METODO MONTECARLO EN ESTIMACION DE PARAMETROS

Por: Luis German Polanco Contreras

Universidad de los Andes
lg.polanco75@uniandes.edu.co

Por: Angela Jaramillo Noratto

Universidad de los Andes
a.jaramillo234@uniandes.edu.co

Resumen:

Este trabajo presenta un análisis de la aplicación del metodo Monte Carlo en la estimación de parámetros poblacionales de diferentes distribuciones de probabilidad, para que de esta manera se evalue la eficiencia del metodo propuesto para su aplicación práctica. El trabajo primero desarrolla un algoritmo para generar distribuciones aleatorias a partir de una distribución uniforme por medio del método de la función inversa y se evalan los resultados obtenidos del algoritmo aplicado. A continuación se calculan diferentes estimadores de un parmetro poblacional de algunas distribuciones; estos estimadores estan dados por medio de aproximaciones de integrales que han utilizado el metodo Monte Carlo; aqui es evaluada la eficiencia, sesgo y ECM de los estimadores con el fin de analizar los resultados obtenidos con el procedimiento planteado. Al final del árticulo se presenta el código implementado en lenguaje R, utilizado para la obtenición de los resultados presentados aquí

Palabras clave: Metodo Montecarlo, sesgo, eficiencia, error cuadratico medio, ECM, distribucion de probabilidad, método de la función inversa.

Abstract:

This paper presents an analysis in the application of the Monte Carlo methods for the estimation of sample space parameters in some probability distributions; which leads to the evaluation of these procedures and their application in practical porposes. First, we introduce an algorithm to create random samples of dierent distributions from an uniform distribution using the inverse function method; those results are evaluated o conrm their suitability. Then we calculate some estimators for dierent sample space parameters; those calculations are made though the Monte Carlo method. Now it is possible to evaluate such estimators using dierent measures such as biasedness, eciency and mean square error (MSE). Finally it is presented the raw code, which was used to obtain the results presented in this paper, the language used is R.

Keywords: Monte Carlo method, biased, eciency, mean quadratic error, MCE, probability distribution, inverse function method.

1. Introducción

Este proyecto tiene varios propósitos: en primer lugar, se desea generar una muestra de datos aleatoria usando el método de F inversa para el caso particular de una $Beta(\alpha, 1)$. En segundo lugar se desea hacer la comparación, mediante simulación Monte Carlo, de dos estimadores del parámetro θ.

El método de F inversa para la primera parte, se utiliza por un lado, porque se sabe que una distribución $Beta(\alpha, 1)$ corresponde a datos en (0,1) con densidad: $f(x) = \alpha x^{(\alpha-1)}, x \in (0, 1)$. Por el otro lado, si F es una distribución continua, invertible en el rango de la variable correspondiente, y $U \sim Unif(0, 1)$, entonces $X = F^{(-1)}(U) \sim F$. Lo que significa que si le aplicamos F^{-1} a una $Unif(0, 1)$, obtenemos la muestra que necesitamos con distribución F.

Para la segunda parte, la simulación o el Método Monte Carlo es muy útil para proporcionar soluciones aproximadas a problemas con experimentos en los que se generan muestras de números pseudoaleatorios por computador para examinar algún problema. Hay varios tipos de Métodos Monte Carlo pero todos tienden a seguir un patrón particular: primero se define un dominio de entradas posibles, segundo, se generan entradas aleatoriamente a partir de una función de probabilidad sobre el dominio, tercero, se realiza un cálculo deterministico sobre las entradas y finalmente, se agregan los resultados. Por ejemplo, para modelar problemas físicos, es posible resolver ecuaciones que describen la interacción entre dos átomos de una forma sencilla, pero cuando hay cientos o miles de átomos ésto se hace imposible. Con los Métodos Monte Carlo, sistemas con muchos elementos y configuraciones complejas que están muestreados en un número aleatorio de configuraciones, pueden visaualizarse y analizarse como un todo.Además el l Método Monte Carlo, a diferencia de los métodos numéricos tiene un error absoluto que decrece como $\frac{1}{\sqrt{n}}$ en virtud del Teorema del Límite Central.

De esta forma, por medio del Método Monte Carlo compararemos el desempeño de dos estimadores para el parámetro α de la distribución $Beta(\alpha, 1)$

2. Generación de una Distribución Beta

Para la primera parte del proyecto, debemos generar una muestra aleatoria con distribución $Beta(\alpha, 1)$, para ello utilizamos el método de la F inversa, por medio del cuál sabemos que si F es una distribución continua e invertible en el rango de la variable correspondiente y adicionalmente contamos con una muestra $U \sim Unif(0, 1)$, por tanto, obtenemos que $F^{-1}(U) \sim F$.

Así pues generamos una muestra aleatoria de tamaño n=1000 con distribución uniforme de parámetros 0 y 1 y almacenamos ésta en un vector de tamaño n. Como tenemos que $f(x) = \alpha x^{\alpha-1}$ con $x \in (0, 1)$, de podemos ver que se sigue fácilmente $F(X) = x^{\alpha}$ y entonces $F^{-1}(y) = y^{1/\alpha}$. Aplicando entonces esta función inversa a los datos con distribución uniforme anteriormente generados, obtenemos una muestra con de datos $Beta(\alpha, 1)$ de tamaño n. Este mismo procedimiento ha de realizarse para diferentes parárametros $\alpha = 0{,}3$, 1 y 3.

2.1. Histogramas

Se requiere entonces verificar que los datos obtenidos con el método mencionado correspondan con una muestra de datos de la distribución anunciada; para ello se realizan histogramas comparativos entre muestras generadas con el método de la F inversa y el comando "$rbeta(n, \alpha, 1)$"de R, este procedimiento se realiza para cada α deseado. A continuación se presentan dichos histogramas comparativos entre las muestras deseadas.

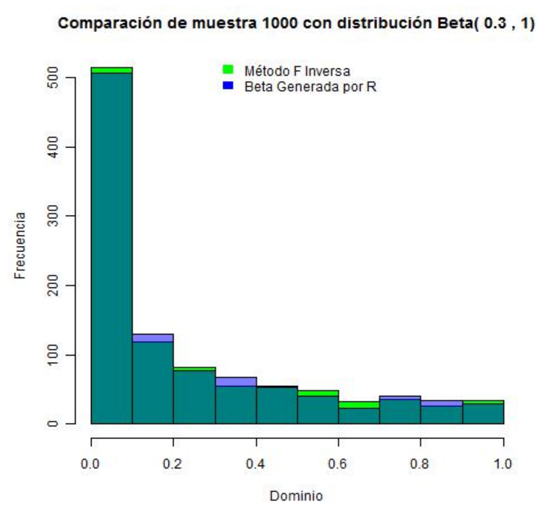

Figura 1: Histograma n=1000 y $Beta(\alpha, 1)$ con $\alpha = 0{,}3$

En el caso de $\alpha = 0{,}3$, podemos ver en la Figura 1 que la mayoría de los datos se acumulan en valores cercanos a 0 y su frecuencia va decreciendo drásticamente. Este comportamiento coincide con el esperado para una $Beta(0{,}3, 1)$; esto se puede verificar al comparar los datos obtenidos del método de la F inversa con aquellos generados aleatoriamente por R, mostrando así la confiabilidad del procedimiento sugerido para la generación de muestras aleatorias $Beta(\alpha, 1)$.

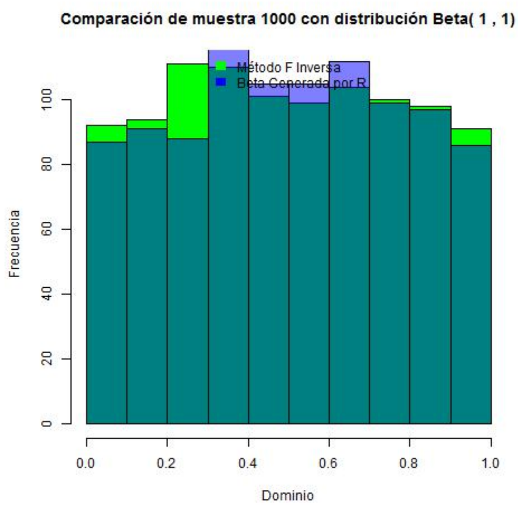

Figura 2: Histograma n=1000 y $Beta(\alpha, 1)$ con $\alpha = 1$

Cuando encontramos $\alpha = 1$, vemos claramente en la Figura 2, cómo una distribución $Beta(1, 1)$ tiende en distribución a una $Unif(0, 1)$, lo cual se puede verificar directamente de la fórmula de la densidad acumulada, pues cuando $\alpha = 1$ tenemos que $F(x) = x$, con $x \in (0, 1)$ podemos ver que todos los datos aparecen casi con la misma frecuencia a lo largo del dominio; cabe anotar adicionalmente que las diferencias presentes se deben al carácter aleatorio del procedimiento realizado al generar datos muestrales. Así mismo, también podemos observar que las dos distribuciones, tanto la generada por R y la generada por el método F inversa, se comportan similarmente como en el caso anterior.

En el caso de $\alpha = 3$, podemos ver en la Figura 3 que la mayoría de los datos se acumulan en valores cercanos a 1 y su frecuencia va aumentando drásticamente. Similarmente a los casos anteriores, podemos ver que este comportamiento coincide con el esperado para una $Beta(3, 1)$, y lo podemos verificar al comparando, otra vez, los datos obtenidos del método de la F inversa con aquellos generados aleatoriamente por R. Con esto reforzamos nuestra afirmación en la cual estamos mostrando así la confiabilidad del procedimiento sugerido para la generación de muestras aleatorias $Beta(\alpha, 1)$.

2.2. Q-Q Plots

El q-q plot es un tipo de gráfico que nos permite comparar dos muestras aleatorias que tienen distribuciones de probabilidad dadas; permitiendonos ver qué tan similares o qué tan discímiles son estas en su comportamiento a lo largo del dominio de definición de las mismas. En este grafico de probabilidades se comparan quiantil a quantil las distribuciones, poniendo los cuantiles de una ellas en el eje-x versus los cuantiles de la otra distribución en el eje y, haciendo posible establecer algún tipo de relacin entre estos. Esta gráfica es por tanto muy representativa, porque indica qué tan semejantes son las dos distribuciones. Si estas son altamenete concordantes, los puntos aparecen casi colineales (aunque no necesariamente) sobre la línea $y = x$. Por otro lado, entre menos colineales sean los datos, se observará que las dos distribuciones que están siendo comparadas no guardan una relacin lineal.

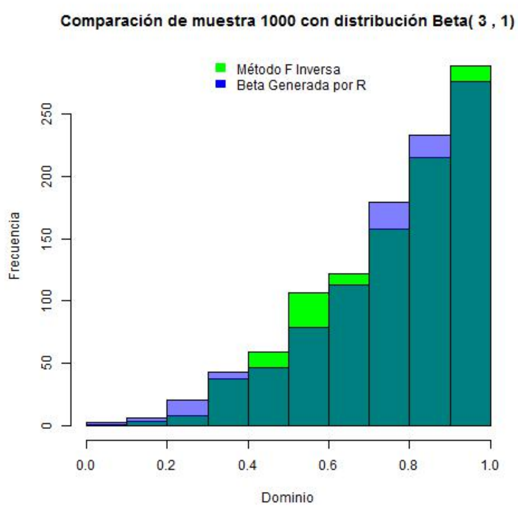

Figura 3: Histograma n=1000 y $Beta(3, 1)$

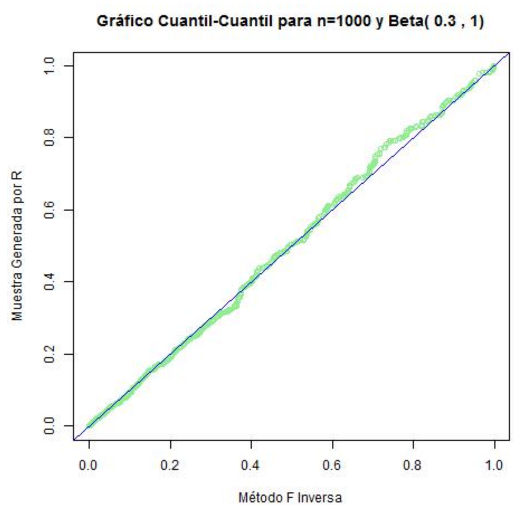

Figura 4: Q-Q Plot Beta(α,1) con $\alpha = 0{,}3$

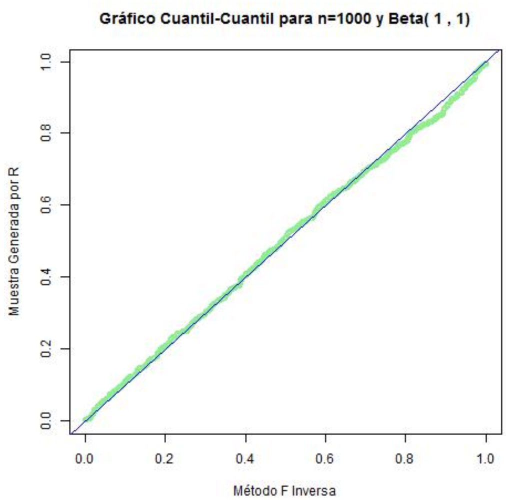

Figura 5: Q-Q Plot Beta(α,1) con $\alpha = 1$

Al comparar la muestra generada con el método de la F^{-1} y la muestra generada aleatoriamente por R, encontramos en las Figuras 4, 5 y 6 que cuantil a cuantil podemos evidenciar una fuerte concordancia entre las distribuciones acumuladas de las muestras comparadas. Estas gráficas confirman las conclusiones extraídas de los tres primeros histogramas donde afirmábamos que las distribuciones, tanto la generada por el método de F inversa, como la generada por R eran muy similares.

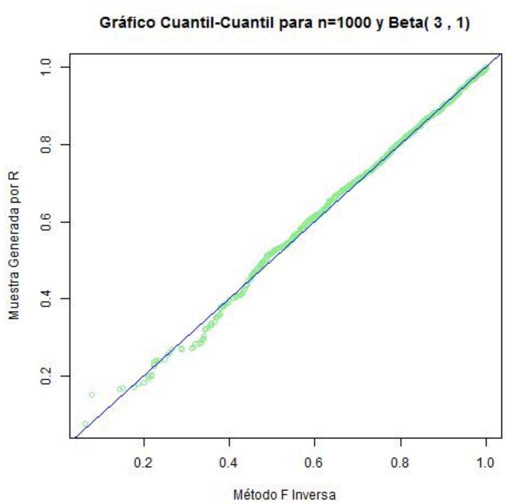

Figura 6: Q-Q Plot Beta(α,1) con $\alpha = 3$

Se puede ver también la frecuencia de los datos, cómo para $\alpha = 0{,}3$ estos se acumulan al principio (cerca de 0), mientras que para $\alpha = 1$ la dsitribucin de los mismos es mucho mas üniformeçomo se esperaria de una $Beta(1,1)$ y por último $\alpha = 3$ se observa que los datos se acumulan fuertemente al final de la gráfica, debido esto a la naturaleza misma de la distribucin construida. Todo esto es consistente con los datos obtenidos de los histogramas presentadoe en la sección 2.1.

Podemos concluir a grandes rasgos, que nuestro método de F inversa para generar una muestra con distribución $Beta(\alpha, 1)$, es muy adecuado y produce entonces una distribución muy similar a la $Beta$ generada por R (o $Beta$ teórica), se puede deducir también que independiente del parametro α utilizado (ya sea grande o pequeño) en la distribución de los datos se obtienen muestras consistentes y concordantes para la distribución esperada.

3. Comparación de Estimadores

Suponga que tenemos una una variable aleatoria X quetiene función de densidad de probabilidad $f(x;\theta)$, donde θ corresponde a un parámetro própio de la distribución estudiada. En la práctica muchas veces (sino la mayoría) desconocemos el valor de θ y debido a la imposibilidad de acceder a datos poblacionales, la única forma de acercarnos a θ es por medio de muestras (”samples”) de la población que nos permitan estimar el parámetro buscado.

Es entonces importante saber cuando un estimador es *bueno*, es decir, toma valores cercanos al parámetro deseado; por ello es necesario estudiar el comportamiento del mismo. Para ello se tienen: el sesgo, la varianza y el error cuadrático medio (ECM); quienes nos permiten medir que tan acertado es un estimador.

El sesgo se define como la diferencia entre el valor esperado del estimador y el valor real del parametro y se nota como $Sesgo(\widehat{\theta}) = E(\widehat{\theta}) - \theta$; con base en el sesgo se calcula la varianza del mismo como $Var(\widehat{\theta}) = E[(E(\widehat{\theta}) - \theta)^2]$ y finalmente el ECM del estimador $\widehat{\theta}$ se define como $ECM(\widehat{\theta}) = E[(\widehat{\theta} - \theta)^2]$.

Adicionalmente vale la pena mencionar que muchas veces se cuenta con más de un estimador para un mismo párametro poblacional, por lo que es necesario dar un criterio de comparación entre los estimadores. Es por ello que deseamos halar de la *eficiencia relativa* de dos estimadores, definida como:

$$eff(\widehat{\theta}_1, \widehat{\theta}_2) = \frac{ECM(\widehat{\theta}_1)}{ECM(\widehat{\theta}_2)}$$

dando esta un criterio sencillo y muy útil al momento comparar diferentes estimadores del mismo parámetro.

Si recordamos la definición del valor esperado de una variable aleatoria continua:

$$E(x) = \int_{-\infty}^{\infty} xf(x)dx$$

podemos ver la pertienencia de la aplicación del Método Montecarlo para el cálculo del sesgo, varianza y ECM de un estimador llevandonos esto al sigueinte desarollo y análisis de los datos obtenidos con dicho método.

3.1. Boxplot

Como mencionábamos en la introducción por medio del Método Monte Carlo queremos comparar el desempeño de dos estimadores para el parámetro α de la distribución $Beta(\alpha, 1)$. Tenemos por un lado, el estimador de momentos

$$\hat{\theta}_1 = \frac{\bar{X}}{1-\bar{X}}$$

y por otro lado, el estimador de máxima verosimilitud

$$\hat{\theta}_2 = \frac{-n}{\Sigma ln(X_i)}$$

Como vimos en clase, el boxplot es un método que captura gráficamente información acerca de un grupo de datos a partir de un resumen de cinco datos: el mínimo, el primer cuartil, la mediana, el tercer cuartil y el máximo,. Adicionalmente esta representación indica qué valores deben ser considerados como 'outliers' o atípicos ya que los bigotes (whiskers), se grafican, en su mayora a una distancia de 1,5 por la Distancia Inter-Cuartil por debajo del primer cuartil y por arriba del tercero. Los datos por fuera de esta distancia se consideran 'outliers'. en conlcusión gráficamente es posible ver la dispersión y el sesgo de los datos así como la distancia intercuartil como medida de la varianza de los mismos.

3.1.1. Boxlpot Comparativo Entre Estimadores

A continuación graficaremos, en los ı ejes los boxplots de $\hat{\theta}_1$ y $\hat{\theta}_2$, para $\alpha = 0{,}3$, 1, junto con diferentes tamaños de muestras tribuciones $Beta(\alpha, 1)$ correspondientes a ı 50, 100, 200, 500.

Distribución Beta(0.3, 1)

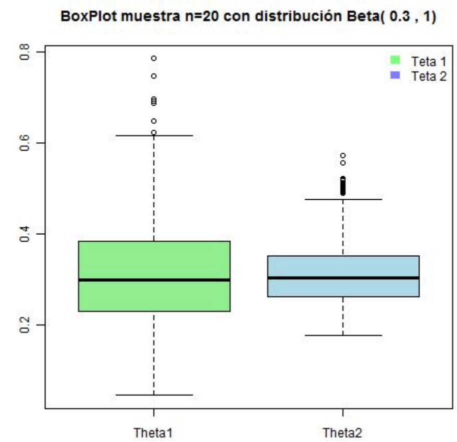

Figura 7: BoxPlot n=20 y Beta(0.3 , 1)

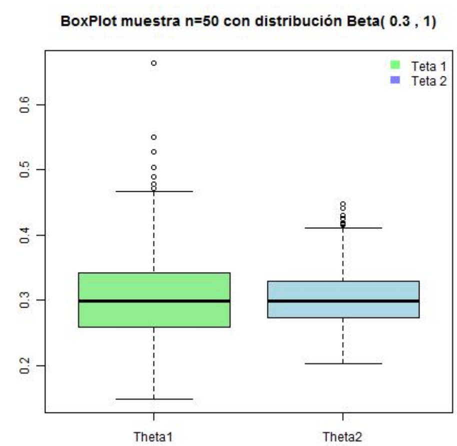

Figura 8: BoxPlot n=50 y Beta(0.3 , 1)

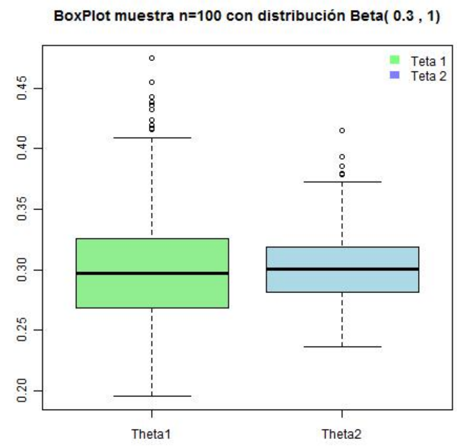

Figura 9: BoxPlot n=100 y Beta(0.3 , 1)

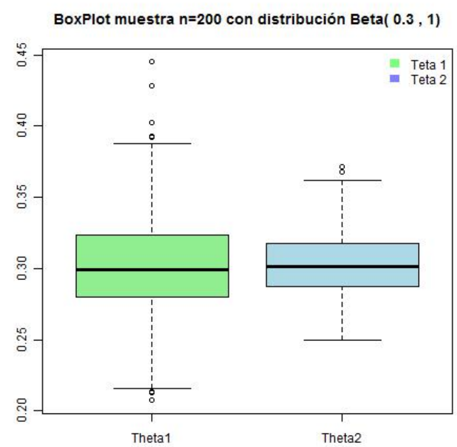

Figura 10: BoxPlot n=200 y Beta(0.3 , 1)

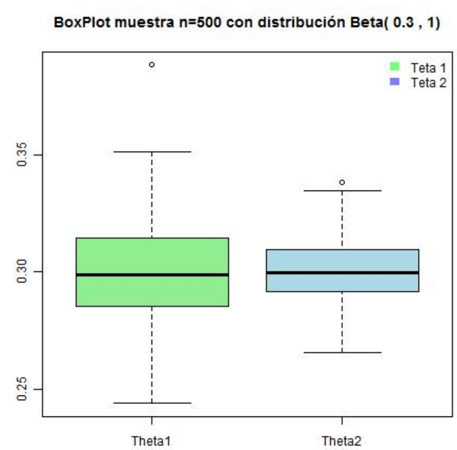

Figura 11: BoxPlot n=500 y Beta(0.3 , 1)

Distribución Beta(1,1)

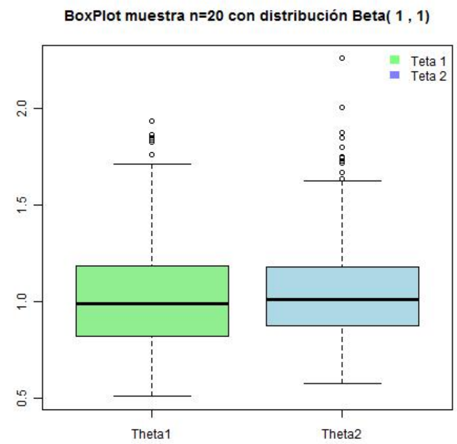

Figura 12: BoxPlot n=20 y Beta(1 , 1)

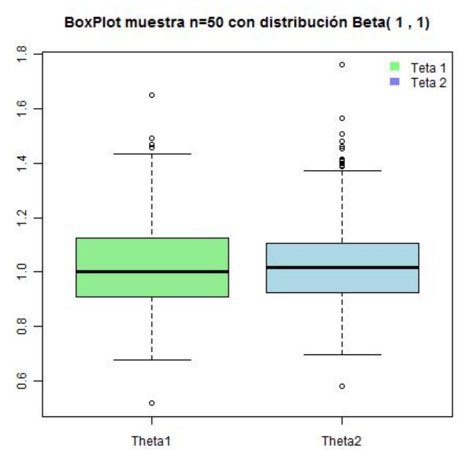

Figura 13: BoxPlot n=50 y Beta(1 , 1)

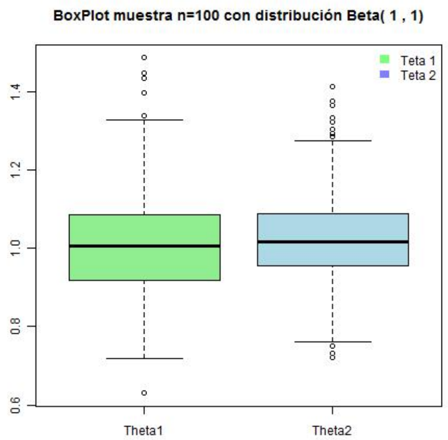

Figura 14: BoxPlot n=100 y Beta(1 , 1)

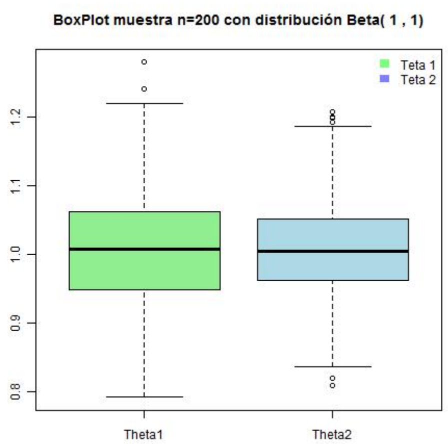

Figura 15: BoxPlot n=200 y Beta(1 , 1)

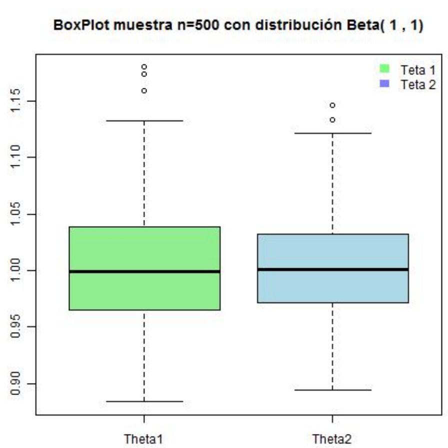

Figura 16: BoxPlot n=500 y Beta(1 , 1)

Distribución Beta(3,1)

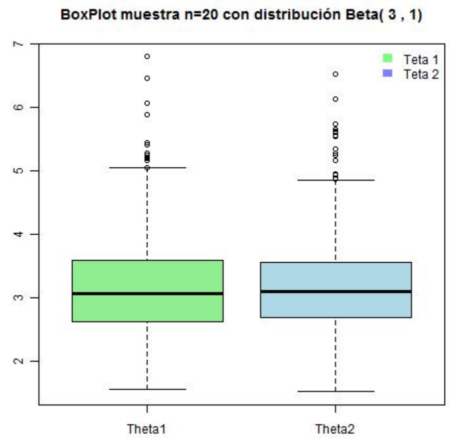

Figura 17: BoxPlot n=20 y Beta(3 , 1)

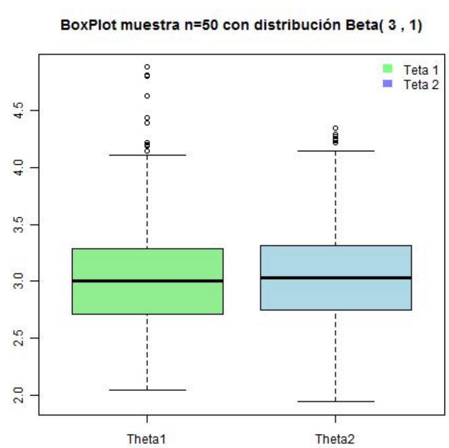

Figura 18: BoxPlot n=50 y Beta(3 , 1)

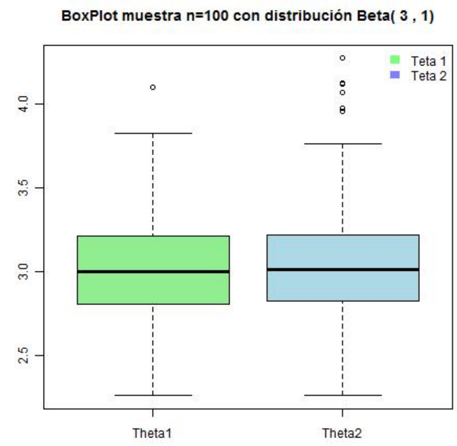

Figura 19: BoxPlot n=100 y Beta(3 , 1)

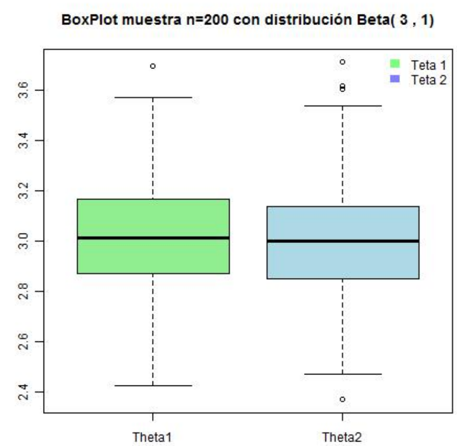

Figura 20: BoxPlot n=200 y Beta(3 , 1)

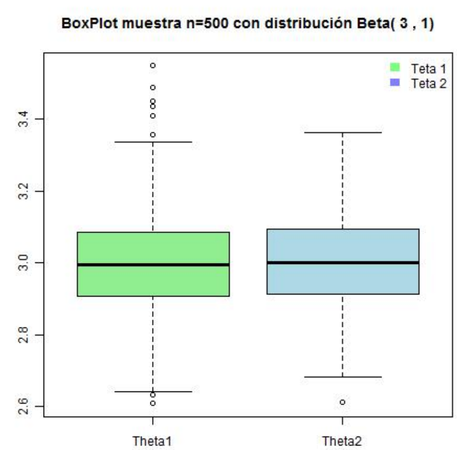

Figura 21: BoxPlot n=500 y Beta(3 , 1)

Distribución Beta(10,1)

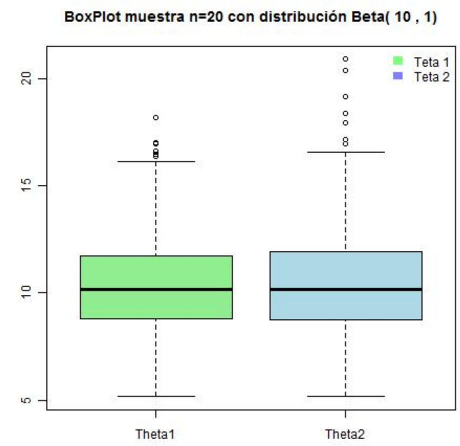

Figura 22: BoxPlot n=20 y Beta(10 , 1)

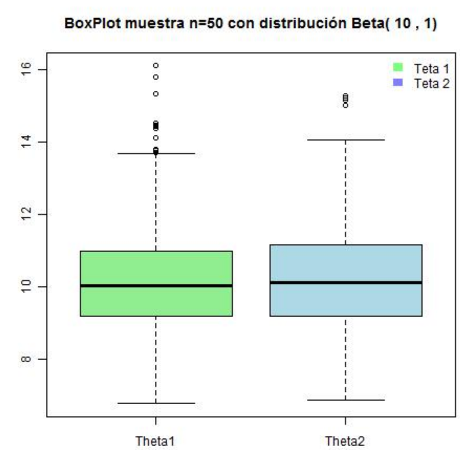

Figura 23: BoxPloty n=50 y Beta(10 , 1)

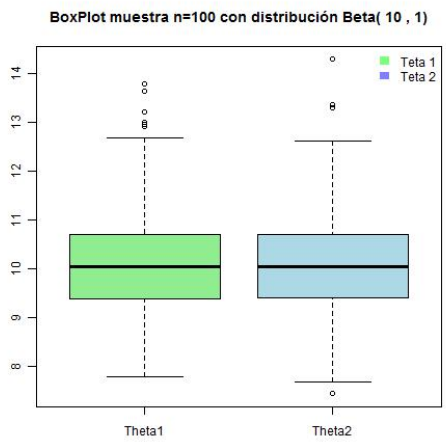

Figura 24: BoxPlot n=100 y Beta(10 , 1)

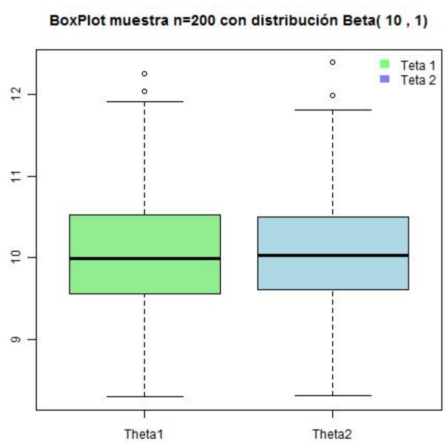

Figura 25: BoxPlot y n=200 y Beta(10 , 1)

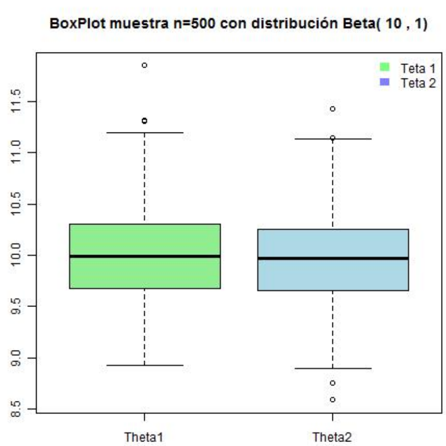

Figura 26: BoxPlot n=500 y Beta(10 , 1)

Aunque las grficas a primera vista se ven muy parecidas para cada valor fijo de α, y donde para cada gráfica varía el valor de n, podemos darnos cuenta con la escala en el eje y de los boxplot que, a medida de que aumenta n, su Distancia Inter Cuartil, su Error Cuadrático Medio (y por lo tanto su varianza y su sesgo) disminuyen. Gráficamente se ve que los datos se van agrupando hacia el centro y la mediana se mantiene fija.

También podemos darnos cuenta que los dos poxplot, con un α fijo para cada estimador son muy similares, aunque se notan pequeñas diferencias para $\alpha = 0{,}3$ pues en las Figuras 7, 8, 9, 10 y 11 se puede notar que el estimador $\widehat{\theta}_2$ tiene menor distancia intercuartíl y la diferencia entre los datos extremos es menor que para $\widehat{\theta}_1$, pese a esto dada la aleatoriedad de los datos, después de correr el programa varias veces, se puede constatar que ningún estimador es contundentemente mejor que el otro.

El comportamiento anteriormente mencionado se presenta también con $\alpha = 1$, pero la diferencia entre los estimadores es mucho menor como se evidencia en las Figuras 12, 13, 14, 15 y 16 y disminuye mucho más notoriamente cuando aumenta el tamao de la muestra n, siendo casi inperceptible cuando $n = 500$; no obstante no hay diferencias contundentes entre ambos estimadores.

En las figuras 17, 18, 19, 20 y 21, correspondientes a $\alpha = 3$ no se puede evidenciar diferencia significativa entre los estimadores, lo mismo sucede cuando $\alpha = 10$ y esto es sencillo de ver al observar con detenimiento las Figuras 22, 23, 24, 25 y 26, en donde es posible predecir que la diferencia entre los 2 estimadores será irrelevante.

Cabe anotar que los valores atípicos presentados en las diferentes muestras de estimadores son producto del carácter aleatorio de las muestras $Beta(\alpha, 1)$ y de las muestras de estimadores, tanto $\widehat{\theta}_1$ como $\widehat{\theta}_2$; motivo por el cúal se presentan dichos datos 'outliers'.

Es entonces razonable concluir que para α pequeño (preferiblemente menor que 1) y para tamaños de muestra pequeos ($n = 20$ y 50) el estimador $\widehat{\theta}_2$ presenta un comportamiento con menor varianza que $\widehat{\theta}_1$, sin embargo esta tendencia desaparece rapidamente con el aumento de n y más aún con el aumento de α.

3.1.2. Boxplot Comparativo Entre Muestras

Quisimos incluir gráficas comparativas entre las diferentes muestras, en ellas se grafican en un mismo eje los cinco boxplot correspondientes a cada valor de $n = 20$, 50, 100, 200 y 500, para cada α fijo y un solo estimador, ya sea $\hat{\theta}_1$ ó $\hat{\theta}_2$. De esta manera podemos comprobar nuestras conclusiones tras el análisis de los boxplots comparativos entre estimadores, en los cuales notábamos que la Distancia Inter Cuartil, el Error Cuadrático Medio, la varianza y el sesgo disminuían con el aumento de n.

Se ve claramente en las Figuras 27, 28, 29, 30, 31, 32, 33 y 34 cómo se acercan los datos hacia la mediana, cómo se acercan los 'outliers', se reducen los bigotes y la distancia intercuartiles mientras la mediana permanece fija. Esto parece indicarnos que ambos estimadores son asintóticamente insesgados y consistentes, y que ambos se comportan práctimente igual.

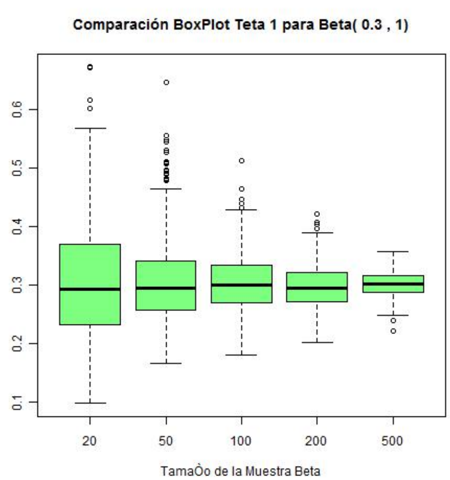

Figura 27: BoxPlot Comparativo $\hat{\theta}_1$ Beta(0.3 , 1)

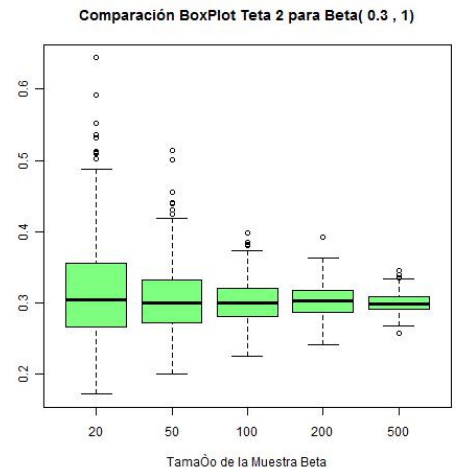

Figura 28: BoxPlot Comparativo $\hat{\theta}_2$ Beta(0.3 , 1)

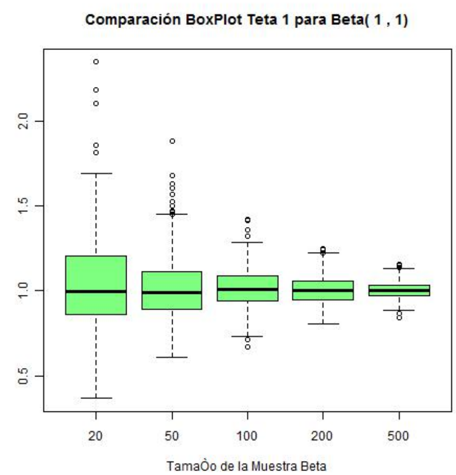

Figura 29: BoxPlot Comparativo $\hat{\theta}_1$ Beta(1 , 1)

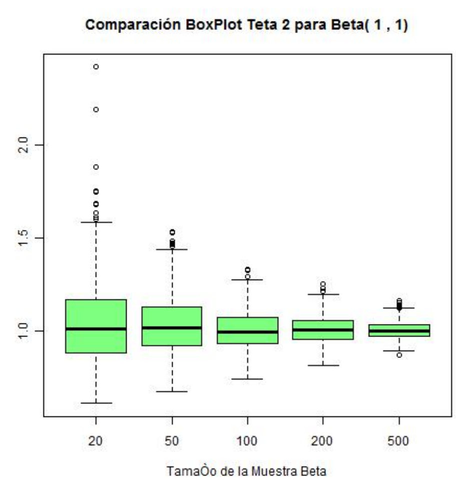

Figura 30: BoxPlot Comparativo $\hat{\theta}_2$ Beta(1 , 1)

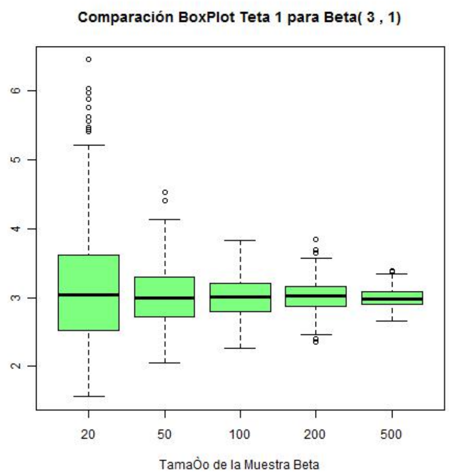

Figura 31: BoxPlot Comparativo $\hat{\theta}_1$ Beta(3 , 1)

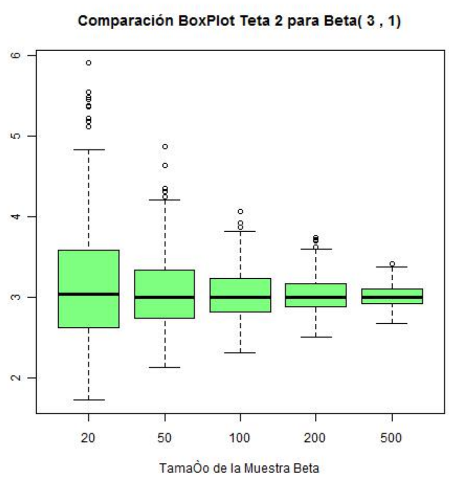

Figura 32: BoxPlot Comparativo $\hat{\theta}_2$ Beta(3 , 1)

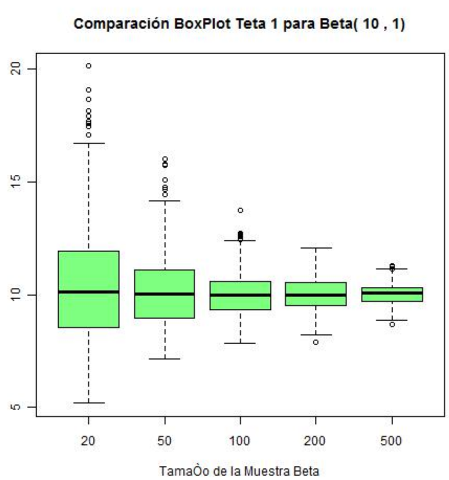

Figura 33: BoxPlot Comparativo $\hat{\theta}_1$ Beta(10 , 1)

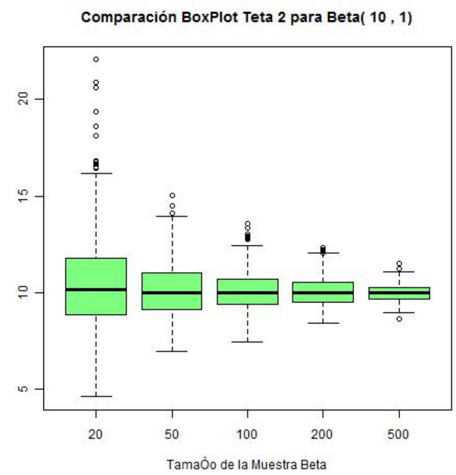

Figura 34: BoxPlot Comparativo $\hat{\theta}_2$ Beta(10 , 1)

3.2. Sesgo, ECM y Eficiencia

A continuación analizaremos las tablas del sesgo, ECM para $\widehat{\theta}_1$ y $\widehat{\theta}_2$ y la eficiencia relativa de cada par de estimadores, para todos los valores de α y de n.

Recordemos que el sesgo es la diferencia entre la esperanza matemática y el valor numérico del parámtreo que se está estimando, es decir, en este caso es $\hat{\theta}_i - \alpha, i = 1, 2$.

El Error Cuadrático Medio mide, para un estimador, en promedio el error en la estimación y se define por $E[(\hat{\theta}_i - \alpha)^2], i = 1, 2$. Y recordemos que una de sus propiedades más interesantes es que éste puede descomponerse como la suma de la varianza del estimador y el sesgo al cuadrado (i.e. $E[(\hat{\theta}_i - \alpha)^2] = V(\hat{\theta}_i + sesgo^2(\hat{\theta}_i)))$.

La Eficiencia, a su vez, indica cuál de los estimadores usados es mejor. Se dice que un estimador $\hat{\theta}_1$ es más eficiente o más preciso que otro estimador $\hat{\theta}_2$, si $Var(\hat{\theta}_1) < Var(\hat{\theta}_2)$. Ésta se calcula con la fórmula $eff(\theta_1, \theta_2) = \frac{ECM(\theta_1,\alpha)}{ECM(\theta_2,\alpha)}$.

Con las Tablas 1 y 2 dodne está el sesgo para $\hat{\theta}_1$ y $\hat{\theta}_2$, podemos comprobar que el sesgo tiende a 0 a medida de que aumenta n, es decir, que son asintoticamente insesgados, esto sucede para cada α, pese a que tiende a 0 mas rapidamente para α pequeños, pero esto no afecta la convergenciadel sesgo cuando $n \to \infty$.

Con las Tablas 3 y 4 de ECM para $\hat{\theta}_1$ y $\hat{\theta}_2$, también podemos ver cómo el error cuadrático medio disminuye a medida de que aumenta n, permitiendonos deducir que gracias a que ya $sesgo(\hat{\theta}_i) \to 0$ y el $ECM(\hat{\theta}_i) \to 0$, entonces $var(\hat{\theta}_i) \to 0$. De lo que se sigue facilmente que los dos estimadores $\hat{\theta}_1$ y $\hat{\theta}_2$ son consistentes.

La tabla 5 con los datos de la eficiencia relativa nos muestran que para $\alpha = 0,3$ sí hay una desproporción entre los ECM de los estimadores, pero para $\alpha = 1, 3, 10$ los valores, al oscilar alrededor de 1, por encima y por abajo, nos indican que los dos estimadores están muy cercanos.

3.3. Comparación Logarítmica entre Estimadores

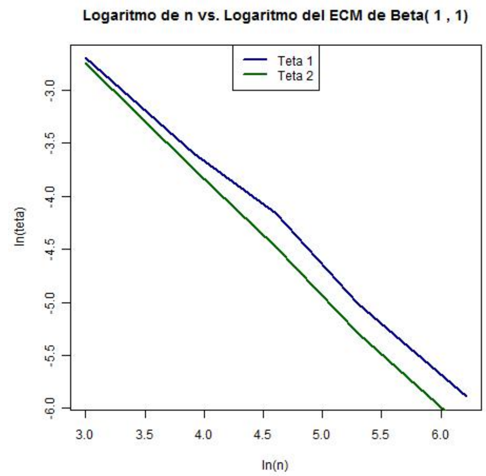

Figura 36: $ln(n)$ vs. $ln(ECM(\theta_i))$, Beta(1 , 1)

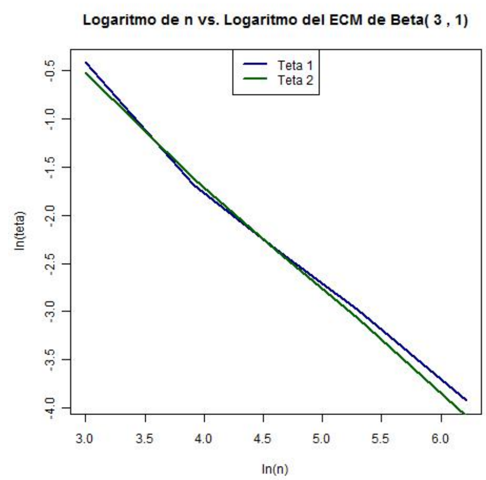

Figura 37: $ln(n)$ vs. $ln(ECM(\theta_i))$, Beta(3 , 1)

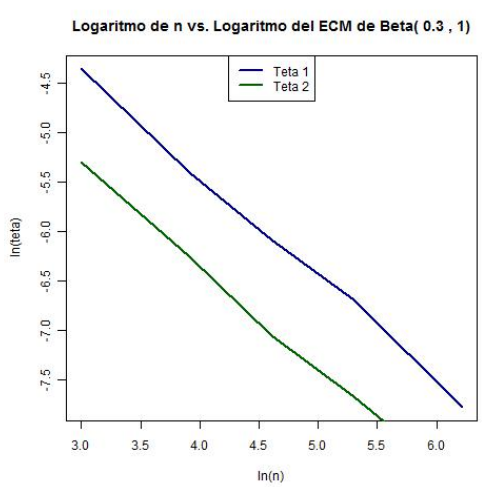

Figura 35: $ln(n)$ vs. $ln(ECM(\theta_i))$, Beta(0.3 , 1)

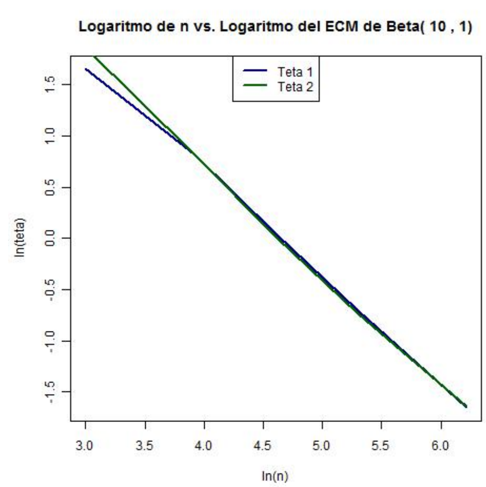

Figura 38: $ln(n)$ vs. $ln(ECM(\theta_i))$, Beta(10 , 1)

Con las gráficas de comparación logarítmica entre estimadores, podemos concluir finalmente que el ECM decrece de manera exponencial como función de n. Cuando uno grafica en un log-log plot, cualquier función de la forma $f(x) = cx^m$, va a tener una línea como representación log-log. Podemos ver entonces que la relación entre n y el ECM es de manera exponencial decreciente. Est revela, a su vez, la exactitud de de los dos estimadores.

Bibliografía

[1] Robert V. Hogg, Joseph W. McKean, and Allen Thornton Craig. *Introduction To Mathematical Statistics*. Pearson Education, 6th edition, 2006.

[2] D.J.C. Mackay. Introduction to monte carlo methods. http://www.stat.ucla.edu/ yuille/courses/Stat202C-Spring10/mackay.pdf, 2010. Department of Physics, Cambridge University.

[3] Albert Tarantola. *Inverse Problem Theory and Methods for Model Parameter Estimation*. Society for Industrial and Applied Mathematics, 2005.

Anexos

Tablas

Sesgo Teta 1	0.3	1	3	10
20	0.016263101	0.019153363	0.192511419	0.349944767
50	0.002706895	0.019236057	0.027948655	0.199659206
100	0.00005549	0.009669141	0.023828616	0.099576694
200	0.00122311	0.006925168	0.024794306	0.052513011
500	0.00004226	0.001862872	0.000439207	0.000964085

Tabla 1: Sesgo de θ_1 para cada n

Sesgo Teta 2	0.3	1	3	10
20	0.012763875	0.047478597	0.186157994	0.487962216
50	0.003179878	0.026283502	0.057003354	0.25732896
100	0.002108188	0.02004941	0.039965911	0.101612271
200	0.002208405	0.007368937	0.006772574	0.069292246
500	0.000196527	0.002160897	0.00115006	0.030592754

Tabla 2: Sesgo de θ_2 para cada n

ECM Teta 1	0.3	1	3	10
20	0.012906084	0.067393753	0.658582985	5.215248834
50	0.004513180	0.027560307	0.184856915	2.271533908
100	0.00228814	0.015633733	0.095349428	1.040041215
200	0.00124277	0.006616489	0.050372459	0.493174620
500	0.00041638	0.002769774	0.019772687	0.191227269

Tabla 3: ECM de θ_1 para cada n

ECM Teta 2	0.3	1	3	10
20	0.005021383	0.064317430	0.590704419	6.436476150
50	0.001915980	0.023699504	0.196519712	2.275815352
100	0.00086268	0.011381703	0.093779061	1.008492401
200	0.00046270	0.005017827	0.046348879	0.479969211
500	0.00018746	0.002012649	0.016917261	0.194456589

Tabla 4: ECM de θ_2 para cada n

Eficiencias	0.3	1	3	10
20	2.570224737	1.047830326	1.114911221	0.81026461
50	2.355546602	1.162906488	0.9406533	0.99811872
100	2.652347838	1.373584607	1.016745396	1.031283145
200	2.685867468	1.31859654	1.086810731	1.027513034
500	2.221168257	1.376182941	1.168787719	0.983393105

Tabla 5: $Eff(\theta_1, \theta_2)$ Eficiencia relativa

Código

```
#———————————————————————————
# INFORMACION FUNCIONAMIENTO PROGRAMA
#———————————————————————————

separador=";"
parar=FALSE
guardar="NO"
guardar=winDialog(type ="yesno", "El siguiete programa mostrar cada una de las
    graficas requeridas. Usted debe presionar la tecla [ENTER]  para continuar a la
    siguiente gr fica. Desea guardar las graficas en el directorio de R")

#———————————————————————
#——— PARTE 1 DEL PROYECTO ———
#———————————————————————

#———————————————————————
#———— FUNCIONES BSICAS ————
#———————————————————————

#——————————————————————————————————————————————
#  Funcion que genera una muestra beta aleatoria de parametros
#  alfa y 1 con el m todo de F inversa
#——————————————————————————————————————————————
#——————————————— PARAMETROS ——————————————————
#  n = tama o muestra
#  alfa = parametro alfa de la distribucion beta
#——————————————————————————————————————————————

genBeta_F_Inver=function(alfa,n)
{
  u = runif(n)
  beta=rep(0,n)
  for(i in 1:n)
  {
    beta[i] = (u[i])^(1/alfa)
  }
  return (beta)
}

#————————————————————————————————————————————
#  Funcion que crea un Histograma comparando la muestra
#  beta construida con el
#  m todo de F inversa con la muestra obtenida usando R
#————————————————————————————————————————————
#——————————————— PARAMETROS ————————————————
#  n = tama o muestra
#  alfa = parametro alfa de la distribucion beta
#————————————————————————————————————————————

compararGraficosBeta=function(alfa,n)
{
  t1=genBeta_F_Inver(alfa,n)
  t2=rbeta(n,alfa,1)

#———————————— Primero Guardo el plot ————————

  if(guardar=="YES")
  {
    a=sprintf("/histograma%iyBeta(%s,1).jpg",n,alfa)
    jpeg(paste("E:/imagenes",a,sep=""))

    hist(t1, main = "", col=rgb(0,1,0),xlab="Dominio",ylab="Frecuencia")
```

```
    title(main=sprintf("Comparacin de muestra %i con distribucin Beta( %s , 1)",n,
        alfa))
    hist(t2,add=T,col= rgb(0,0,1,0.5))
    legend('top',c('Mtodo F Inversa','Beta Generada por R'), fill = rgb
        (0:0.5,1:0,0:1), bty = 'n', border = NA)
    dev.off()
  }

#—————————— luego se visualiza ——————————

  if(parar)
  {
    hist(t1, main = "", col=rgb(0,1,0),xlab="Dominio",ylab="Frecuencia")
    title(main=sprintf("Comparacin de muestra %i con distribucin Beta( %s , 1)",n,
        alfa))
    hist(t2,add=T,col= rgb(0,0,1,0.5))
    legend('top',c('Mtodo F Inversa','Beta Generada por R'), fill = rgb
        (0:0.5,1:0,0:1), bty = 'n', border = NA)
  }
}

#————————————————————————————————————
#  Funcion que crea un grafica en la que se comparan
#  los cuantiles de la muestra beta construida con el
#  mtodo de F inversa  vs. los cuntiles de la muestra
#  aleatoria obtenida usando R
#————————————————————————————————————
#———————————— PARAMETROS ————————————
#  n = tamao muestra
#  alfa = parametro alfa de la distribucion beta
#————————————————————————————————————

compararGraficosBetaQ_Q=function(alfa,n)
{
  t1=genBeta_F_Inver(alfa,n)
  t2= rbeta(n,alfa,1)

#—————————— Primero Guardo el plot ——————————

  if(guardar=="YES")
  {
    a=sprintf("/qqplot%iyBeta(%s,1).jpg",n,alfa)
    jpeg(paste("E:/imagenes",a,sep=""))
    qqplot(t1,t2, main=sprintf("Grfico Cuantil−Cuantil para n=%i y Beta( %s , 1)",n,
        alfa), xlab="Mtodo F Inversa", ylab="Muestra Generada por R", col="lightgreen"
        )
    abline(a=0, b=1, col="blue")
    dev.off()
  }

#—————————— luego se visualiza ——————————

  if(parar)
  {
    qqplot(t1,t2, main=sprintf("Grfico Cuantil−Cuantil para n=%i y Beta( %s , 1)",n,
        alfa), xlab="Mtodo F Inversa", ylab="Muestra Generada por R", col="lightgreen"
        )
    abline(a=0, b=1, col="blue")
  }
}

#—————— EJECUCION FINAL ——————

#——————————————————————————————————————————————
# Comparaciones entre el metodo f inversa y muestraS generadas por R
```

```
#-------------------------------------------------------------------------------

#--Distribucion Beta con muestra 1000 alfa 0.3--
compararGraficosBeta(0.3,1000)
par(ask=parar)

#--Distribucion Beta con muestra 1000 alfa 1--
compararGraficosBeta(1,1000)
par(ask=parar)

#--Distribucion Beta con muestra 1000 alfa 3--
compararGraficosBeta(3,1000)
par(ask=parar)

#-------------------------------------------------------------------------------
# Graficos Cuantil-Cuantil entre F inversa y muestras generadas por R
#-------------------------------------------------------------------------------

#--Distribucion Beta con muestra 1000 alfa 0.3--
compararGraficosBetaQ_Q(0.3,1000)
par(ask=parar)

#--Distribucion Beta con muestra 1000 alfa 1--
compararGraficosBetaQ_Q(1,1000)
par(ask=parar)

#--Distribucion Beta con muestra 1000 alfa 3--
compararGraficosBetaQ_Q(3,1000)
par(ask=parar)

#-----------------------------------------------
#----------- PARTE 2 DEL PROYECTO -----------
#-----------------------------------------------

#------------------------------------
#---------- FUNCIONES ----------
#------------------------------------

#-----------------------------------------------------
# Funcion que retorna el valor del estimdor
# teta 1 de una distribuci n aleatoria obtenida
# con el m todo f inversa
#-----------------------------------------------------
#----------------- PARAMETROS -----------------
# n = tama o muestra
# alfa = parametro alfa de la distribucion beta
#-----------------------------------------------------

darTeta1=function(alfa,n)
{
  teta1=0
  beta=genBeta_F_Inver(alfa,n)
  p=mean(beta)
  teta1=(p/(1-p))
  return(teta1)
}

#--------------------------------------------------------------
# Funcion que retorna el valor del estimdor teta 2
# de una distribuci n aleatoria obtenid con el m todo
# f inversa
#--------------------------------------------------------------
#-------------------- PARAMETROS --------------------
# n = tama o muestra
# alfa = parametro alfa de la distribucion beta
```

```
#----------------------------------------------------------------

darTeta2=function(alfa,n)
{
  teta2=0
  beta = genBeta_F_Inver(alfa,n)
  suma = sum(log(beta))
  teta2= (-n)/(suma)
  return (teta2)
}

#------------------------------------------------------------------
#  Funcion que retorna un vector con el tamao solicitado
#  de los valores del estimador teta 1 o 2 para un muestra
#  aleatoria obtenida a traves del mtodo f inversa para
#  un alfa y un n dados
#------------------------------------------------------------------
#------------------------ PARAMETROS ------------------------------
#  n = tamao muestra
#  alfa = parametro alfa de la distribucion beta
#  id = identifica si es el etimador 1 o 2
#  muestra = la cantidad de estimadore que se quiere
#------------------------------------------------------------------

darMuestraEstimador=function(id,muestra,alfa,n)
{
  res = rep(0,muestra)
  if(id==1)
  {
    for(i in 1:muestra)
    {
      res[i] = darTeta1(alfa,n)
    }
  }
  else if(id == 2)
  {
    for(i in 1:muestra)
    {
      res[i] = darTeta2(alfa,n)
    }
  }
  return (res)
}

#--------------------------------------------------------------------
#  Funcion que retorna el sesgo del indicador solicitado
#--------------------------------------------------------------------
#------------------------ PARAMETROS --------------------------------
#  muestraEstimadores = la muestra del estimador que se
#  desea el sesgo
#  alfa = parametro alfa de la distribucion beta
#--------------------------------------------------------------------

darSesgo=function(muestraEstimadores,alfa)
{
  t=muestraEstimadores
  esperanza=mean(t)
  sesgo=esperanza-alfa
  return (sesgo)
}

#--------------------------------------------------------------------
#  Funcion que retorna error cuadratico medio de del
#  estimador indicado
#--------------------------------------------------------------------
```

```
#-------------------------------- PARAMETROS --------------------------------
#  muestraEstimadores = la muestra del estimador que se
#  desea el error cuadratico medio
#  alfa = parametro alfa de la distribucion beta
#----------------------------------------------------------------------------

darECM = function(muestraEstimadores,alfa)
{
  t=muestraEstimadores
  varianza = var(t)
  ecm= varianza + (darSesgo(t,alfa))^2
  return(ecm)
}

#----------------------------------------------------------------------------
#  Funcion que retorna la eficiencia relativa del estimador
#  teta 1 respecto al estimador teta 2
#----------------------------------------------------------------------------
#------------------------------ PARAMETROS ----------------------------------
#  muestraTeta1 = la muestra del estimador teta 1
#  muestraTeta2 = la muestra del estimador teta 2
#  alfa = parametro alfa de la distribucion beta
#----------------------------------------------------------------------------

darEficiencia = function(muestraTeta1,muestraTeta2,alfa)
{
  eficiencia=darECM(muestraTeta1,alfa)/darECM(muestraTeta2,alfa)
  return(eficiencia)
}

#----------------------------------------------------------------------------
#  Funcion que grafica un boxplot de las distintas muestras
#  dadas como parametros
#----------------------------------------------------------------------------
#------------------------------ PARAMETROS ----------------------------------
#  n = tamao muestra
#  alfa= parametro alfa de la distribucion beta
#  muestraTeta1 = la muestra de Teta 1
#  muestraTeta1 = la muestra de Teta 2
#----------------------------------------------------------------------------

crearBoxPlot=function(n,alfa,muestraTeta1,muestraTeta2)
{

#-------------------------- Primero Guardo el plot --------------------------

  if(guardar=="YES")
  {
    a=sprintf("/boxplotmuestra%iyBeta(%s,1).jpg",n,alfa)
    jpeg(paste("E:/imagenes",a,sep=""))
    boxplot(muestraTeta1, muestraTeta2, main=sprintf("BoxPlot muestra n=%i con
        distribuci n Beta( %s , 1)",n,alfa), col=(c("lightgreen","lightblue")), names=c
        ("Theta1","Theta2"))
    legend('topright',c('Teta 1','Teta 2'), fill = rgb(0:0,1:0,0:1,0.5), bty = 'n',
        border = NA)
    dev.off()
  }

#-------------------------- luego se visualiza ------------------------------

  if(parar)
  {
    boxplot(muestraTeta1, muestraTeta2, main=sprintf("BoxPlot muestra n=%i con
        distribuci n Beta( %s , 1)",n,alfa), col=(c("lightgreen","lightblue")), names=c
        ("Theta1","Theta2"))
```

```
    legend('topright',c('Teta 1','Teta 2'), fill = rgb(0:0,1:0,0:1,0.5), bty = 'n',
        border = NA)
    par(ask=parar)
  }
}

#-------------------------------------------------------------------------
#  Funcion que grafica un boxplot comprativo de las distintas
#  muestras para un mismo estimador
#-------------------------------------------------------------------------
#------------------------- PARAMETROS ------------------------------------
#  id = identifica si es el etimador 1 o 2
#  n = tamao muestra
#  alfa= parametro alfa de la distribucion beta
#  muestra=la cantidad de estimadore que se quiere
#-------------------------------------------------------------------------

crearBoxplotEstimador=function(id,muestra,alfa)
{
  muestra1=darMuestraEstimador(id,muestra,alfa,20)
  muestra2=darMuestraEstimador(id,muestra,alfa,50)
  muestra3=darMuestraEstimador(id,muestra,alfa,100)
  muestra4=darMuestraEstimador(id,muestra,alfa,200)
  muestra5=darMuestraEstimador(id,muestra,alfa,500)

#-------------------- Primero Guardo el plot -----------------------------

  if(guardar=="YES")
  {
    a=sprintf("/boxplotcomparadoTeta%iyBeta(%s,1).jpg",id,alfa)
    jpeg(paste("E:/imagenes",a,sep=""))
    boxplot(muestra1,muestra2,muestra3,muestra4,muestra5, main=sprintf("Comparacin
        BoxPlot Teta %i para Beta( %s , 1)",id,alfa), col=rgb(0,1,0,0.5), names=c("20",
        "50","100","200","500"), xlab="Tamao de la Muestra Beta")
    dev.off()
  }

#------------------- luego se visualiza ----------------------------------

  if(parar)
  {
    boxplot(muestra1,muestra2,muestra3,muestra4,muestra5, main=sprintf("Comparacin
        BoxPlot Teta %i para Beta( %s , 1)",id,alfa), col=rgb(0,1,0,0.5), names=c("20",
        "50","100","200","500"), xlab="Tamao de la Muestra Beta")
    par(ask=parar)
  }
}

#-------------------------------------------------------------------------
#  Funcion que grafica el logaritmo de n vs el logaritmo del
#  error cuadratico medio
#-------------------------------------------------------------------------
#------------------------- PARAMETROS ------------------------------------
#  muestraLogTeta1 = la muestra de los logaritmos del ECM teta1
#  muestraLogTeta2 = la muestra de los logaritmos del ECM teta2
#  alfa= parametro alfa de la distribucion beta
#-------------------------------------------------------------------------

darGraficaLogaritmos=function(alfa, muestraLogTeta1, muestraLogTeta2)
{
  x=log(c(20,50,100,200,500))
  y1 = muestraLogTeta1
  y2 = muestraLogTeta2

#------------------- Primero Guardo el plot ------------------------------
```

```
  if(guardar=="YES")
  {
    a=sprintf("/graficalogBeta(%s,1).jpg",alfa)
    jpeg(paste("E:/imagenes",a,sep=""))
    plot(x,y1, type="l", col="darkblue", lwd="2", xlab="ln(n)",ylab="ln(teta)")
    lines(x,y2, type="l", col="darkgreen", lwd="2")
    legend("top", col = c("darkblue", "darkgreen"), lty = 1, lwd="2", legend = c("Teta
        1", "Teta 2"))
    title(main=sprintf("Logaritmo de n vs. Logaritmo del ECM de Beta( %s , 1)",alfa))
    dev.off()
  }

#------------------- luego se visualiza ---------------------------

  if(parar)
  {
    plot(x,y1, type="l", col="darkblue", lwd="2", xlab="Logaritmo de n",ylab="
        Logaritmo del estimador Teta")
    lines(x,y2, type="l", col="darkgreen", lwd="2")
    legend("top", col = c("darkblue", "darkgreen"), lty = 1, lwd="2",legend = c("Teta
        1", "Teta 2"))
    title(main=sprintf("Logaritmo de n vs. Logaritmo del ECM de Beta( %s , 1)",alfa))
    par(ask=parar)
  }
}

#---------------------------------------------------------------------
#  Funcion que crea las funciones aleatorias y grafica los
#  boxplots para luego exportar a tablas de excel los valores
#  de la eficiencia, sesgo y ECM
#---------------------------------------------------------------------
#------------------------- PARAMETROS -------------------------------
#  muestra = la cantidad de estimadores que se quiere
#---------------------------------------------------------------------

correrProyecto2=function(muestra)
{
  tam=c(20,50,100,200,500)
  alfa=c(0.3,1,3,10)
  eficiencia=c()
  sesgoTeta1=c()
  sesgoTeta2=c()
  ECMTeta1=c()
  ECMTeta2=c()

  for(j in 1:4)
  {
    logaECMTeta1=c()
    logaECMTeta2=c()

    for(i in 1:5)
    {
      teta1 = darMuestraEstimador(1,muestra,alfa[j],tam[i])
      teta2 = darMuestraEstimador(2,muestra,alfa[j],tam[i])

      crearBoxPlot(tam[i],alfa[j],teta1 ,teta2 )

      eficiencia = c(eficiencia,darEficiencia(teta1,teta2,alfa[j]))

      sesgoTeta1=c(sesgoTeta1,darSesgo(teta1, alfa[j]))
      sesgoTeta2=c(sesgoTeta2,darSesgo(teta2, alfa[j]))

      ECMTeta1=c(ECMTeta1,darECM(teta1, alfa[j]))
      ECMTeta2=c(ECMTeta2,darECM(teta2, alfa[j]))
```

```
        logaECMTeta1=c(logaECMTeta1, log(darECM(teta1, alfa[j])))
        logaECMTeta2=c(logaECMTeta2, log(darECM(teta2, alfa[j])))
      }
      crearBoxplotEstimador(1,muestra,alfa[j])
      crearBoxplotEstimador(2,muestra,alfa[j])

      darGraficaLogaritmos(alfa[j], logaECMTeta1, logaECMTeta2)
    }

  #---------------- Ahora se exportan las Tablas ----------------
  #------------------------ Tablas Sesgo --------------------------

    sesgoTeta1= c(20,50,100,200,500,sesgoTeta1)
    matriz = matrix(sesgoTeta1,ncol=5)
    write.table(matriz,file=paste("E:/imagenes/tabla sesgo teta1.csv",sep=""),sep=
        separador,col.names=c("Sesgo Teta 1",0.3,1,3,10), row.names=FALSE)

    sesgoTeta2= c(20,50,100,200,500,sesgoTeta2)
    matriz = matrix(sesgoTeta2,ncol=5)
    write.table(matriz,file=paste("E:/imagenes/tabla sesgo teta2.csv",sep=""),sep=
        separador, col.names=c("Sesgo Teta 2",0.3,1,3,10), row.names=FALSE)

  #------------------------ Tablas ECM ----------------------------

    ECMTeta1= c(20,50,100,200,500,ECMTeta1)
    matriz = matrix(ECMTeta1,ncol=5)
    write.table(matriz,file=paste("E:/imagenes/tabla ECM teta1.csv",sep=""),sep=
        separador,col.names=c("ECM Teta 1",0.3,1,3,10), row.names=FALSE)

    ECMTeta2= c(20,50,100,200,500,ECMTeta2)
    matriz = matrix(ECMTeta2,ncol=5)
    write.table(matriz,file=paste("E:/imagenes/tabla ECM teta2.csv",sep=""),sep=
        separador,col.names=c("ECM Teta 2",0.3,1,3,10), row.names=FALSE)

  #------------------------ Tablas Eficiencia ----------------------

    eficiencia = c(20,50,100,200,500,eficiencia)
    matriz = matrix(eficiencia,ncol=5)
    write.table(matriz,file=paste("E:/imagenes/tabla Eficiencias teta.csv",sep=""),sep=
        separador,col.names=c("Eficiencias",0.3,1,3,10), row.names=FALSE)
  }

  #-------------- EJECUCION FINAL----------------

  correrProyecto2(500)
```