa corporal y cómo se mide? La Buena Nutrición. https://labuenanutricion.com/blog/que-es-el-indice-de-masa-corporal-y-como-se-mide/ Konrad, L. T. F., & Emilio, C. M. (s. f.). Distribución del indice de masa corporal (IMC) y prevalencia de obesidad primaria en niños pre-púberes de 6 a 10 años de edad en el distrito de San Martín de Porres - Lima. http://www.scielo.org.pe/scielo.php?script=sci_arttext&pid=S1018-130X2003000300002 Estudian cómo afectan las variaciones en la distribución del IMC en los cambios en la prevalencia de obesidad y peso bajo en el mundo | CIBERISCIII. (s. f.). https://www.ciberisciii.es/noticias/estudian-como-afectan-las-variaciones-en-la-distribucion-del-imc-en-los-cambios-en-la-prevalencia-de-obesidad-y-peso-bajo-en-el-mundo
BBC News Mundo. (2017, 20 enero). El impresionante aumento del sobrepeso y la obesidad en América Latina. BBC News Mundo. https://www.bbc.com/mundo/noticias-38693438#:~:text=El%20sobrepeso%20en%20Am%C3%A9rica%20Latina%20se%20increment%C3%B3%20en,Naciones%20Unidas%20%28FAO%2C%20por%20sus%20siglas%20en%20ingl%C3%A9s%29Hay que acabar con la obesidad, un trastorno que afecta a más de mil millones de personas. (2022, 9 marzo). Noticias ONU. https://news.un.org/es/story/2022/03/1505062#:~:text=Seg%C3%BAn%20los%20c%C3%A1lculos%20de%20la%20Organizaci%C3%B3n%2C%20la%20obesidad,peor%20salud%20por%20motivos%20de%20sobrepeso%20u%20obesidad

7 Anexos

```
###DATA BMI
data<-read.csv("bmi.csv")
data1<-data %>% filter(BmiClass=='Normal Weight')
length((data1 %>% filter(Age<=18))$Age)
length((data1 %>% filter(Age>=19&Age<=30))$Age)
length((data1 %>% filter(Age>=31&Age<=60))$Age)
peso<-data1$Weight
altura<-data1$Height
ind1<-peso/altura^2

###PLOT
den3d <- kde2d(peso, altura)
#persp(den3d, box=FALSE)
plot_ly(x=den3d$x, y=den3d$y, z=den3d$z) %>%add_surface() %>%
layout(xaxis=list(title='Peso'),yaxis=list(title='Altura'))
data1 %>% ggplot(aes(x=Weight/Height^2))+
geom_density(fill="#69b3a2", color="#e9ecef", alpha=0.8)+
labs(x='IMC')

#ALGORITMO EM
set.seed(1)
```

```
matriz<-matrix(c(peso,altura),nrow=342,ncol=2)
colnames(matriz)<-c('Peso','Altura')
em<-mvnormalmixEM(matriz,k=3)
mus<-rbind(em$mu[[1]],em$mu[[2]],em$mu[[3]])
sigmas<-rbind(em$sigma[[1]],em$sigma[[2]],em$sigma[[3]])
lambdas<-as.vector(em$lambda)
#PLOT EM
set.seed(1)
dat<-rmvnorm.mixt(5000, mus=mus, Sigmas=sigmas, props=as.vector(lambdas))
den3d <- kde2d(dat[,1],dat[,2])
#persp(den3d, box=FALSE)
plot_ly(x=den3d$x, y=den3d$y, z=den3d$z) %>% add_surface()


###TEST AD
testAD<-function(data,mus,sigmapob,lambdapob){
  if (!is.data.frame(data) && !is.matrix(data))
    stop('data supplied must be either of class \"data frame\" or \"matrix\"')
  if (dim(data)[2] < 2 || is.null(dim(data)))
  {stop("data dimesion has to be more than 1")}
  if (dim(data)[1] < 3) {stop("not enough data for assessing mvn")}
  data.name <- deparse(substitute(data))
  xp <- as.matrix(data)
  p <- dim(xp)[2]
  n <- dim(xp)[1]
  ## getting MLEs...
  s.mean <- colMeans(xp)
  s.cov <- (n-1)/n*cov(xp)
  s.cov.inv <- solve(s.cov) # inverse matrix of S (matrix of sample covariances)
  D <- rep(NA,n) # vector of (Xi-mu)'S^-1(Xi-mu)...
  for (j in 1:n)
    D[j] <- t(xp[j,]-s.mean)%*%(s.cov.inv%*%(xp[j,]-s.mean))
  D.or <- sort(D) ## get ordered statistics
  Gp <- pchisq(D.or,df=p)
  ## getting the value of A-D test...
  ind <- c(1:n)
  an <- (2*ind-1)*(log(Gp[ind])+log(1 - Gp[n+1-ind]))
  AD <- -n - sum(an) / n
  ## getting the p-value...
  N <- 1e4
  U <- rep(0,N) ## initializing values of the AD test
  for (i in 1:N) { ## loop through N reps
    dat<-rmvnorm.mixt(1000, mus=mus, Sigmas=sigmapob, props=lambdapob)
    mean1 <- colMeans(dat)
    cov1 <- (n-1)/n*cov(dat)
    cov.inv <- solve(cov1) # inverse matrix of S (matrix of sample covariances)
    D <- rep(NA,n) # vector of (Xi-mu)'S^-1(Xi-mu)...
    for (j in 1:n)
      D[j] <- t(data[j,]-mean1)%*%(cov.inv%*%(data[j,]-mean1))
    Gp <- pchisq(sort(D),df=p)
    ## getting the value of A-D test...
    an <- (2*ind-1)*(log(Gp[ind])+log(1 - Gp[n+1-ind]))
    U[i] <- -n - sum(an) / n
```

```
  }
  p.value <- (sum(U >= AD)+1)/(N+1)
  result<-new('ad',AD=AD,p.value=p.value,data.name=data.name)
  result
}
set.seed(1)
test<-testAD(matriz,mus,sigmas,lambdas)

###PLOT GRAFICAS
#montecarlo
indicador_sim<-(dat[,1]/dat[,2]^2)
#bootstrap
x <- ind1
n<-length(x)
h <- bw.SJ(x)
npden <- density(x, bw = h)
plot(npden,col='blue',main = 'Distribución del indicador IMC')
lines(density(indicador_sim),col="green")
lines(density(ind1),col='red')
legend("topleft",legend=c("Bootstrap", "Montecarlo",'Datos'),
       col=c("blue", "green",'red'), lty=1, cex=0.8)


###ESTIMACIONES

#MONTECARLO
nsim<-5000
nx<-100
media<-numeric(nsim)
varianza<-numeric(nsim)
desv<-numeric(nsim)
for(i in 1:5000){
  datos_sim<-rmvnorm.mixt(nx, mus=mus, Sigmas=sigmas, props=as.vector(lambdas))
  indicador_sim<-datos_sim[,1]/datos_sim[,2]^2
  media[i]<-mean(indicador_sim)
  varianza[i]<-var(indicador_sim)
  desv[i]<-sd(indicador_sim)
}
#Estimacion media, varianza, desvest(precision)
estimediaMont<-mean(media)
estivarMont<-mean(varianza)
estidesMont<-mean(desv)
varmedia <- (1/(nsim^2)) * sum((media - mean(media))^2)
sqrt(varmedia)
#Sesgo
sesgoMontmedia<-estimediaMont-mean(ind1)
sesgoMontvar<-estivarMont-var(ind1)
sesgoMontdes<-estidesMont-sd(ind1)


###BOOTSTRAP
x <- ind1
n<-length(x)
```

```r
h <- bw.SJ(x)
npden <- density(x, bw = h)
plot(npden)
range_x<-range(npden$x)

#MEDIA, VARIANZA, PRECISION

B <- 1000
n<-length(ind1)
estadistico_boot <- numeric(B)
var_boot <- numeric(B)
desv_boot <- numeric(B)
for (k in 1:B) {
  remuestra <- sample(ind1, n, replace = TRUE)
  estadistico_boot[k] <- mean(remuestra)
  var_boot[k]<-var(remuestra)
  desv_boot[k]<-sd(remuestra)

}
estimedia_boot <- mean(estadistico_boot)
estivar_boot <- mean(var_boot)
estisd_boot <- mean(desv_boot)
varmediaboot <- (1/B) * sum((estadistico_boot - mean(estadistico_boot))^2)
sqrt(varmediaboot)
#SESGO
sesgobootmedia<-estimedia_boot-mean(ind1)
sesgobootvar<-estivar_boot-var(ind1)
sesgobootdes<-estisd_boot-sd(ind1)


#INTERVALOS DE CONFIANZA
#montecarlo
#Intervalos confianza
#media
alfa <- 0.05
ic<-quantile(media,c(alfa/2, 1 - alfa/2))

#bootstrap

#INTERVALOS - METODO PERCENTIL T
alfa<-0.05
B <- 1000
n<-length(ind1)
remuestra<-numeric(n)
estadistico_boot <- numeric(B)
for (k in 1:B) {
  remuestra <- sample(ind1, n, replace = TRUE)
  x_barra_boot<-mean(remuestra)
  cuasi_dt_boot <- sd(remuestra)
  estadistico_boot[k] <- sqrt(n) * (x_barra_boot - mean(ind1))/cuasi_dt_boot
}
pto_crit <- quantile(estadistico_boot, c(alfa/2, 1 - alfa/2))
# Construcción del IC
```

```r
ic_inf_boot <- mean(ind1) - pto_crit[2] * sd(ind1)/sqrt(n)
ic_sup_boot <- mean(ind1) - pto_crit[1] * sd(ind1)/sqrt(n)
IC_boot <- c(ic_inf_boot, ic_sup_boot)
names(IC_boot) <- paste0(100*c(alfa/2, 1-alfa/2), "%")

###CONSTRASTE DE HIPOTESIS
set.seed(1)
#Montecarlo

#CONSTRASTE DE HIPOTESIS
#Ho: u=22.64
#Ha: u!=22.64
#Constatar normalidad de la media del indicador
nsim<-5000
nx<-1000
pvalor<-numeric(nsim)
#Constrastes
for (i in 1:nsim) {
  datos_sim<-rmvnorm.mixt(nx, mus=mus, Sigmas=sigmas, props=as.vector(lambdas))
  indicador_sim<-datos_sim[,1]/datos_sim[,2]^2
  t<-(mean(indicador_sim)-22.64)/(sd(indicador_sim)/sqrt(nx))
  p.value<-1-pt(abs(t),nx-1)+pt(-abs(t),nx-1)
  pvalor[i]<-p.value
}
{
  cat("\nProporción de rechazos al 1% =", mean(pvalor < 0.01), "\n")
  cat("Proporción de rechazos al 5% =", mean(pvalor < 0.05), "\n")
  cat("Proporción de rechazos al 10% =", mean(pvalor < 0.1), "\n")
}

#bootstrap


#CONTRASTE DE HIPOTESIS

#CONSTRASTE DE HIPOTESIS
#Ho: u=22.64
#Ha: u!=22.64
n<-length(ind1)
nsim<-5000
pvalor2<-numeric(nsim)
#Constrastes
for (i in 1:nsim) {
  remuestra <- sample(ind1, n, replace = TRUE)
  t<-(mean(remuestra)-22.64)/(sd(remuestra)/sqrt(n))
  p.value<-1-pt(abs(t),n-1)+pt(-abs(t),n-1)
  pvalor2[i]<-p.value
}
{
  cat("\nProporción de rechazos al 1% =", mean(pvalor2 < 0.01), "\n")
  cat("Proporción de rechazos al 5% =", mean(pvalor2 < 0.05), "\n")
  cat("Proporción de rechazos al 10% =", mean(pvalor2 < 0.1), "\n")
}
```

```r
#TAMAÑO CONSTRASTE
#montecarlo
curve(ecdf(pvalor)(x), type = "s", lwd = 2,
      main = 'Tamaño del contraste', ylab = 'Proporción de rechazos',
      xlab = 'Nivel de significación')
abline(a=0, b=1, lty=2)
#BOOTSTRAP
curve(ecdf(pvalor2)(x), type = "s", lwd = 2,
      main = 'Tamaño del contraste', ylab = 'Proporción de rechazos',
      xlab = 'Nivel de significación')
abline(a=0, b=1, lty=2)


###MINIMO
#MONTECARLO
nsim<-5000
nx<-100
minimo<-numeric(nsim)
for(i in 1:5000){
  datos_sim<-rmvnorm.mixt(nx, mus=mus, Sigmas=sigmas, props=as.vector(lambdas))
  indicador_sim<-datos_sim[,1]/datos_sim[,2]^2
  minimo[i]<-min(indicador_sim)
}
alfa <- 0.05
ic<-quantile(minimo,c(alfa/2, 1 - alfa/2))
ic
#precision
varmin <- (1/(nsim^2)) * sum((minimo - mean(minimo))^2)
sqrt(varmin)


#BOOTSTRAP
#INTERVALOS - METODO PERCENTIL T
alfa<-0.05
B <- 1000
n<-length(ind1)
remuestra<-numeric(n)
estadistico_boot <- numeric(B)
for (k in 1:B) {
  remuestra <- sample(ind1, n, replace = TRUE)
  estadistico_boot[k] <-min(remuestra)
}
IC_boot<- quantile(estadistico_boot, c(alfa/2, 1 - alfa/2))
IC_boot
#precision
varminboot <- (1/B) * sum((estadistico_boot - mean(estadistico_boot))^2)
sqrt(varminboot)



####MAXIMO
#MONTECARLO
nsim<-5000
```

```r
nx<-100
maximo<-numeric(nsim)
for(i in 1:5000){
  datos_sim<-rmvnorm.mixt(nx, mus=mus, Sigmas=sigmas, props=as.vector(lambdas))
  indicador_sim<-datos_sim[,1]/datos_sim[,2]^2
  maximo[i]<-max(indicador_sim)
}
alfa <- 0.05
ic<-quantile(maximo,c(alfa/2, 1 - alfa/2))
ic
#precision
varmax <- (1/(nsim^2)) * sum((maximo - mean(maximo))^2)
sqrt(varmax)


#BOOTSTRAP
#INTERVALOS - METODO PERCENTIL T
alfa<-0.05
B <- 1000
n<-length(ind1)
remuestra<-numeric(n)
estadistico_boot <- numeric(B)
for (k in 1:B) {
  remuestra <- sample(ind1, n, replace = TRUE)
  estadistico_boot[k] <-max(remuestra)
}
IC_boot<- quantile(estadistico_boot, c(alfa/2, 1 - alfa/2))
IC_boot
#precision
varmaxboot <- (1/B) * sum((estadistico_boot - mean(estadistico_boot))^2)
sqrt(varmaxboot)
```