eman ta zabal zazu

Universidad del País Vasco
Euskal Herriko Unibertsitatea

Seminar and Activity 2

1 Synopsis.

What this activity is about. In the previous seminar you saw how to estimate things using the Monte Carlo method. You now have to use this technique again in conjunction with the theory to examine relative efficiency of moment and maximum likelihood estimators.

You will see on a simple example that, as expected from the theory, MLE estimators outperform other estimators, including moment estimators, at least for large sample size. For small sample sizes, this may not be the case.

What you need. You need to be fully acquainted with the content of previous seminars and practice assignment. You will also need access to a computer equiped with R.

2 Background

Assume data with a histogram like that in Figure 1. If we were to fit a distribution to such data, a good initial choice might be a $\gamma(a, r)$ which we know can have a similar shape for well chosen a and r. Just to keep our example simple, we will consider a subset of the gamma family: $\gamma(\frac{1}{2}, r)$, i.e. the first parameter is fixed at $a = \frac{1}{2}$. (For integer values of r, $\gamma(\frac{1}{2}, r)$ is the χ^2_{2r} degrees of freedom.) With this simplification, we are concerned with only one parameter to estimate: r.

2.1 Moment estimator of r

Since (check the theory) the mean of a $\gamma(\frac{1}{2}, r)$ is $2r$, the moment estimator of r is quite simple:

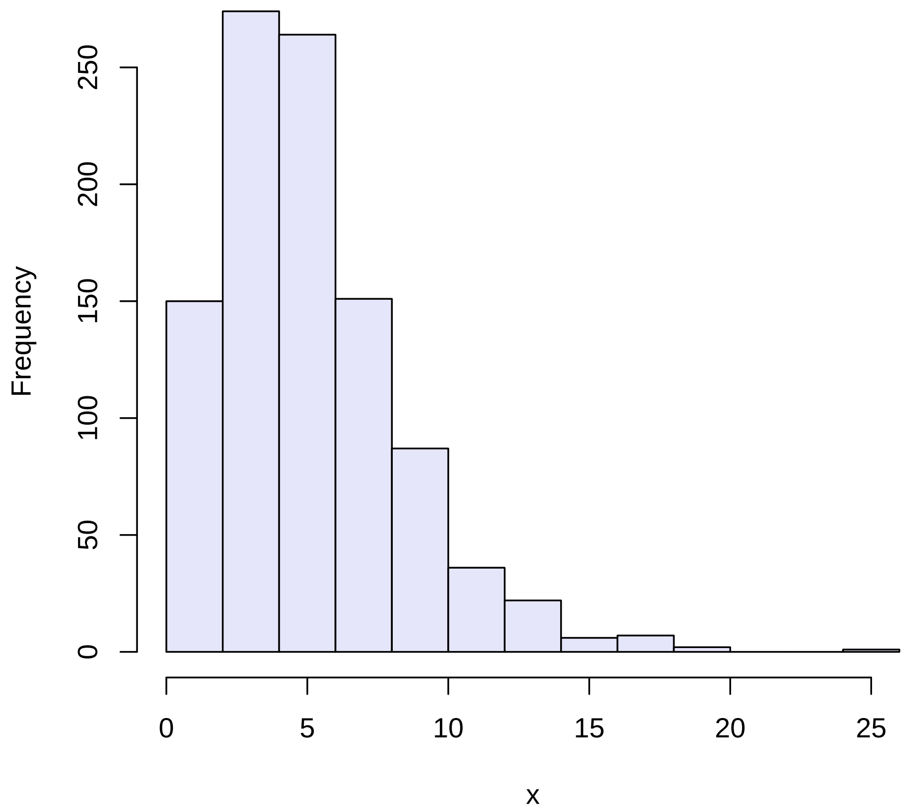
$$\hat{r}_{\mathrm{M}} = \overline{X}/2.$$

2.2 Maximum likelihood estimator of r

This is much more complex, but do not be intimidated by the formidable-looking formulae below! Conceptually, it is quite simple. Remember that the density of the distribution $\gamma(\frac{1}{2}, r)$ is:

$$f(x) = \frac{(\frac{1}{2})^r}{\Gamma(r)} x^{r-1} e^{-x/2}$$

Figure 1: Histogram of data to fit



Hence, the likelihood associated with a sample $(x_1, \ldots, x_n)$ is:

$$L(r; x_1, \ldots, x_n) = \frac{(\frac{1}{2})^{nr}}{(\Gamma(r))^n} \left[\prod_{i=1}^{n} x_i^{(r-1)} \right] e^{-\sum_{i=1}^{n} x_i/2}, \tag{1}$$

and the log-likelihood:

$$\ell(r; x_1, \ldots, x_n) = nr \log\left(\frac{1}{2}\right) - n \log \Gamma(r) + \sum_{i=1}^{n} (r-1) \log(x_i) - \frac{1}{2} \sum_{i=1}^{n} x_i. \tag{2}$$

In order to find the maximum likelihood estimator of the single parameter r characterizing the distribution, we would take the derivative of (2) with respect to r and equate to zero:

$$\frac{\partial \ell(r; \vec{x})}{\partial r} = n \log\left(\frac{1}{2}\right) - n \frac{\partial \log \Gamma(r)}{\partial r} + \sum_{i=1}^{n} \log(x_i) \tag{3}$$

The derivative

$$\frac{\partial \log \Gamma(r)}{\partial r} = \frac{\Gamma'(r)}{\Gamma(r)} \stackrel{\text{def}}{=} \psi(r)$$

cannot be obtained in closed form, but is a common function that arises quite frequently, the so-called ψ or digamma function[1]. We can write the expression (3) as:

$$n \log\left(\frac{1}{2}\right) - n\psi(r) + \sum_{i=1}^{n} \log(x_i) \tag{4}$$

For a sample `X` and a given r that expression is quite easy to compute in R. For instance, let

```
> X <- c(2, 3.4, 5.2, 1.3, 4.7)
> r <- 3
> n <- length(X)
```

Then (4) can be computed as follows:

```
> ll <- n * log(0.5) - n * digamma(r) +  sum(log(X))
> ll
```

```
[1] -2.70415
```

We will have to compute that expression over an over again, so we can define it as a function:

```
> dloglik <- function(r, x) {
      n <- length(x)
      ll <- n * log(0.5) - n * digamma(r) +  sum(log(x))
      return(ll)
      }
```

We can test that `dloglik` works:

```
> dloglik(r=3, x=X)
```

```
[1] -2.70415
```

What `dloglik` returns us is the derivative of the log likelihood, that we need to equate to zero to obtain the MLE estimator. We know that $r > 0$. Given the values in `X` we can guess that $\hat{r}_{\text{MLE}}$ will be no larger than, say, 10. (Remember that $2r$ is the mean of the distribution, so the values in `X`, with an average of 3.32, suggest an r of about half that value; 10 is a very conservative upper bound for r.) To find the value of r which makes the derivative equal to zero, we can use function `uniroot`. Function `uniroot`, when given a function (here `dloglik`) and an interval, finds a root of said function in that interval[2].

```
> uniroot(dloglik, lower=0.1, upper=10, x=X)
```

[1]Entirely unrelated to the characteristic function, even though the notation is similar. The values of $\psi(r)$ can be numerically approximated for different values of r; see [1], § 6.3, or [2], p. 220. We need not worry about its computation, because we can use function `digamma` in R for that.

[2]It is required that the value of the function at the extremes of the interval be of opposite signs and that the function be continuous; this ensures that it crosses zero at some spot.

```
$root
[1] 1.93793

$f.root
[1] 2.549052e-07

$iter
[1] 8

$init.it
[1] NA

$estim.prec
[1] 6.103516e-05
```

Notice that `uniroot` returns several pieces of information in a list. Component `root` contains the value of r which makes null the function `dloglik`, i.e. the MLE estimate. `iter` is the number of iterations, `f.root` the value of the function at `root` (which should be within tolerance of zero if the function ended properly).

The code above found a root (and hence $\hat{r}_{\text{MLE}}$) equal to 1.93795. Compare the value we would obtain using the method of moments estimator:

```
> mean(X) / 2

[1] 1.66
```

2.3 Cramér-Rao lower bound

The derivative of the likelihood associated to a single observation is (see (3) above):

$$\frac{\partial \ell(r;X)}{\partial r} = \log\left(\frac{1}{2}\right) - \frac{\partial \log \Gamma(r)}{\partial r} + \log(X) \tag{5}$$

and the second derivative is

$$\frac{\partial^2 \ell(r;X)}{\partial r^2} = -\frac{\partial^2 \log \Gamma(r)}{\partial r^2}. \tag{6}$$

The second derivarive of $\log \Gamma(r)$ is known as the trigamma function and can be computed in R using function `trigamma`. For given r, the Cramér-Rao lower bound for an unbiased estimator based in n observations is:

$$\text{CR}(r) = \frac{1}{nE\left[-\frac{\partial^2 \ell(r;X)}{\partial r^2}\right]} = \frac{1}{n\left[\frac{\partial^2 \log \Gamma(r)}{\partial r^2}\right]} \tag{7}$$

which can be computed easily in R by:

```
>  1 / ( n * trigamma(r) )
```

2.4 Relative performance of both estimators

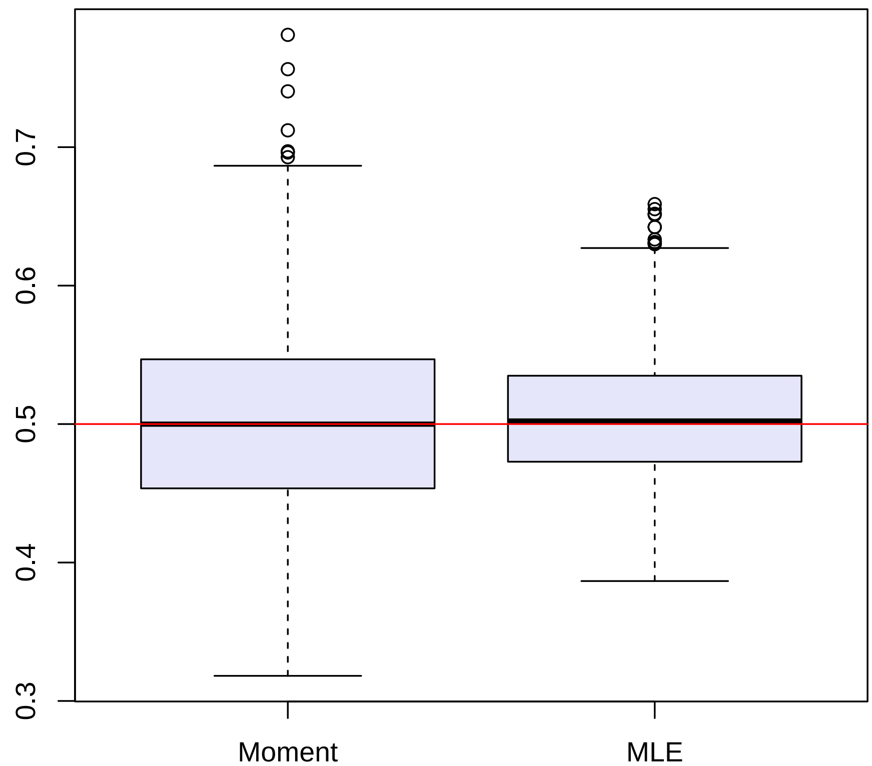
The properties of the maximum likelihood estimator, in particular, are not obvious; but we can simulate, taking repeated samples from a $\gamma(\frac{1}{2}, r)$ distribution with known r and see how the MLE and other estimators perform. For instance, to simulate $N = 1000$ estimation problems of r from a $\gamma(\frac{1}{2}, r = 0.5)$ distribution with samples of size $n = 100$ we could use the following code:

```
> N  <- 1000 ; n <- 100 ; r <- 0.5       # Values controlling the simulation
> rM <- rMLE <- rep(0,N)                 # Vectors to keep estimates
> for (i in 1:N) {                       # Simulation loop
      X <- rchisq(n, df=2*r)             # Generate random sample
      rM[i] <- mean(X) / 2               # Moment estimator
      rMLE[i]  <- uniroot(dloglik,       # MLE estimator
                       lower=0.1,
                       upper=10,
                       x=X)$root
      }
```

Notice that `rchisq(n, df=2*r)` generates values from a $\gamma(\frac{1}{2}, \frac{2r}{2})$; it does not matter if the degrees of freedom are not an integer value. We could use also `rgamma`, but then we would have to wrestle with a more complex syntax, passing explicitly $a = \frac{1}{2}$, etc.

On exit of this loop, we can draw histograms of the estimates, compute average bias, variance, whatever. For instance, we can compare both estimators using boxplots:

```
> boxplot(list(Moment=rM,MLE=rMLE), col="lavender")
> abline(h=0.5, col="red")
```



It is apparent the better performance of the MLE estimator. The moment estimator happens to be unbiased[3] and the MLE estimator is not far from unbiased. Let's find an approximation of the bias of both estimators:

```
> mean(rM) - r

[1] 0.002334014

> mean(rMLE) - r

[1] 0.005626007
```

Let's check the MSE:

```
> (MSE.moment <- mean( (rM - r)^2 ) )

[1] 0.004986294

> (MSE.mle    <- mean( (rMLE - r)^2 ) )
```

[3]You will have to show this in the upcoming Activity 2.

```
[1] 0.002130636
```

Although the MLE is slightly worse in terms of bias, the mean square error is far better, because of its lower variance. The ratio of MSE is:

```
> MSE.mle / MSE.moment
```

```
[1] 0.4272985
```

in favour of the MLE. This ratio, assuming both estimators were unbiased, is an approximation of the efficiency of the moment estimator relative to the MLE estimator.

3 Problems Activity 2

Consider for all problems data generated by a $\gamma(\frac{1}{2}, r = 0.5)$, unless otherwise stated. Generating your data from a known distribution affords you the luxury of comparing the estimates to the "true" value of the parameter, something you cannot do in practice.

1. Check (with pen and paper) that $\hat{r}_{\mathrm{M}}$ is unbiased.

   Respuesta: Sabemos que una $\gamma(\frac{1}{2}, r)$ tiene valor medio $r/a = 2r$. Por tanto,

   $$E[\hat{r}_{\mathrm{M}}] = E[\overline{X}/2] = 2r/2 = r.$$

   Por tanto, $\hat{r}_{\mathrm{M}}$ es efectivamente insesgado para r.

2. Compute the variance of $\hat{r}_{\mathrm{M}}$ based on n observations: it depends on r. What is the numerical value of the variance when $r = 0.5$?

   Respuesta: La varianza de una $\gamma(\frac{1}{2}, r)$ es $r/(\frac{1}{2})^2 = 4r$. Por tanto,

   $$\mathrm{Var}(\hat{r}_{\mathrm{M}}) = \mathrm{Var}(\overline{X}/2) = \frac{1}{4n}(4r) = r/n$$

3. Is $\hat{r}_{\mathrm{M}}$ consistent?

   Respuesta: Claramente sí, ya que es insesgado y su varianza tiende a cero.

4. Is $\hat{r}_{\mathrm{M}}$ sufficient?

   Respuesta: Mirando la expresión de la verosimilitud (1) vemos que puede factorizarse así:

   $$\frac{(\frac{1}{2})^{nr}}{(\Gamma(r))^n}\left[\prod_{i=1}^{n} x_i^{(r-1)}\right] \times e^{-\sum_{i=1}^{n} x_i/2}; \tag{8}$$

   la función de las observaciones que aparece en compañía de r es $\prod_{1=1}^{n} x_i$. Cualquier función uno-uno de $\prod_{1=1}^{n} x_i$ (como $\sum_{1=1}^{n} \log(x_i)$) es también suficiente, pero $\overline{X}$ no lo es. Por tanto no es un estadístico suficiente, e incurre en potencial ineficiencia.

5. What is the numerical value when $r = 0.5$ of the Cramér-Rao lower bound for an unbiased estimator based on n observations? What is the efficiency of $\hat{r}_{\mathrm{M}}$ for such value of r?

   Respuesta: La cota de Cramér-Rao se obtuvo en (7). La eficiencia de $\hat{r}_{\mathrm{M}}$ es:

   $$\mathrm{Eff}(\hat{r}_{\mathrm{M}}) = \frac{\mathrm{CR}(\hat{r})}{\mathrm{Var}(\hat{r}_{\mathrm{M}})} = \frac{n}{nr\left[\frac{\partial^2 \log\Gamma(r)}{\partial r^2}\right]} = \frac{1}{r\left[\frac{\partial^2 \log\Gamma(r)}{\partial r^2}\right]} \tag{9}$$

   We can easily calculate its value in R as `1 / ( r * trigamma(r) )`. When $r \approx 0$ esta eficiencia es muy baja. Cuando $r \to \infty$ puede verse que esta eficiencia tiende a 1. When $r = 0.5$ the efficiency is 0.405. This value is not too different from what we found by simulation in Seminar 2. On the one hand, both estimators are at least asymptotically unbiased, so the MSE≈Var; on the other, the MLE approaches at least asymptotically the Cramér-Rao lower bound. Therefore,

   $$\mathrm{Eff}(\hat{r}_{\mathrm{M}}) = \frac{\mathrm{CR}(\hat{r})}{\mathrm{Var}(\hat{r}_{\mathrm{M}})} \approx \frac{\mathrm{Var}(\hat{r}_{\mathrm{MLE}})}{\mathrm{Var}(\hat{r}_{\mathrm{M}})} \approx \frac{\mathrm{MSE}(\hat{r}_{\mathrm{MLE}})}{\mathrm{MSE}(\hat{r}_{\mathrm{M}})}$$

   which is what was estimated in Seminar 2.

6. Is $\hat{r}_{\mathrm{M}}$ efficient?

   Respuesta: Consecuencia de la respuesta anterior, no; y la ineficiencia puede ser muy grande cuando r es pequeño.

7. For $\hat{r}_{\mathrm{MLE}}$ answering to the same questions is not nearly as easy; but you can always resort to simulation, and you know that $\hat{r}_{\mathrm{MLE}}$ will be (at least asymptotically) unbiased and (at least asymptotically) will approach the Cramér-Rao lower bound, so $\mathrm{MSE}(\hat{r}_{\mathrm{MLE}}) \approx \mathrm{CR}(r)$.

   (a) For each $r = 0.1, 0.5, 1, 3, 5$ and 8 generate $N = 1000$ samples of size $n = 100$ from a $\gamma(\frac{1}{2}, r)$ and compute each time $\hat{r}_{\mathrm{M}}$ and $\hat{r}_{\mathrm{MLE}}$.
   Respuesta: Ejemplo de código para resolver esta cuestión en el Apéndice.

   (b) Compute the average bias of both estimators for each r.
   Respuesta: Ejemplo de código para resolver esta cuestión en el Apéndice.

   (c) Compute the average mean square error (MSE) of both estimators for each r.
   Respuesta: Ejemplo de código para resolver esta cuestión en el Apéndice.

   (d) Your know that $\hat{r}_{\mathrm{M}}$ is unbiased, and $\hat{r}_{\mathrm{MLE}}$ appears to be close to unbiased: therefore, the MSE is equal or close to the variance. Check that the MSE of $\hat{r}_{\mathrm{MLE}}$ is close to the Cramér-Rao lower bound.
   Respuesta: Ejemplo de código para resolver esta cuestión en el Apéndice.

   (e) Compute the efficiency of $\hat{r}_{\mathrm{M}}$ and $\hat{r}_{\mathrm{MLE}}$ for the different r.
   Respuesta: Ejemplo de código para resolver esta cuestión en el Apéndice.

,

References

[1] M. Abramovitz and I. Stegun, editors. *Handbook of Mathematical Functions.* Dover Pub., 1965.

[2] A. Jeffrey. *Handbook of Mathematical Formulas and Integrals.* Academic Press, 1995.

A Apéndice

Cuestión 7.a

```
> N  <- 1000 ; n <- 100                 # Valores controlando la simulación
> r  <- c(0.1, 0.5, 1, 3, 5, 8)
> rM <- rMLE <- matrix(0,N,length(r))   # Matrices para guardar las estimaciones
> colnames(rM) <-
      colnames(rMLE) <- as.character(r) # Rotulamos las columnas
> for (j in seq_along(r)) {
  for (i in 1:N) {
      X <- rchisq(n, df=2*r[j])         # Generate random sample
      rM[i,j] <- mean(X) / 2            # Moment estimator
      rMLE[i,j]  <- uniroot(dloglik,    # MLE estimator
                       lower=0.05,
                       upper=10,
                       x=X)$root
      }
  }
```

Cuestión 7.b

```
> colMeans(rM) - r
```

```
          0.1           0.5             1             3             5
-0.0002648742 -0.0017101150 -0.0012368091 -0.0055953067  0.0001949151
            8
 0.0015349666
```

```
> colMeans(rMLE) - r
```

```
         0.1          0.5            1            3            5
 0.001027628  0.001156924  0.003715193 -0.001699272  0.004543251
           8
 0.008398649
```

Cuestión 7.c

```
> true.r <- matrix(r,N,length(r), byrow=TRUE)
> (MSE.moment <- colMeans( (rM - true.r)^2 ) )
```

```
         0.1          0.5            1            3            5
0.0009887945 0.0050587014 0.0106505024 0.0299038778 0.0535076445
           8
0.0791663883
```

```
> (MSE.mle    <- colMeans( (rMLE - true.r)^2 ) )
```

```
         0.1          0.5            1            3            5
0.0001009212 0.0020433079 0.0062252364 0.0250890912 0.0474365899
           8
0.0737341100
```

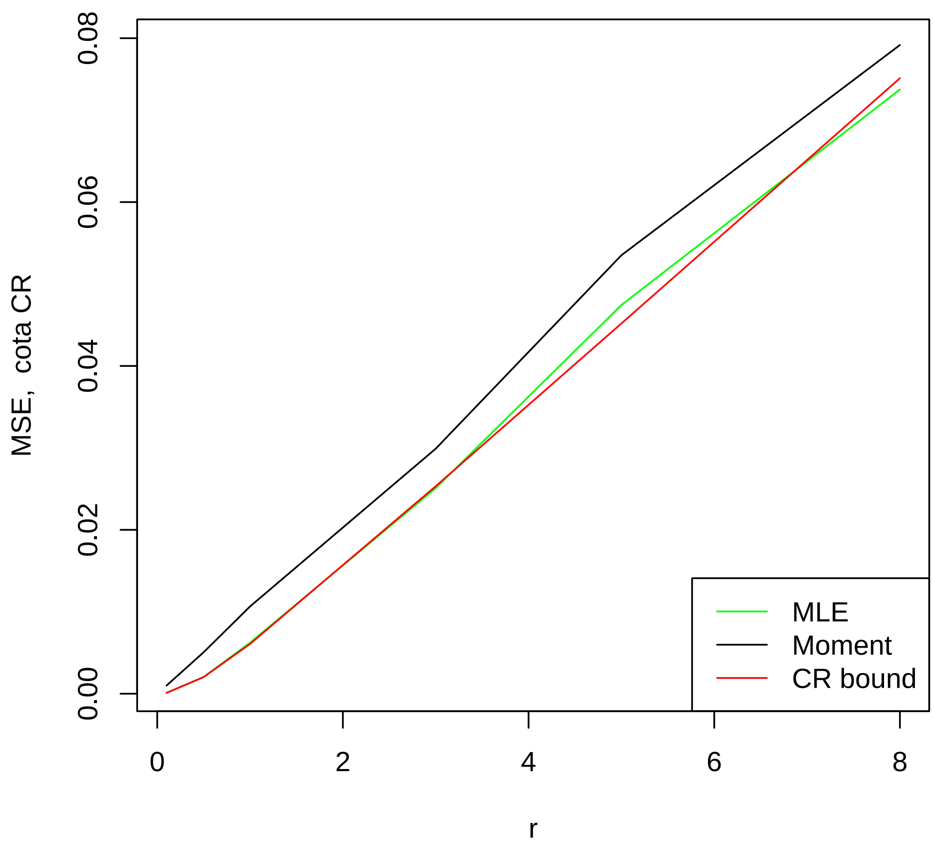
Cuestión 7.d Tenemos primero que calcular la cota de Cramér-Rao para los diferentes valores de `r` y un tamaño de muestra n, cosa que sabemos como hacer:

```
> CR <-  1 / ( n * trigamma(r) )
> CR

[1] 9.858695e-05 2.026424e-03 6.079271e-03 2.532068e-02 4.518284e-02
[6] 7.511059e-02
```

Representemos ahora los valores de MSE estimados (iguales a la varianza, en el caso de $\hat{r}_{\mathrm{M}}$ y casi iguales en el caso de $\hat{r}_{\mathrm{MLE}}$) junto con los valores óptimos dados por CR:

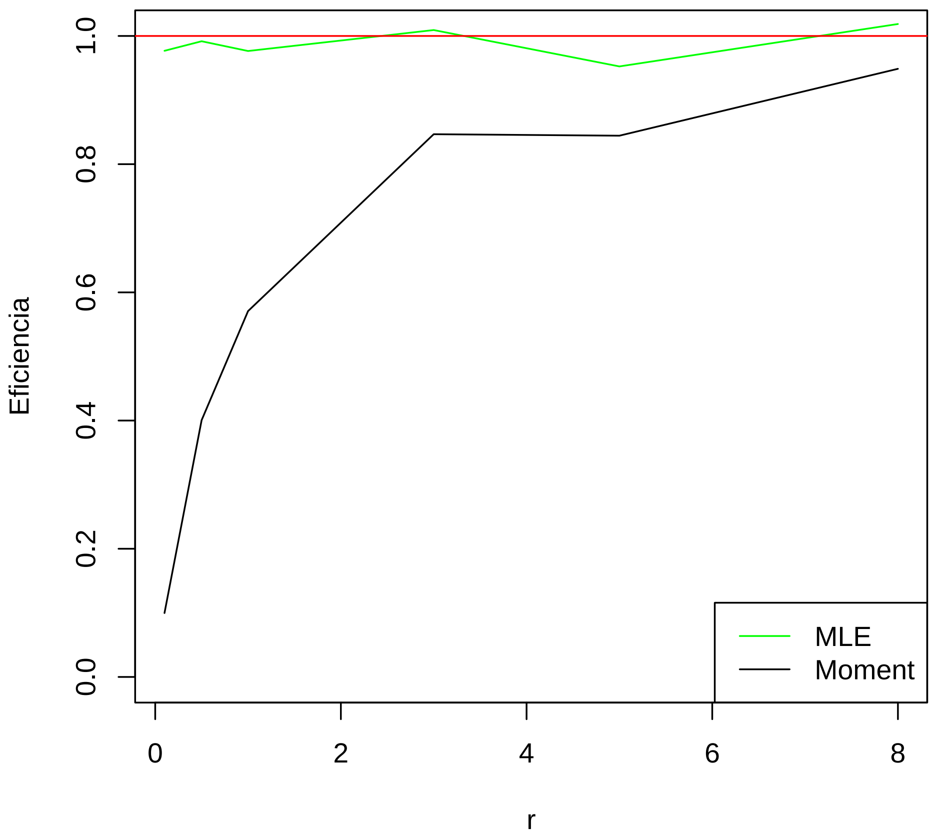
```
> plot(r,MSE.moment, col="black", type="l", ylab="MSE,  cota CR")
> lines(r, MSE.mle, col="green")
> lines(r, CR, col="red")
> legend("bottomrigh", legend=c("MLE", "Moment","CR bound"), col=c("green", "black", "red"), 
```



Vemos que el estimador $\hat{r}_{\mathrm{MLE}}$ tiene MSE practicamente coincidente con la cota de Cramér-Rao salvo para los valores de r 4 y 5, mientras que el estimador $\hat{r}_{\mathrm{M}}$ se separa apreciablemente.

Es más ilustrativo representar las respectivas eficiencias estimadas en la simulación:

```
> plot(r, CR/MSE.moment, col="black", type="l", ylim=c(0,1.), ylab="Eficiencia")
> lines(r, CR/MSE.mle, col="green")
> abline(h=1, col="red")
> legend("bottomrigh", legend=c("MLE", "Moment"), col=c("green", "black"), lty=c(1,1))
```



Nótese que las eficiencias son **estimadas** en la simulación, y pueden por ello en ocasiones exceder ligeramente de 1 (lo que nunca sucede para un estimador regular). La eficiencia del estimador por el método de momentos, muy baja para r cercano a cero, se aproxima a la del estimador MLE cuando r se hace moderado. Si tuviéramos la certeza de encontrarnos ante una distribución con r no muy pequeño, habría poco estímulo para emplear $\hat{r}_{\text{MLE}}$, de cómputo mucho más complejo que el de $\hat{r}_{\text{M}}$.