

# Proyecto 2 Estadística

Alejandro Borda

Mayo 2024

## 1 Introducción

En este documento nos proponemos estudiar los estimadores de máxima verosimilitud y los estimadores por método de momentos para la distribución  $\text{Gamma}(\alpha, \beta)$ , la cual sabemos que tiene función de densidad (para  $x > 0$ )

$$f_{\theta}(x) = \frac{x^{\alpha-1} e^{-x/\beta}}{\Gamma(\alpha) \beta^{\alpha}} \quad (1)$$

con  $\mu = \alpha\beta$  y  $\sigma^2 = \alpha\beta^2$ . Con el propósito de estudiar los estimadores mencionados de forma computacional definimos los valores reales a estimar en el resto del proyecto como  $\alpha_0 = 2.5$ ,  $\beta = 4.6$ .

Finalmente, antes de empezar con el desarrollo computacional, es conveniente encontrar la matriz de información de Fisher de una  $\text{Gamma}(\alpha, \beta)$ . Sabemos que bajo condiciones de regularidad, la posición  $i, j$  de la matriz de información de Fisher se puede escribir como  $I(\theta)_{i,j} = -\frac{\partial^2}{\partial \theta_i \partial \theta_j} \ln(f_{\theta}(x))$ , entonces como en nuestro caso tenemos

$$\ln(f_{\theta}(x)) = (\alpha - 1)\ln(x) - \frac{x}{\beta} - \ln(\Gamma(\alpha)) - \alpha \ln(\beta) \quad (2)$$

Entonces si definimos las funciones digamma y trigamma respectivamente como

$$\Psi(\alpha) = \frac{\partial \ln(\Gamma(\alpha))}{\partial \alpha} \text{ y } \Psi'(\alpha) = \frac{\partial^2 \ln(\Gamma(\alpha))}{\partial \alpha^2} \quad (3)$$

es claro que tenemos

$$-\frac{\partial^2 \ln(f_{\theta}(x))}{\partial \alpha^2} = \Psi'(\alpha) \quad (4)$$

$$-\frac{\partial^2 \ln(f_{\theta}(x))}{\partial \alpha \partial \beta} = \frac{\partial^2 \ln(f_{\theta}(x))}{\partial \beta \partial \alpha} = \frac{1}{\beta} \quad (5)$$

$$-\frac{\partial^2 \ln(f_{\theta}(x))}{\partial \beta^2} = \frac{\alpha}{\beta^2} \quad (6)$$

Entonces finalmente obtenemos

$$I_f(\theta) = \begin{pmatrix} \Psi'(\alpha) & \frac{1}{\beta} \\ \frac{1}{\beta} & \frac{\alpha}{\beta^2} \end{pmatrix} \quad (7)$$

## 2 Estimación por Newton-Raphson

Primero nos proponemos encontrar los estimadores de máxima verosimilitud de nuestro modelo  $\text{Gamma}(\alpha, \beta)$ , el cual tiene como ecuaciones de verosimilitud

$$\frac{\partial l(\theta)}{\partial \alpha} = f_1 = 0 \text{ y } \frac{\partial l(\theta)}{\partial \beta} = f_2 = 0 \quad (8)$$

```

NR = function(alpha_k, beta_k, data) {
  if (!all(is.numeric(data))) {
    stop("Data vector contains non-numeric values.")
  }

  tol = 1e-6 # Threshold de convergencia (recomendado)
  max_iter = 500
  iter = 0
  converged = FALSE

  while (!converged && iter < max_iter) {
    n = length(data)
    Dfinv = Df_inverse(alpha_k, beta_k, data)
    coord_n = c(alpha_k, beta_k) - Dfinv %% f(alpha_k, beta_k, data)

    # Revisamos convergencia
    if (!any(is.na(coord_n))) {
      if (max(abs(coord_n - c(alpha_k, beta_k))) < tol) {
        converged = TRUE
      } else {
        alpha_k = coord_n[1]
        beta_k = coord_n[2]
        iter = iter + 1
      }
    } else {
      stop("Convergence failed: NaN values encountered.")
    }
  }
}

```

Figure 1: Implementación en R del método Newton-Raphson bivariado para  $\text{Gamma}(\alpha, \beta)$

sin embargo, dado que estas ecuaciones no tienen solución analítica nos proponemos usar el método de Newton-Raphson bivariado.

Primero notamos que las ecuaciones de (7) tienen la forma exacta

$$\frac{\partial l(\theta)}{\partial \alpha} = \sum_{i=1}^n \ln(x_i) - \Psi(\alpha) - \ln(\beta) \quad (9)$$

$$\frac{\partial l(\theta)}{\partial \beta} = \sum_{i=1}^n \frac{x}{\beta^2} - \frac{\alpha}{\beta} \quad (10)$$

de tal forma que podemos obtener la matriz gradiente de estas dos ecuaciones como

$$Df(\alpha, \beta) = \begin{pmatrix} \frac{\partial f_1}{\partial \alpha} & \frac{\partial f_1}{\partial \beta} \\ \frac{\partial f_2}{\partial \alpha} & \frac{\partial f_2}{\partial \beta} \end{pmatrix} = \begin{pmatrix} -n\Psi'(\alpha) & -\frac{n}{\beta} \\ -\frac{n}{\beta} & -\frac{2n\mu}{\beta^3} + \frac{n\alpha}{\beta^2} \end{pmatrix} \quad (11)$$

Entonces, acorde con [1], para aplicar el método de Newton-Raphson bivariado el paso iterativo esta dado por

$$\begin{pmatrix} \alpha_k \\ \beta_k \end{pmatrix} = \begin{pmatrix} \alpha_{k-1} \\ \beta_{k-1} \end{pmatrix} - (Df(\alpha_{k-1}, \beta_{k-1}))^{-1} f(\alpha_{k-1}, \beta_{k-1}) \quad (12)$$

donde  $f(\alpha, \beta) = (f_1(\alpha, \beta), f_2(\alpha, \beta))$  y el punto inicial ( $k = 0$ ) lo vamos a tomar como los estimadores de momento  $\tilde{\theta} = (\tilde{\alpha}, \tilde{\beta})$  donde naturalmente  $\tilde{\alpha} = \frac{\bar{x}^2}{s}$  y  $\tilde{\beta} = \frac{s}{\bar{x}}$  para la media y varianza muestral donde se obtuvieron las ecuaciones por medio de la media y varianza conocida de la Gamma. Computacionalmente podemos implementar el método de Newton-Raphson bivariado como se puede ver en la figura 1, en donde la función "Df-inverse" calcula la matriz  $Df$  dado la longitud del conjunto de datos  $n$  y los valores de  $\alpha$  y  $\beta$  como se definió en la ecuación (11). Entonces en esta implementación se define un valor de convergencia el cual detiene la implementación si el cambio de una iteración a otra es menor a el valor definido y definimos un máximo de iteraciones para el caso en que la implementación no converja. Una vez con estas condiciones definidas, se repite el paso iterativo definido por la ecuación (12) y se revisa en cada paso si ya se cumplió convergencia para finalizar la implementación.

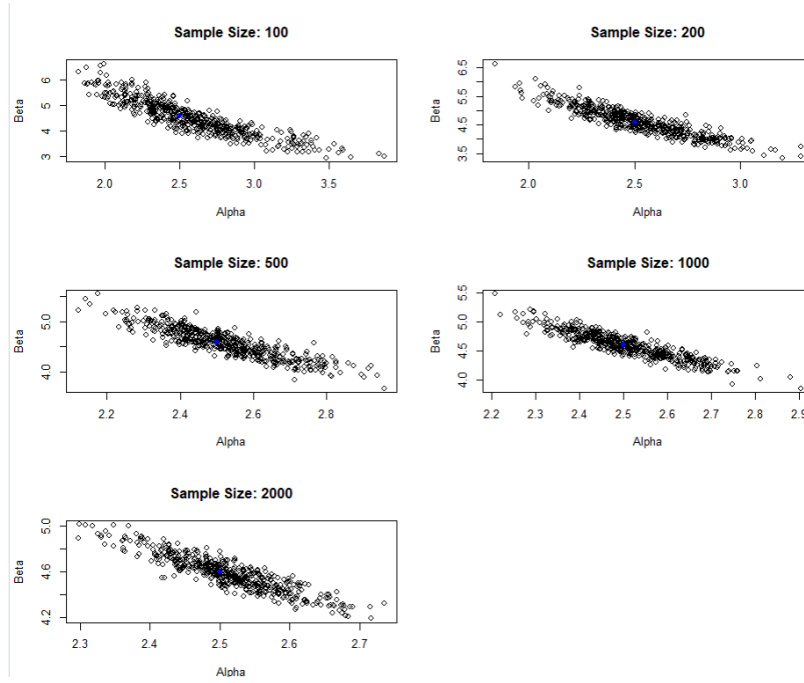


Figure 2: Gráficas de los estimadores de máxima verosimilitud obtenidos por NR, además se incluye en azul el valor real de  $(\alpha, \beta) = (2.5, 4.6)$

### 3 Gráficas y resultados de los estimadores

En esta sección nos proponemos estudiar los resultados obtenidos para los estimadores por método de momentos y los estimadores de máxima verosimilitud encontrados por NR en la sección anterior. Con este propósito definimos los 5 valores del tamaño muestral  $n \in \{100, 200, 500, 1000, 2000\}$  y generamos 500 muestras de dicho tamaño para cada caso, obtenemos por ambos métodos las gráficas de los parámetros estimados dado por las coordenadas  $(\alpha, \beta)$  para cada tamaño muestral y los boxplots correspondientes de ambos parámetros por separado para cada tamaño muestral. De esta forma obtenemos los siguientes resultados para los estimadores de máxima verosimilitud obtenidos por el método de NR que se pueden ver en las figuras 2 y 3.

En estas se puede notar en primera instancia que la distribución de los EMVs en el plano toma la forma de una elipse centrada en el valor real, la cual se concentra mas alrededor de este punto central a medida que aumenta el tamaño muestral. También podemos notar en los boxplots que efectivamente los valores obtenidos del EMV se concentra rápidamente alrededor del valor real de tal forma que el boxplot del tamaño  $n = 2000$  es una fracción del correspondiente a  $n = 100$  en cuanto al rango de valores que componen para el caso de ambos parámetros.

En cuanto a los estimadores de momento obtenemos las gráficas 4 y 5, en donde primero podemos notar que aunque también se encuentra una forma aproximadamente elíptica alrededor del valor verdadero, esta es mucho mas difusa y de área mucho mayor como se puede notar en la escala del eje  $y$ . Además, esta no es tan simétrica como esperaríamos encontrar en una elipse verdadera. En cuanto a los boxplots, también encontramos que la repartición de datos cubre un rango mucho mayor que los EMV tanto en sus valores extremos como en sus cuartiles, reforzando la idea de que los EMV son mucho mas acertados con respecto al valor real.

En cuanto a la implementación de NR, se encontró en el proceso de obtener los datos para esta

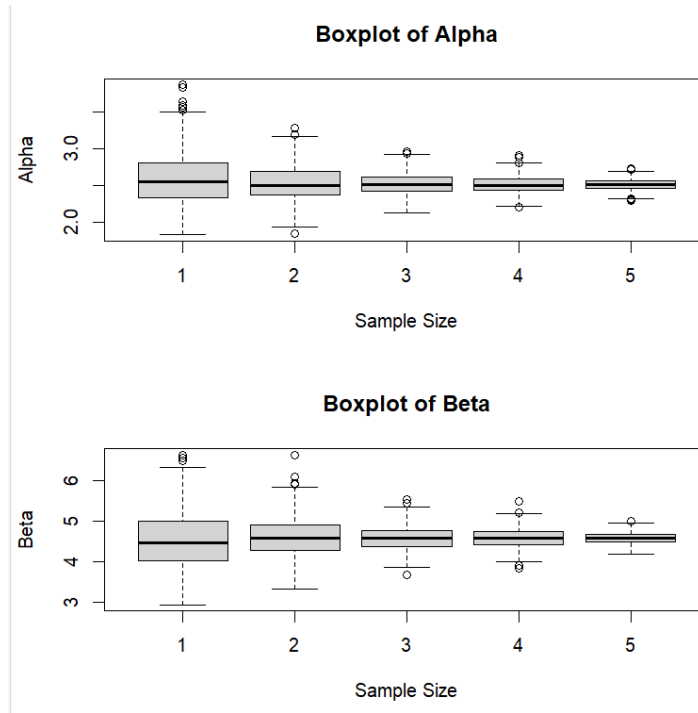


Figure 3: Gráficas boxplot de  $\alpha$  y  $\beta$  de el EMV para los diferentes tamaños muestrales. Los tamaños muestrales  $n \in \{100, 200, 500, 1000, 2000\}$  corresponden en el eje  $x$  a 1,2,3,4 y 5 respectivamente.

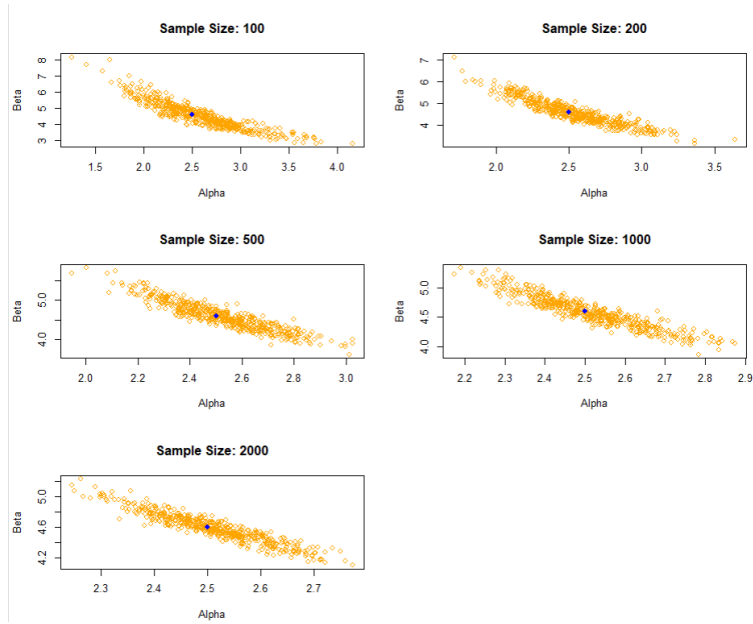


Figure 4: Gráficas de los estimadores por método de momentos, además se incluye en azul el valor real de  $(\alpha, \beta) = (2.5, 4.6)$ .

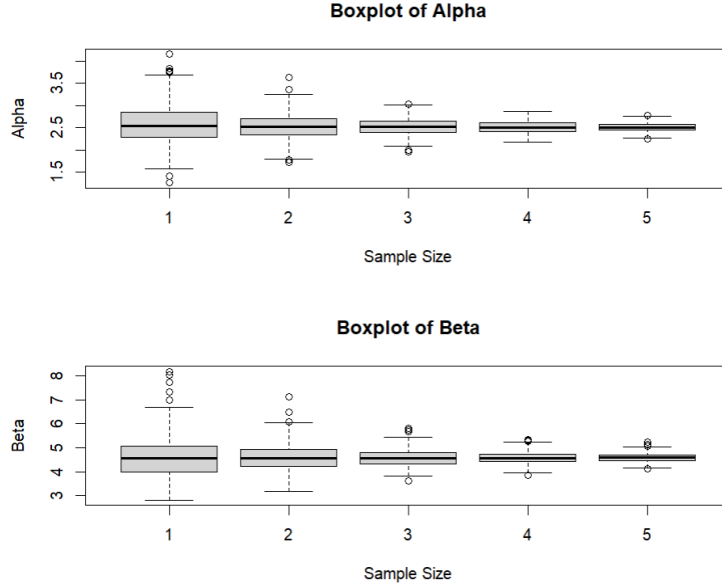


Figure 5: Gráficas boxplot de  $\alpha$  y  $\beta$  por método de momentos para los diferentes tamaños muestrales. Los tamaños muestrales  $n \in \{100, 200, 500, 1000, 2000\}$  corresponden en el eje  $x$  a 1,2,3,4 y 5 respectivamente.

sección que la implementación bivariada de Newton-Raphson descrita converge increíblemente rápido con el parámetro de convergencia mencionado anteriormente, pues prácticamente nunca se necesitan mas de 4 o 5 iteraciones para que converja la implementación y en muchos casos incluso hasta 2 iteraciones son suficientes. Esta rapidez de convergencia lleva a que la complejidad computacional sea muy baja y el programa para calcular NR en los 500 datos para los 5 diferentes tamaños muestrales no tarde mas de 10 segundos.

## 4 Conjuntos de confianza

Si  $\hat{\theta}$  es el EMV y  $\theta_0$  es el valor verdadero, por el TLC tenemos que

$$\sqrt{n}I_F(\hat{\theta})^{1/2}(\hat{\theta} - \theta_0) \xrightarrow{(d)} N(0, I) \quad (13)$$

de donde podemos deducir de manera similar que (donde ' representa la transpuesta)

$$n(\hat{\theta} - \theta_0)'I_f(\hat{\theta})(\hat{\theta} - \theta_0) \xrightarrow{(d)} \chi_1^2 \quad (14)$$

Entonces si definimos  $\rho$  como el cuantil superior correspondiente al 0.95 de la  $\chi_1^2$ , es claro que el siguiente conjunto  $C$  contiene a  $\theta_0$  con probabilidad 0.95, donde

$$C = \{\theta : n(\hat{\theta} - \theta_0)'I_f(\hat{\theta})(\hat{\theta} - \theta_0) \leq \rho\} \quad (15)$$

Ahora, podemos implementar este mismo procedimiento como se puede ver en la figura 6 en donde dado los datos correspondientes a un tamaño muestral se formatean los EMV obtenidos anteriormente en un vector y se calcula la condición de la ecuación (15) para calcular la cuenta de datos que lo cumplen y calcular el porcentaje al dividir sobre el numero de datos totales. En este se encuentra que aunque la implementación para ser correcta a primera vista, tiene que incluir alguna falla pues para tamaños muestrales de hasta  $n = 200000$  se encuentran probabilidades encontradas no mayores a 0.88 y no sería

```

C = function(theta_0, data, n) {
  rho = qchisq(0.95, 1)
  counter = 0

  for (i in 1:length(data)) {

    d = c(data[[i]][1], data[[i]][2]) #Formateamos los datos para calcular el rango
    thetaVec = c(theta_0[[1]], theta_0[[2]])
    F_info = If(data[[i]][1], data[[i]][2])
    mag = n*t(d - thetaVec) %% F_info %% (d - thetaVec) #Valor para revisar <= rho

    if (mag <= rho) { #Condicion de estar en C
      counter = counter + 1
    }
  }

  return (counter / length(data))
}

```

Figure 6: Programa de calculo de porcentaje de datos en el conjunto  $C$ .

razonable concluir que la falla esta en el proceso de obtención de los EMV pues tanto en las secciones anteriores como en las secciones próximas utilizando los mismos datos de EMV estos siguen la teoria apropiadamente en todos los casos.

#### 4.1 Área del elipse

Como discutimos anteriormente, los datos obtenidos de los EMV parecían tomar la forma de una elipse, en particular se puede mostrar que para una matriz diagonal  $D$  tal que  $C'I_f(\theta)C = D$ , la condición (15) del conjunto  $C$ , es equivalente a

$$n(\hat{\theta} - \theta_0)'UDU'(\hat{\theta} - \theta_0) \leq \rho \quad (16)$$

donde ahora podemos notar que  $C$  efectivamente toma la forma de una elipse en la variable  $Y = U'(\hat{\theta} - \theta_0)$ , de esta forma podríamos calcular el área de la elipse correspondiente a los datos de los EMV que caen en el conjunto  $C$  como

$$A = \pi \sqrt{\lambda_{\min}} \sqrt{\lambda_{\max}} \quad (17)$$

donde  $\lambda_{\max}$  y  $\lambda_{\min}$  son los valores propios máximos y mínimos respectivamente de la matriz  $Y$ . Este proceso se implemento en R como se puede ver en la figura 7 donde para cada  $\hat{\theta}$  calculamos su área con el mismo proceso descrito de manera teórica para cada dato que cumpla la condición.

Finalmente con este procedimiento podemos realizar un boxplot de las áreas calculadas para los diferentes tamaños muestrales, en donde encontramos los resultados que se pueden encontrar en la figura 8, donde podemos notar que a medida que aumenta el tamaño muestral el área de la elipse rápidamente disminuye en magnitud y varianza, lo cual sustenta nuestra intuición de la sección 2 en donde la gráfica de los EMV nos indicaba que los valores rápidamente se concentraban alrededor del valor real en una forma elíptica, a pesar de tener bastantes valores fuera de lo esperado para tamaños muestrales pequeños como consecuencia del alto azar en estos casos. A pesar de no tener los mismos resultados para los estimadores del método de momentos, podemos suponer razonablemente que si calculamos las áreas de las elipses para estos estimadores siempre vamos a encontrar un área mayor a el caso análogo para EMV dado que como nos indicaban las gráficas de la sección 2, por el método de momentos también encontramos una distribución elíptica pero con un rango mucho mayor de datos lo cual naturalmente llevaría a un área mayor.

## 5 Convergencia de la covarianza

Recordando que en la sección (3) obtuvimos para cada tamaño muestral 500 muestras de dicho tamaño, y acorde con el TLC del EMV, esperaríamos encontrar que la covarianza muestral de los  $\hat{\theta}_1, \dots, \hat{\theta}_{500}$  se aproxime a  $\frac{I_F(\theta_0)^{-1}}{n}$  o equivalentemente que  $nCov(\hat{\theta}_1, \dots, \hat{\theta}_{500})$  se aproxime a  $I_F(\theta_0)^{-1}$ . En la figura 9

```

#AREA DEL ELISPE:
AE = function(theta_0, EMVs) {
  areas = list(list(), list(), list(), list(), list())
  rho = qchisq(0.95, 1)

  for (n in n_list){
    ind = which(n_list == n)
    data = EMVs[[ind]]
    A_values = list()

    for (i in 1:length(data)) {
      d = c(data[[i]][1], data[[i]][2]) #Formateamos los datos para calcular el rango
      thetaVec = c(theta_0[[1]], theta_0[[2]])
      F_info = If(theta_0[[1]], theta_0[[2]])

      U = eigen(F_info)$vectors #Obtenemos U para tener la variable Y
      lambda = eigen(F_info)$values
      D = diag(lambda)
      Y = t(U) %*% (d - thetaVec)
      cov_Y = cov(Y)
      eigen_Y = eigen(cov_Y)
      eigenvalues = eigen_Y$values
      eigenvectors = eigen_Y$vectors

      a = sqrt(max(eigenvalues)) #Semiejes de la elipse
      b = sqrt(min(eigenvalues))
      Area = pi * a * b #Area de la elipse
      A_values = c(A_values, list(Area)) #Agregamos a la lista
    }
    areas[[ind]] = A_values
  }
}

```

Figure 7: Implementación del calculo de área de elipses para EMV.

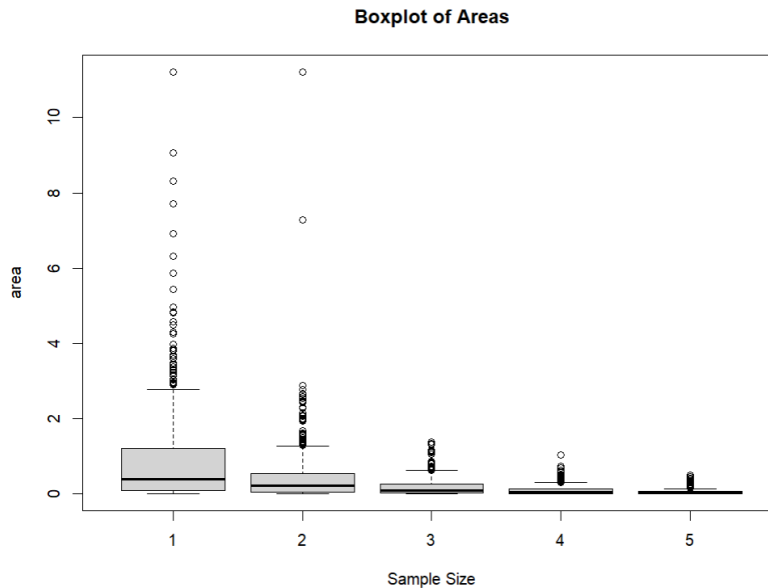


Figure 8: Boxplot para las áreas de las elipses del EMV en sus distintos tamaños muestrales

```

If_inv = solve(If(alpha_v, beta_v)) #Matriz inversa de If(theta_0)
covariances = list()
for (n in n_list){
  ind = which(n_list == n)
  EMV_val = unlist(EMVs[[ind]])
  x_list = list()
  y_list = list()

  for (k in 1:500){
    x = c(EMV_val[(k - 1) * 2 + 1])
    y = c(EMV_val[(k - 1) * 2 + 2])

    x_list[[k]] = x
    y_list[[k]] = y
  }
  x_matrix = matrix(c(unlist(x_list), unlist(y_list)), ncol = 2, byrow = FALSE)
  covariances[[ind]] = n*cov(x_matrix) #n*Cov a comparar con If(theta_0) inversa
}

```

Figure 9: Programa para obtener covarianza muestral de los 500 EMV para cada tamaño muestral.

se puede ver la implementación de este método en donde para cada  $n$ , creamos la matriz conteniendo todos los 500 EMV y obtenemos la covarianza de esta matriz lo cual nos permite comparar con la matriz inversa de la información de Fisher de los parámetros reales.

En una ejecución típica del código se encuentran los siguientes valores como ejemplo.

- La inversa de la matriz de información la cual es constante,  $I_F(\theta_0)^{-1} = \begin{pmatrix} 11.067 & -20.37 \\ -20.37 & 45.93 \end{pmatrix}$
- Para  $n = 100$ ,  $nCov = \begin{pmatrix} 12.98 & -21.43 \\ -21.43 & 47.13 \end{pmatrix}$
- Para  $n = 200$ ,  $nCov = \begin{pmatrix} 12.61 & -22.16 \\ -22.16 & 48.0 \end{pmatrix}$
- Para  $n = 500$ ,  $nCov = \begin{pmatrix} 11.79 & -22.34 \\ -22.34 & 50.72 \end{pmatrix}$
- Para  $n = 1000$ ,  $nCov = \begin{pmatrix} 10.74 & -19.43 \\ -19.43 & 44.34 \end{pmatrix}$
- Para  $n = 100$ ,  $nCov = \begin{pmatrix} 11.49 & -20.87 \\ -20.87 & 45.88 \end{pmatrix}$

Con estos resultados y con repetidas ejecuciones del código es claro que aunque el azar inherente de la simulación no permite que sea una tendencia completamente clara en todas las iteraciones, si se puede afirmar que incrementar el tamaño muestral rápidamente lleva a que  $nCov$  tienda a la información de Fisher inversa de los valores reales, incluso al simular el código con valores de  $n$  mayores a los exigidos en el enunciado también se mantenía esta rápida aproximación de tal forma que incluso  $nCov$  ya era prácticamente idéntica a la información inversa de Fisher. En general también se encontró que los parámetros mas propensos a error con respecto a  $I_F(\theta_0)^{-1}$  son aquellos de la anti-diagonal (es decir los dos valores idénticos), sin embargo en estos también se puede identificar la tendencia al valor esperado. El hecho de que se note claramente la correcta aproximación de estas matrices también nos indica que el proceso de obtención de los EMV fue ejecutado correctamente en etapas mas tempranas de la simulación.

## 6 Conclusiones

En base a los resultados computacionales obtenidos sobre la teoría discutida, podemos afirmar que los EMV para la Gamma( $\alpha, \beta$ ) son mas acertados con respecto a los valores reales que los estimados



por el método de momentos, sin embargo también se paga un precio a cambio de este aumento en la precisión en el incremento de complejidad computacional de implementar Newton-Raphson. Sin embargo, dada la rápida convergencia del método de NR, este precio del incremento en complejidad computacional es de poca relevancia pues en la maquina de prueba usada para ejecutar el programa la obtención de los 500 EMV para los 5 distintos tamaños muestrales no tarda mas de diez segundos en completar la ejecución. Después de notar que el precio computacional a pagar para generar los EMV no es sustancial, es claro en base a los métodos gráficos empleados que estos EMV efectivamente son considerablemente mas precisos. Esta precisión también nos permitió modelar la distribución de EMV como una elipse cuyo área se reduce rápidamente a medida que aumenta el tamaño muestral y a pesar de no tener el valor análogo para los estimadores de momento podemos afirmar con cierta confianza basada en los métodos gráficos de la sección (3) que de hacer el mismo proceso para los estimadores de momentos las áreas encontradas siempre serian mayores en comparación con las de EMV. Finalmente, para los EMV pudimos evidenciar la convergencia de la covarianza al valor esperado en función del tamaño muestral lo cual como parte final de nuestra practica nos permite justificar la correcta ejecución del proceso empleado para los EMV los cuales fueron usados repetidamente durante la simulación. El código completo utilizado para la simulación se puede encontrar adjunto a este documento o en: <https://github.com/AlejandroBorda/Proyectos-Estadistica> .

## References

- [1] Susan Colley, "Multi-Variable Newton's Method, Vector calculus", <https://azrael.digipen.edu/MAT180/NewtonsMethod.pdf>.