# Título

**Carlos Lezama**[a,1,2], **Marco Medina**[a,1,2], **Emiliano Ramírez**[a,1,2], **and Santiago Villarreal**[a,1,2]

[a]Instituto Tecnológico Autónomo de México

Este manuscrito fue compilado el 8 de diciembre de 2021

**Please provide an abstract of no more than 250 words in a single paragraph. Abstracts should explain to the general reader the major contributions of the article. References in the abstract must be cited in full within the abstract itself and cited in the text.**

análisis bayesiano | aproximación estocástica | estimación | muestreo de importancia

## Introducción

De acuerdo a la Organización Mundial de la Salud, una de cada seis muertes en el mundo se debe al cáncer, convirtiéndole en la segunda causa de muerte a nivel global. En 2018, el cáncer fue responsable de aproximadamente 9.6 millones de muertes a nivel global. La mayoría de las muertes por cáncer están asociadas con malos hábitos: mala alimentación, falta de actividad física, así como el uso y abuso de sustancias como el tabaco y el alcohol. En la mayoría de los países, el cáncer se ubica como uno de los principales problemas de salud pública, sobre todo en aquellos países de ingreso medio o bajo.

En México, de acuerdo a la Agencia Internacional para la Investigación en Cáncer de la OMS, estima que el cáncer de colon y recto es el tercero más frecuente en México, con 14,900 casos nuevos por año. Las cifras de esta enfermedad parecen seguir en aumento, y una de las principales razones está relacionada con la falta de conocimiento de este cáncer por parte del sistema de salud, particularmente en conocer la distribución de su reaparición meses después de concluir tratamientos oncológicos.

Estudiar la distribución de los meses que transcurren antes de una reaparición de cáncer es de especial importancia ya que ayudará a entender mejor la efectividad de los tratamientos contra el cáncer e incluso brindaría información acerca de la persistencia del tipo de cáncer ya se por condiciones genéticas o patológicas del individuo.

En este proyecto, exploramos un método para estimar la distribución del número de meses libres de cáncer colorrectal antes de su reaparición, mediante un método de momentos utilizando simulaciones Monte Carlo. El problema consiste en estimar desde un enfoque de Optimización los parámetros de la distribución a priori que se supone siguen los datos. En este caso, dada la característica de supervivencia que tienen los datos de meses antes de la reparación de cáncer (donde podemos interpretar al evento "cáncer colorrectal reaparece" como el evento exitoso) suponemos que la distribución sigue una distribución $Gamma(\alpha, \beta)$.

Ahora bien, la estimación por método de momentos con estimaciones Monte Carlo y Newton-Raphson tiene una utilidad especial ya que es efectivo cuando los momentos de la distribución y sus derivadas no tienen forma analítica cerrada ya sea por construcción de la función o porque simplemente no se conoce. Es decir, dados los datos de la realización de alguna variable aleatoria puedes imponer condiciones a la distribución, en términos de momentos y parámetros, para adjudicarle alguna propiedad de interés que la literatura diga sobre tu conjunto de datos y estimar dichos momentos con este método. El alcance pragmático del método es amplio para este tipo de problemas pues es una herramienta que se puede usar cuando los parámetros están sobre-identificados; como alternativa al método de estimación por Máxima Verosimilitud, e , inclusive, se pueden añadir técnicas de reducción de varianza para la estimación Monte Carlo. En definitiva, es un método que, si bien no es el más eficiente, brinda flexibilidad y otro enfoque en la resolución del problema de estimación de parámetros.

## Datos

Los datos con los que trabajaremos son una muestra de 62 pacientes que tuvieron cáncer colorrectal y recibieron tratamiento y extracción de tumor, no obstante, reincidieron en él. La variable aleatoria en este caso es el número de meses que estuvieron libres del cáncer después de haber recibido el tratamiento y fueron dados de alta de la enfermedad. A continuación se presenta una pequeña tabla con estadísticos descriptivos de los datos que usaremos.

| Tabla de estadísticos descriptivos | |
|---|---|
| media | 41.77 |
| desv. estandar | 26.68 |
| mediana | 38 |
| percentil 25 | 19 |
| percentil 75 | 58 |

La naturaleza de la variable aleatoria nos dice que es sensato suponer que se distribuye como una $Gamma(\alpha, \beta)$. Observando histograma de los datos podemos ver que la suposición corresponde con la visualización gráfica de los datos. En la sección de resultados se comentará si fue acertada la suposición.

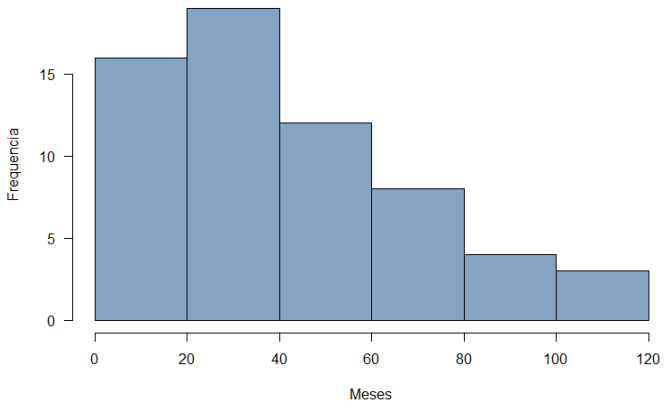
## Métodos

**A. Método de Momentos (1).** El método de momentos para estimar parámetros es considerado uno de los métodos más viejos y "confiables." Es ideal usarlo cuando los parámetros distribucionales que se quieren estimar están involucrados en la función de los momentos teóricos (comúnmente en media y varianza poblacional). El argumento en el que se basa su implementación es muy intuitivo: usar el principio de analogía muestral, estimando con las contrapartes muestrales los momentos teóricos, para obtener las estimaciones de los parámetros distribucionales deseados.

[1] Todos los autores contribuyeron a este trabajo por igual.

[2] Trabajo presentado para el curso de **Simulación (EST-24107)** impartido por Jorge Francisco de la Vega Góngora. E-mail: jorge.delavegagongora@gmail.com

**Fig. 1.** Histograma de meses antes de recaída al cáncer




Sea $X_1, \ldots, X_n$ una muestra aleatoria de una población cuya función de densidad o masa es $f(x \mid \theta_1, \ldots, \theta_k)$, donde $\theta_i$ es el i-ésimo parámetro de la distribución que se desea estimar. El método de momentos consiste en igualar los $k$ momentos teóricos con los $k$ momentos muestrales para resolver el sistema de ecuaciones simultáneas que se genera.

Formalmente definimos:

$$
\begin{aligned}
m_1 &= \frac{1}{n}\sum_{i=1}^{n} X_i^1, \quad \mu_1 = \mathbb{E}[X^1], \\
m_2 &= \frac{1}{n}\sum_{i=1}^{n} X_i^2, \quad \mu_2 = \mathbb{E}[X^2], \\
&\vdots \\
m_k &= \frac{1}{n}\sum_{i=1}^{n} X_i^k, \quad \mu_k = \mathbb{E}[X^k].
\end{aligned}
$$

Los momentos teóricos típicamente son una función del vector de parámetros $(\theta_1, \ldots, \theta_k)$, es decir, $\mu_j(\theta_1, \ldots, \theta_k)$. Así pues, el método de momentos obtiene los estimadores $(\hat{\theta_1}, \ldots, \hat{\theta_k})$ resolviendo el siguiente sistema de ecuaciones de $(\theta_1, \ldots, \theta_k)$ en términos de de $(m_1, \ldots, m_k)$:

$$
\begin{aligned}
m_1 &= \mu_1(\theta_1, \ldots, \theta_k) \\
m_2 &= \mu_2(\theta_1, \ldots, \theta_k) \\
&\vdots \\
m_k &= \mu_k(\theta_1, \ldots, \theta_k)
\end{aligned}
$$

Existen otras formas más flexibles de implementar el método de momentos, como el método de momentos generalizado. Por ejemplo, los coeficientes de un modelo de regresión lineal pueden obtenerse por este método igualando los momentos teóricos de las ecuaciones normales del problema de minimización (condiciones de primer orden) con el vector de ceros. Usando el principio de analogía, igualamos las contrapartes muestrales de los momentos teóricos con un vector de ceros y resolvemos el sistema de ecuaciones homogéneo. Asimismo, existen distintas versiones del método que se pueden utilizar para modelos identificados (mismo número de ecuaciones que parámetros a estimar) y sobre-identificados (más ecuaciones que parámetros a estimar).

**B. Método de Newton (2).** El método de Newton es uno de los métodos numéricos más básicos que se pueden utilizar para encontrar las raíces de una función. Supongamos que nuestra función $f$ es tal que $f \in C^2[a, b]$. Sea $x_k$ una iteración del método. Recordemos que el teorema de Taylor nos dice que para $f \in C^{k+1}[a, b]$ con $x, x_0 \in [a, b]$ y $h \in \mathbb{R}$ tenemos que

$$
\begin{aligned}
f(x_0 + h) = f(x_0) + hf'(x_0) + \frac{h^2}{2}f''(x_0) + \cdots + \frac{h^k}{k!}f^{(k)}(x_0) \\
+ \frac{h^{k+1}}{(k+1)!}f^{(k+1)}(\xi)
\end{aligned}
$$

donde $\xi \in (x_0, x_0 + h)$.

Así pues, podemos escribir a $f(x) = f(x + x_k - x_k)$ como sigue

$$
f(x) = f(x_k) + f'(x_k)(x - x_k) + \frac{f''(\xi(x))(x - x_k)^2}{2}
$$

donde $\xi(x)$ es un punto desconocido en el intervalo $(x, x_k)$.

Sea $x = x^*$ tal que $f(x^*) = 0$. Si $f$ fuera una función lineal entonces el problema sería realmente sencillo ya que $f'' \equiv 0$, por lo que podríamos encontrar la raíz de la función resolviendo $0 = f(x_k) + f'(x_k)(x^* - x_k)$, dándonos como resultado $x^* = x_k - f(x_k)/f'(x_k)$.

Si la función $f$ no es lineal entonces definimos la siguiente formula iterativa para $x_k$:

$$
x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}, \quad k = 0, 1, 2, \ldots
$$

Al definir de esta forma la regla de actualización de la iteración $x_k$, ignoramos el factor $f''(\xi(x^*))(x^* - x_k)^2/2$ de la expansión de Taylor que realizamos, ya que si $x_k$ está cerca de $x^*$ entonces la diferencia $(x^* - x_k)$ es muy pequeña, por lo que podríamos suponer razonablemente que la siguiente iteración $x_{k+1}$ está cerca de $x^*$ aun si no es considere dicho término.

**C. Estimación Monte Carlo (3).** Dado un evento $A$, la estimación Monte Carlo de la probabilidad del evento A $\mathbb{P}(A)$ se obtiene repitiendo el experimento aleatorio un número definido de veces y tomar la proporción de intentos exitosos en los que el evento $A$ sucede como una aproximación de $\mathbb{A}$.

Esta forma de estimación de las probabilidades de eventos aleatorios es intuitiva y corresponde a la concepción general de cómo deberían comportarse las probabilidades. La estimación Monte Carlo nos dice que la probabilidad de un evento es la proporción de largo plazo de que ese evento suceda repetidas veces en pruebas aleatorizadas.

Este método está justificado formalmente por la Ley Fuerte de los Grandes Números. Sea $1_k$ la indicadora de ocurrencia del evento $A$ en el $k$-ésimo intento, luego

$$
\frac{1}{n}\sum_{k=1}^{n} 1_k
$$

es la proporción de que en $n$ intentos suceda el evento $A$. Suponiendo que las $1_k$ son idénticamente distribuidas tenemos que $\mathbb{E}(1_k) = \mathbb{P}(A),\ \forall k$.

Luego, por la Ley Fuerte de los Grandes Números

$$\lim_{n\to\infty} \frac{1}{n}\sum_{k=1}^{n} = \mathbb{P}(A), \text{ con probabilidad 1.}$$

Es decir, para $n$ grande el estimador Monte Carlo de la probabilidad del evento $A$ es

$$\frac{1}{n}\sum_{k=1}^{n} \approx \mathbb{P}(A).$$

**D. Algoritmo para Método de Momentos con Simulación Monte Carlo.** Suponemos que la distribución a estimar sigue una distribución gamma con parámetros $\theta = (\alpha, \beta)$, dónde $\alpha > 0$ es el parámetro de forma y $\beta > 0$ es el parámetro de escala, tal que su función de densidad está dada por:

$$f(x;\theta) = \frac{1}{\Gamma(\alpha)\beta^{\alpha}} x^{\alpha-1} e^{-\frac{x}{\beta}}, \qquad x \in (0,\infty)$$

Sea $\mu(\theta)$ el vector de momentos teóricos de interés de la distribución gamma con parámetros $\theta = (\alpha, \beta)$. En particular, consideramos la media y la varianza de la distribución gamma:

$$\mu(\theta) = \begin{bmatrix} \mu \\ \sigma^2 \end{bmatrix}$$

Sea $\mu_0$ el vector de momentos muestrales de los datos observados en nuestra base que buscamos reproducir. En nuestro caso, buscamos reproducir la media y la varianza muestral, tal que:

$$\mu_0 = \begin{bmatrix} \bar{x} \\ Var(x) \end{bmatrix}$$

Comenzamos nuestra estimación proponiendo un valor inicial para los parámetros que caracterizan a la distribución: $\theta_0 = (\alpha_0, \beta_0)$. Después realizamos el siguiente proceso iterativo por $t$ iteraciones o hasta que alcancemos un nivel de tolerancia deseado en los valores estimados de los parámetros.

Para cada iteración $t = 1, 2, \ldots$:

1. Muestreamos $N_t$ observaciones de la distribución gamma con parámetros $\theta_t$.
2. Estimamos $\mu(\theta)$ y $\mu'(\theta)$ mediante el uso de estimadores Monte Carlo.
3. Utilizamos las estimaciones de $\hat{\mu}(\theta)$ y $\hat{\mu}'(\theta)$ para obtener $\theta_{t+1}$ mediante el método Newton-Raphson.
4. Verificamos que $\theta_{t+1}$ sea parte del espacio de parámetros de la distribución gamma ($\theta > 0$), de lo contrario, mantenemos el valor de $\theta_t$.

Los estimadores Monte Carlo que utilizamos para estimar $\mu(\theta)$ y $\mu'(\theta)$ provienen de los propuestos por Gelman (4). Dado que $\mu(\theta) = \mathbb{E}[h(x)|\theta]$, donde $h(x)$ es función dada, el estimador Monte Carlo para $\mu(\theta)$ es:

$$\hat{\mu}(\theta) = \frac{1}{N}\sum_{i=1}^{N} h(x_i)$$

Por otro lado,

$$\mu'(\theta) = \frac{d}{d\theta}\mathbb{E}[h(x)|\theta] = \int h(x)\frac{d}{d\theta} f(x;\theta)dx = \mathbb{E}[h(x)U(x,\theta)^T]$$

Donde $U(x,\theta)^T = \frac{d}{d\theta}\log f(x;\theta)$. Sea entonces el estimador Monte Carlo para $\mu'(\theta)$:

$$\hat{\mu}'(\theta) = \frac{1}{N}\sum_{i=1}^{N} h(x_i)U(x_i,\theta)^T$$

Para la implementación del algoritmo en nuestro caso particular, tomamos en cuenta lo siguiente. El vector de parámetros a estimar bidimensional, dado por: $\theta = (\alpha, \beta)$; la función $h(\vec{x}) = (x_i, (x_i - \overline{\mathbf{x}})^2)$ y, finalmente, nuestro sistema de ecuaciones $U(x,\theta) = \frac{d}{d\theta} log(f(\vec{x}))$, dado por:

$$\begin{cases} \frac{d}{d\alpha} = -\frac{\Gamma'(\alpha)}{\Gamma(\alpha)} - ln(\beta) + ln(x_i) \\ \frac{d}{d\beta} = -\frac{\alpha}{\beta} + \frac{x_i}{\beta^2} \end{cases} \qquad i = 1, 2, ..., N$$

Después de definir las ecuaciones y funciones necesarias para el algoritmo, usamos los resultados de las simulaciones de la distribución objetivo para calcular $\hat{\mu}'(\theta)$, $\hat{\mu}(\theta)$ y después, realizar la siguiente iteración del algoritmo de Newton-Raphson en 2 dimensiones, dada por:

$$\theta_{t+1} = \theta_t + [\hat{\mu}'(\theta_t)]^{-1} * (\mu_0 - \hat{\mu}(\theta_t)) \qquad t = 1, 2, ...$$

donde $\theta_t$ es un vector de 2x1 que contiene el valor de los parámetros de la iteración actual, $\hat{\mu}'(\theta_t)$ es una matriz de 2x2, $\mu_0$ es un vector que contiene la media y varianza muestral y $\hat{\mu}(\theta)$ es un vector de 2x1.

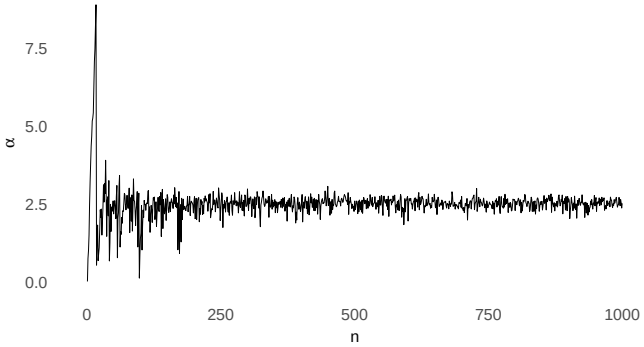
Al implementar el algoritmo propuesto por Gelman, nos encontramos con 2 dificultades principales; la primera, fue que había iteraciones donde el parámetro $\alpha$ era menor a 0 y por lo tanto, se encontraba fuera del espacio paramétrico y el algoritmo no podía concluir. Por lo tanto, implementamos una condición que solo tomara los valores positivos de los parámetros y en caso de encontrar uno negativo, quedarse en la estimación del paso anterior y repetir el algoritmo. De aquí, surgió la segunda dificultad: al encontrarse con un valor negativo, las estimaciones aterrizaban en el valor estimado anterior y no cambiaban durante el resto del algoritmo. Para resolver este problema, cambiamos la condición a que el algoritmo tomara el valor de la estimación anterior más un error estocástico distribuido $Unif(0,1)$ y así, logramos que el algoritmo evitara valores fuera del espacio paramétrico y que no se estancara en la estimación anterior en caso de hacerlo.

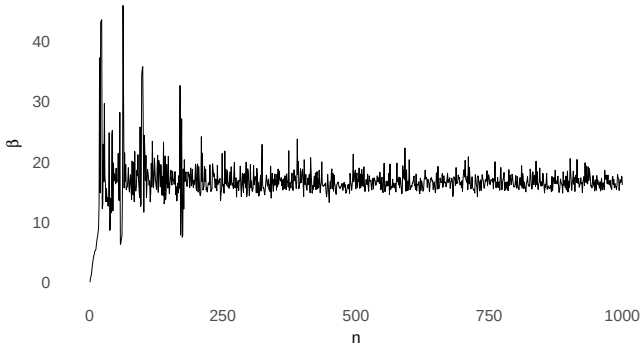
## Resultados

```
gamma.moments <- function(
        data, iters, alpha.0, beta.0
) {
  sample.mean <- mean(data)
  sample.var <- var(data)
  mu.0 <- c(sample.mean, sample.var)
  theta <- matrix(NA, 2, iters)
  theta[, 1] <- c(alpha.0, beta.0)

  for (i in 2:iters) {
```

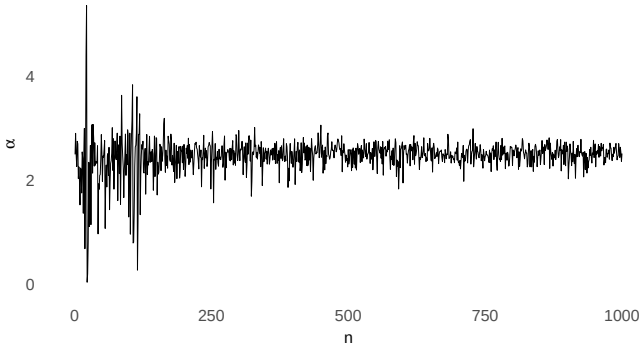
```
  n <- i + 100
  simulated <- rgamma(
          n,
          shape = theta[1, i - 1],
          scale = theta[2, i - 1]
  )
  mu <- c(mean(simulated), var(simulated))
  mu.hat <- matrix(0, 2, 2)
  h <- u <- NULL

  for (j in 1:length(simulated)) {
    u[1] <- -digamma(theta[1, i - 1]) -
            log(theta[2, i - 1]) +
            log(simulated[j])
    u[2] <- (-theta[1, i - 1] / theta[2, i - 1]) +
            simulated[j] / (theta[2, i - 1]^2)
    h[1] <- simulated[j]
    h[2] <- (simulated[j] - mean(simulated))^2
    m <- h %*% t(u)
    mu.hat <- mu.hat + m
  }

  mu.hat <- mu.hat / length(simulated)

  par <- theta[, i - 1] +
          solve(mu.hat) %*% (mu.0 - mu)

  if (par[1] * par[2] > 0) {
    theta[, i] <- par
  } else {
    theta[, i] <- theta[, i - 1] +
            runif(1)
  }
}

theta <- data.frame(
        x = theta[1,],
        y = theta[2,],
        n = 1:iters
)

p.1 <- ggplot(theta) +
        geom_line(aes(x = n, y = x),
                  size = 0.1) +
        labs(title = NULL,
             x = "n",
             y = expression(alpha))

p.2 <- ggplot(theta) +
        geom_line(aes(x = n, y = y),
                  size = 0.1) +
        labs(title = NULL,
             x = "n",
             y = expression(beta))

shape <- mean(theta$x)
scale <- mean(theta$y)

dist.mean <- mean(shape * scale)
dist.var <- mean(shape * (scale^2))

  test <- ks.test(
          data, "pgamma",
          shape = shape, scale = scale
  )

  results <- list(
          test,
          p.1, p.2,
          shape, scale,
          dist.mean, dist.var,
          sample.mean, sample.var
  )

  return(results)
}
```

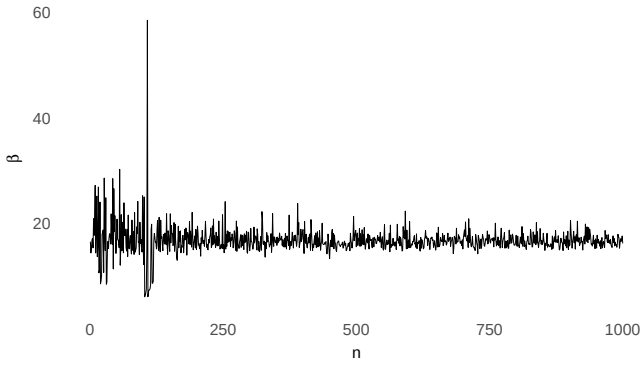
```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  data
## D = 0.092, p-value = 0.7
## alternative hypothesis: two-sided
```



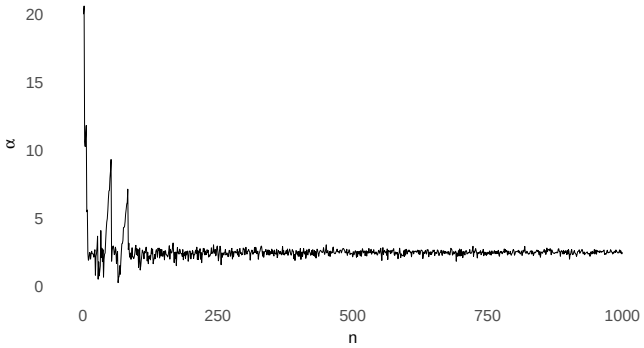



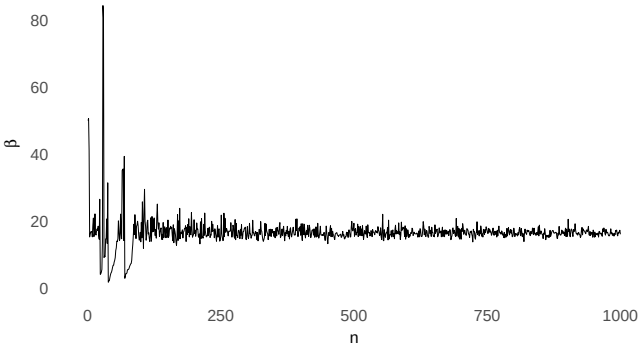

```
## [1] 2.499

## [1] 16.69

## [1] 41.69

## [1] 695.8

## [1] 41.77

## [1] 691.1

##
##  One-sample Kolmogorov-Smirnov test
##
## data:  data
## D = 0.087, p-value = 0.7
## alternative hypothesis: two-sided
```




```
## [1] 2.476

## [1] 16.7

## [1] 41.35

## [1] 690.4

## [1] 41.77

## [1] 691.1

##
##  One-sample Kolmogorov-Smirnov test
##
## data:  data
## D = 0.12, p-value = 0.3
## alternative hypothesis: two-sided
```




```
## [1] 2.62

## [1] 16.68

## [1] 43.71

## [1] 729.1

## [1] 41.77

## [1] 691.1
```

## Conclusiones

De la aplicación del algoritmo, derivamos múltiples resultados útiles para el problema que se nos propuso y también recopilamos un conjunto de limitaciones y alcances que el algoritmo y el trasfondo teórico puede brindar. A continuación mencionaremos qué podemos inferir de dichos resultados, el por qué de las limitaciones que enfrentamos y el desarrollo matemático necesario para ampliar los alcances del estudio.

Los resultados, como se presentaron en la sección anterior, son válidos y nuestro estudio nos permitirá hacer inferencia sobre los meses que tarda en regresar el cáncer a los pacientes de cáncer de colon. La prueba de bondad y ajuste (Kolmogorov-Smirnoff) que realizamos sobre la muestra y los parámetros teóricos que se estimaron con el algoritmo indican que existe evidencia estadística suficiente para asumir que la muestra proviene de una distribución Gamma con los parámetros estimados. A partir de este resultado, podemos hacer inferencia sobre la variable aleatoria estudiada y determinar, por ejemplo, el tiempo esperado en el cual los pacientes volverán a presentar síntomas y la probabilidad de que regrese el tumor en cierto tiempo.

Las dificultades principales que enfrentamos al implementar el algoritmo fueron, como se mencionó en el apartado anterior, estimaciones fuera del espacio paramétrico y el estancamiento del algoritmo en un estado. Como se comentó, se agregaron condiciones sobre los parámetros para que fueran únicamente positivos y en caso de estimar un valor negativo, el algoritmo permanece en el estado anterior más una perturbación estocástica uniforme entre 0 y 1.

Un aspecto importante dentro de la implementación del algoritmo es la velocidad de convergencia del mismo. Al emplear métodos como Newton-Raphson multidimensional y simulación Monte Carlo para estimar los momentos, es evidente que el algoritmo tendrá cierta velocidad de convergencia. En el trabajo publicado por Gelman se menciona que, al momento de simular las observaciones de la distribución teórica, la velocidad del algoritmo depende de lo siguiente: conforme aumente el valor del paso $t$ del algoritmo de Newton-Raphson,

para garantizar convergencia también habrá que aumentar el tamaño de la muestra simulada. Y, en efecto, nuestro primer intento con un tamaño de muestra fijo no se mantenía en un valor exacto y, al actualizar el modelo y aumentar el tamaño de muestra en cada iteración, el algoritmo sí convergió a un valor. También, como se menciona en el artículo, este detalle hace que el algoritmo sea ineficiente, ya que hay que simular una nueva muestra de tamaño creciente en cada iteración.

Para hacer más eficiente el algoritmo, el autor propone hacer muestro por importancia y de esta forma hacer las iteraciones del algoritmo de Newton-Raphson con un conjunto fijo de valores simulados. De esta forma, se pueden hacer $n$ pasos del algoritmo de Newton-Raphson son solo una simulación y después de cierto tiempo, volver a simular una muestra y así acelerar el proceso.

Los alcances que tiene el proyecto son vastos y también mencionados en el trabajo de Gelman. En primer lugar y, evidentemente, el algoritmo es una herramienta con un gran trasfondo matemático y estadístico, por lo cual se puede trabajar con distribuciones de las cuales no conocemos la constante normalizadora. En nuestro caso, se empleó una distribución conocida y simple, para evidenciar que el algoritmo funciona a escala pequeña. Sin embargo, se puede extender, como se mencionó, a distribuciones cuya constante normalizadora sea una expresión difícil de calcular. También, como se menciona en el párrafo anterior, se puede emplear muestreo por importancia para hacer el algoritmo más eficiente. Otra alternativa para la cual se puede ajustar el algoritmo es, como se revisó en el curso, en caso de conocer únicamente la distribución límite, se puede simular la muestra usando el algoritmo de Metrópolis-Hastings. Finalmente, igual es posible implementar un ajuste de mínimos cuadrados en el caso de que el problema tenga más momentos definidos que parámetros (sobredeterminación), y, de esta forma, implementar el algoritmo de Newton-Raphson con un ajuste de regresión sobre las ecuaciones normales del problema.

Como pudimos observar, el algoritmo produce resultados y conclusiones válidas y aplicables en problemas de la vida real e, igualmente, es un método flexible que puede ajustarse a las múltiples condiciones que puede presentar cualquier problema. También, descansa sobre múltiples temas que se revisaron a lo largo del curso y esto permitió mejorar nuestro entendimiento de qué había que hacer al momento de la implementación.

## Anexos

## Referencias


1. Casella G, Berger RL (2021) *Statistical inference* (Cengage Learning).
2. Ascher UM, Greif C (2011) *A first course on numerical methods* (SIAM).
3. Dobrow RP (2016) *Introduction to stochastic processes with r* (John Wiley & Sons).
4. Gelman A (1995) Method of moments using monte carlo simulation. *Journal of Computational and Graphical Statistics* 4(1):36–54.