# Propiedades asintóticas de los estimadores MCO

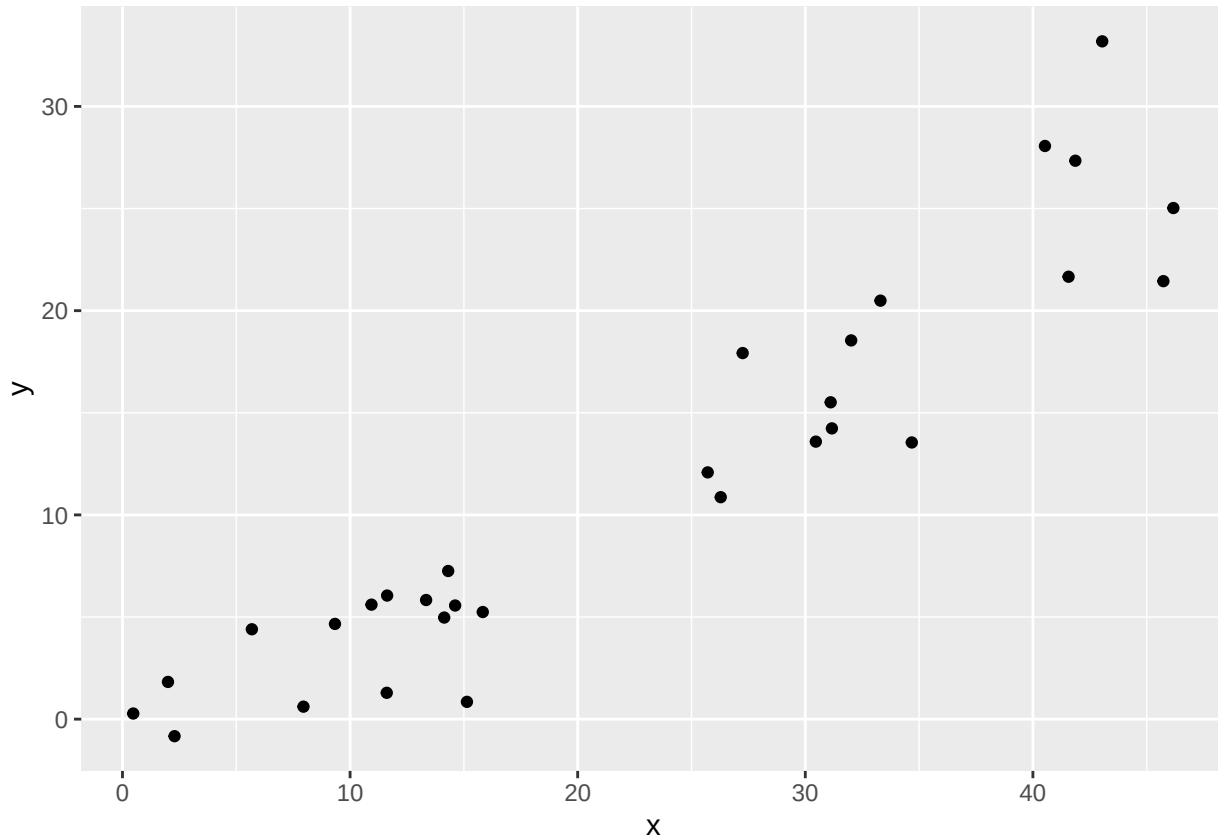
**Queremos llevar a cabo un experimento de Monte Carlo para ilustrar el comportamiento del estimador MCO cuando se aplica a muestras generadas bajo los supuestos clássicos. Considera el siguiente mecanismo de generación de datos:**

$$y_i = 2 + 0,5x_i + u_i$$
$$u_i/x \sim i.i.N(0, 16)$$
$$x_i \sim U[0, 50]$$

## Apartado A

*Genera una muestra de 30 observaciones de este mecanismo, utilizando como semilla 1234. Dibuja la muestra generada en un plot (x,y).*

```
# Medida muestral
n <-30

# Generamos los valores de la variable regresora, que serán los mismos en cada muestra.
set.seed(1234) # Semilla
x <- runif(n, min=0, max=50) # Distribución Uniforme

# Ajuste de la recta y generación de los errores.
y <- 2 + 0.5*x + rnorm(n, 0, 4) # Distribución normal

# Creamos un dataset con los valores x,y
data_xy = data.frame(x, y)

# Plot con la muestra generada
library(ggplot2)
ggplot(data=data_xy, aes(x=x, y=y)) +geom_point()
```




## Apartado B

*Estima el siguiente modelo de regresión con la muestra generada:*

$$y = \beta_0 + \beta_1 x + u$$

*¿Qué estimador obtienes para $\beta_0$? ¿Para $\beta_1$? Dado que conocemos el valor real de $\beta_0$ y $\beta_1$, puedes decir si la estimación anterior sobreestima o subestima estos parámetros.*

```
# Ajuste del modelo
model_b <- lm(y~x)
summary(model_b)
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.0246 -1.9912 -0.4526  2.0714  8.9169
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept) -2.56205    1.16382  -2.201   0.0361 *
## x            0.62332    0.04362  14.290 2.18e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.36 on 28 degrees of freedom
## Multiple R-squared:  0.8794, Adjusted R-squared:  0.8751
## F-statistic: 204.2 on 1 and 28 DF,  p-value: 2.18e-14
```

- $\hat{\beta}_0 = -2.562$ y $\beta_0 = 2$. Por lo tanto, la estimación subestima el parámetro.
- $\hat{\beta}_1 = -0.623$ y $\beta_1 = 0.5$. Por lo tanto, la estimación sobreestima el parámetro.

## Apartado C

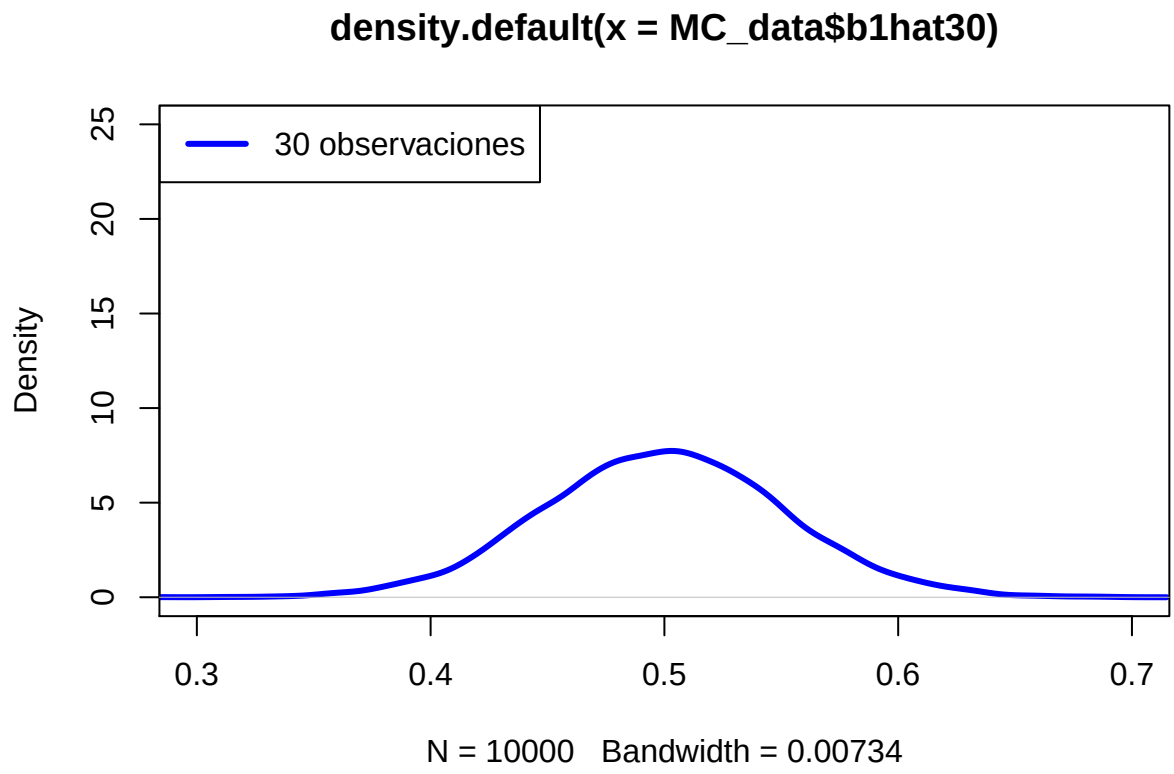
*Genera 10.000 observaciones de acuerdo con el proceso de generación de datos incluido arriba, utilizando los mismos valores de las observaciones de x para cada muestra. Con cada muestra estima la regresión del modelo indicado arriba y guarda las estimaciones del parámetro $\beta_1$. Usa el valor 1234 como semilla.*

```r
# Número de muestras que queremos generar
nsim <- 10000
b0hat <- numeric(nsim)
b1hat30 <- numeric(nsim)

# Generación de las muestras
for (i in 1:nsim){
  set.seed(1234+i)
  y <- 2+0.5*x+rnorm(n, 0, 4)
  Model <- lm(y~x)
  b0hat[i]<-Model$coefficients[1]
  b1hat30[i]<-Model$coefficients[2]
}

# Creamos un dataframe con las estimaciones de las 10000 muestras.
MC_data<-data.frame(b0hat,b1hat30)
```

## Apartado D

*Dibuja un histograma con las 10.000 estimaciones obtenidas de $\beta_1$. ¿Es este gráfico como esperabas?*

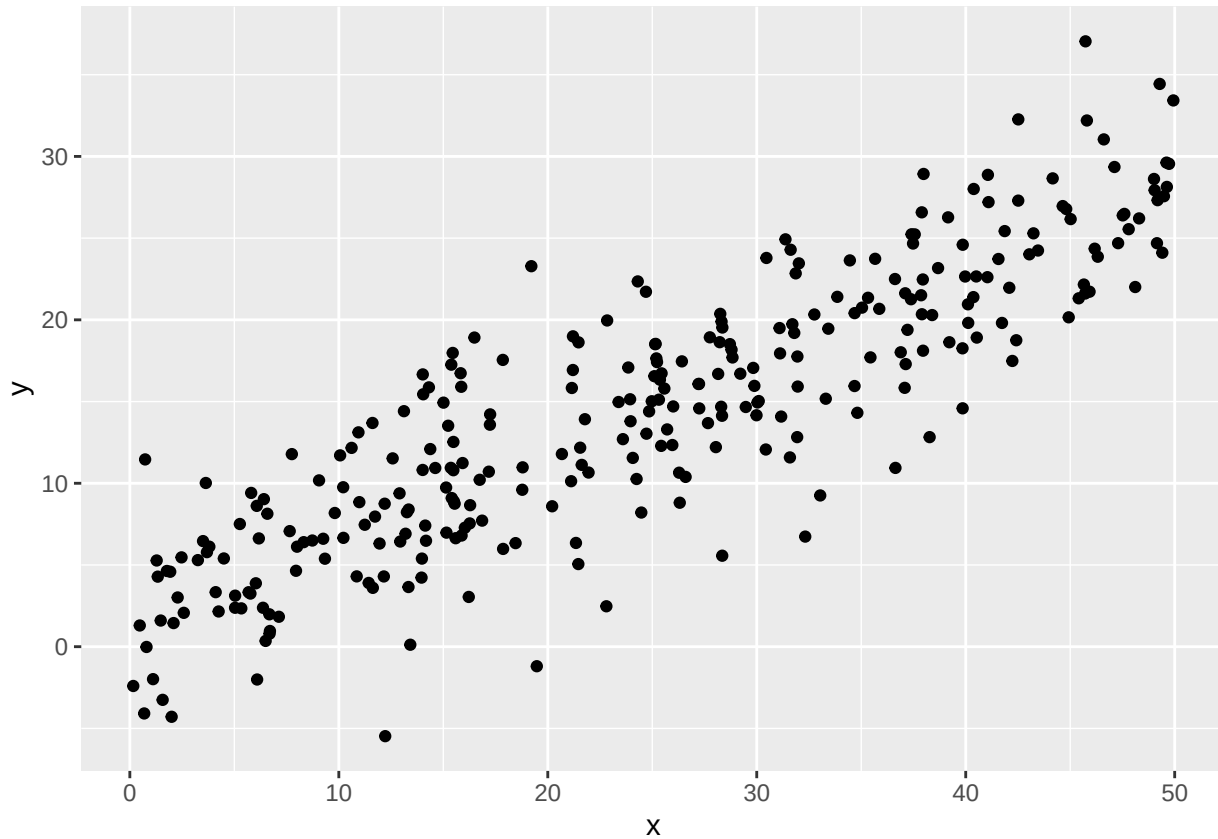
```r
# Densidad de las 10000 estimaciones de b1
plot(density(MC_data$b1hat30), lwd=3, col='blue',  ylim=c(0, 25), xlim = c(0.3, 0.7))
legend('topleft', legend=c('30 observaciones'),
       lwd=3, col=c('blue'))
```




### Apartado E

*Repite este ejercicio pero generando 10.000 muestras de 300 observacions cada una. Compara el histograma que has obtenido ahora (utilizando muestras de 300 observaciones cada una) com el que habias obtenido en el apartado d.*

```
# Medida muestral
n <-300

# Generamos los valores de la variable regresora, que serán los mismos en cada muestra.
set.seed(1234) # Semilla
x <- runif(n, min=0, max=50) # Distribución Uniforme

# Ajuste de la recta y generación de los errores.
y <- 2 + 0.5*x + rnorm(n, 0, 4) # Distribución normal

# Creamos un dataset con los valores x,y
data_xy = data.frame(x, y)

# Plot con la muestra generada
library(ggplot2)
ggplot(data=data_xy, aes(x=x, y=y)) +geom_point()
```




```r
# Ajuste del modelo
model_b <- lm(y~x)
summary(model_b)
```

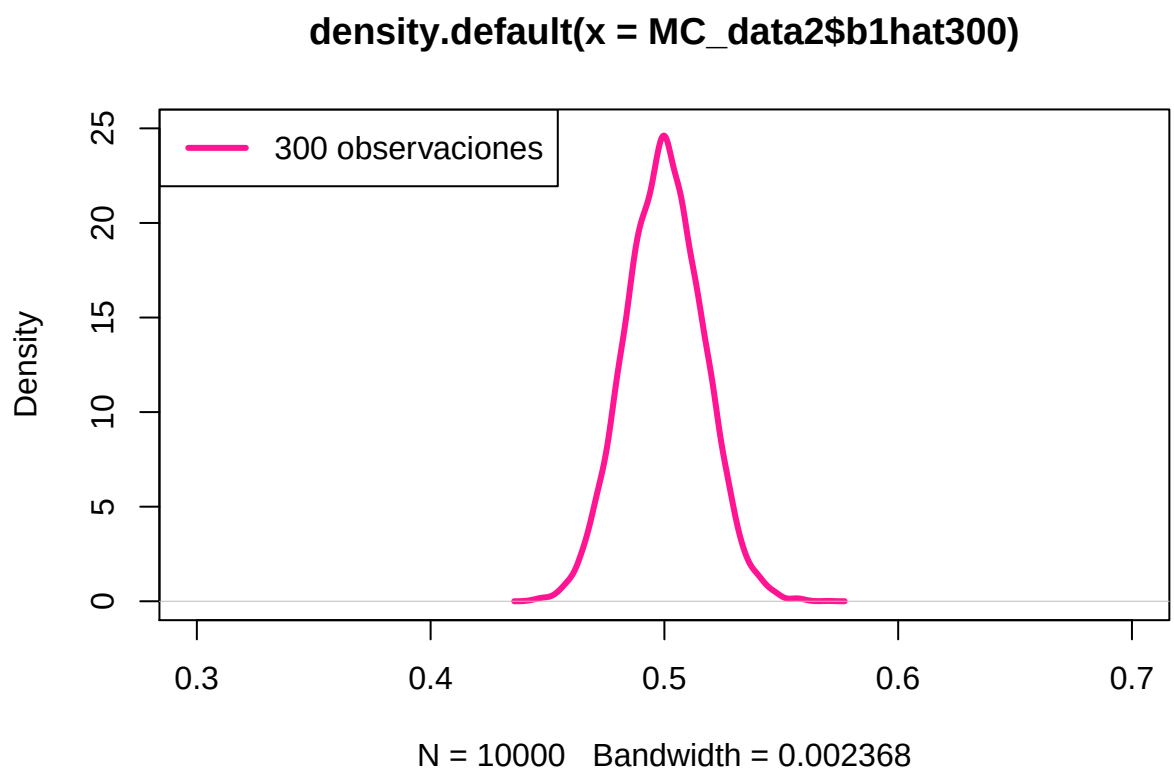
```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.5138  -2.5841   0.1014   2.5199  11.6161
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.70010    0.47876   3.551 0.000446 ***
## x            0.51878    0.01698  30.551  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.123 on 298 degrees of freedom
## Multiple R-squared:  0.758,  Adjusted R-squared:  0.7572
## F-statistic: 933.4 on 1 and 298 DF,  p-value: < 2.2e-16
```

- $\hat{\beta}_0 = 1.70$ y $\beta_0 = 2$. Por lo tanto, la estimación subestima el parámetro, pero mucho menos que considerando únicamente 30 observaciones.

- $\hat{\beta}_1 = -0.51$ y $\beta_1 = 0.5$. Por lo tanto, la estimación sobreestima ligeramente el parámetro.

```
# Número de muestras que queremos generar
nsim <- 10000
b0hat <- numeric(nsim)
b1hat300 <- numeric(nsim)

# Generación de las muestras
for (i in 1:nsim){
  set.seed(1234+i)
  y <- 2+0.5*x+rnorm(n, 0, 4)
  Model <- lm(y~x)
  b0hat[i]<-Model$coefficients[1]
  b1hat300[i]<-Model$coefficients[2]
}

# Creamos un dataframe con las estimaciones de las 10000 muestras.
MC_data2<-data.frame(b0hat,b1hat300)
```

```
# Densidad de las 10000 estimaciones de b1
plot(density(MC_data2$b1hat300), lwd=3, col='deeppink',  ylim=c(0, 25), xlim = c(0.3, 0.7))
legend('topleft', legend=c('300 observaciones'),
       lwd=3, col=c('deeppink'))
```



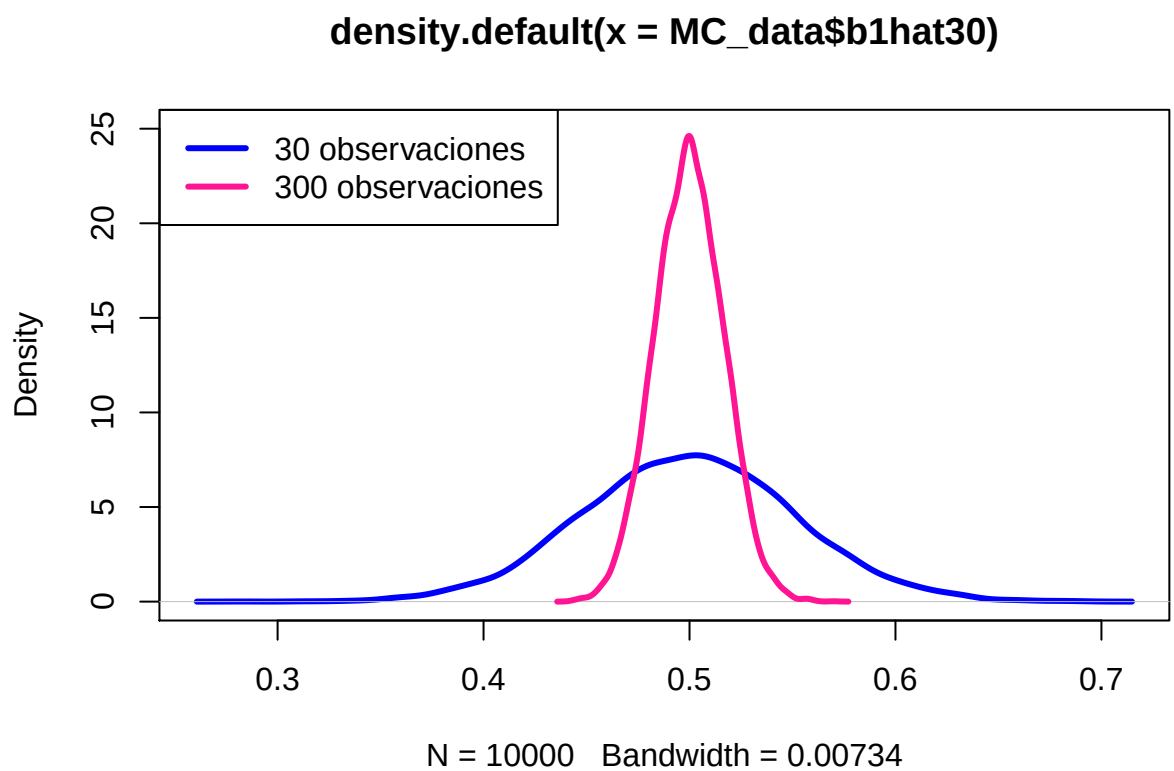
### Apartado F

*Compara los resultados obtenidos utilizando 30 observaciones y utilizando 300.*

- *1. Realiza un único gráfico que incluya las dos densidades obtenidas. Ha cambiado el la densidad de la forma que esperabas? Que propiedad del estimador MCO puedes ilustrar con esta comparación?*

- *2. En una tabla, indica cuáles son las estimaciones obtenidas de los parámetros para cada uno de los casos (30 y 300 observaciones), así como sus desviaciones estándar. ¿Qué conclusión extraes? Que propiedad del estimador MCO puedes ilustrar con esta comparación?*

```
# Superponemos las dos densidades para poder compararlas mejor
plot(density(MC_data$b1hat30), lwd=3, col='blue',  ylim=c(0, 25))
lines(density(MC_data2$b1hat300), lwd=3, col='deeppink')

legend('topleft', legend=c('30 observaciones', '300 observaciones'),
       lwd=3, col=c('blue', 'deeppink'))
```




| | n = 30 | n = 300 |
|---|---|---|
| $\hat{\beta}_0$ | -2.562 | 1.70 |
| $\hat{\beta}_1$ | 0.623 | 0.518 |
| Std. Error ($\hat{\beta}_0$) | 1.163 | 0.479 |
| Std. Error ($\hat{\beta}_1$) | 0.044 | 0.017 |

Teniendo en cuenta que los valores de los parámetros $\hat{\beta}_0$ y $\hat{\beta}_1$ son, respectivamente, 2 y 0.5...

La propiedad que ilustran estas comparaciones es la **consistencia de los estimadores MCO**, puesto que los estimadores $\hat{\beta}_0$ y $\hat{\beta}_1$ se aproximan al valor poblacional al pasar de 30 a 300 observaciones (aumento de la muestra). Debido a esta propiedad, se observa que, aunque el parámetro $\hat{\beta}_1$ siga en los dos casos una distribución normal, dicha distribución es mucho más preciaa (presenta menor varianza) si se consideran 300 observaciones en vez de 30.