Guia 17 Estimacion

Agustin Muñoz González

20/6/2020

Preparamos el entorno

```
rm(list=ls())
library(ggplot2)
library(tidyr)
library(gganimate)
```

1. Estimación bajo modelo uniforme $\mathcal{U}[0, \theta]$:

Sean $(X_i)_{i\geq 1}$ variables aleatorias independientes identicamente distribuídas, con distribución uniforme en el intervalo [0, θ]: X i$\sim$ U[0, θ]. Consideremos los siguientes estimadores de θ basados en una muestra X 1 , . . . , X n :

$$\widehat{\theta}(x_1, \ldots, x_n) = 2\overline{X}_n, \ \tilde{\theta}(x_1, \ldots, x_n) = \max(x_1, \ldots, x_n).$$

1. Implemente las funciones est1 y est2 que tengan por argumento un conjunto de datos (x 1 , . . . , x n) y devuelva el valor de la estimación $\widehat{\theta}(x_1, \ldots, x_n)$ y $\tilde{\theta}(x_1, \ldots, x_n)$, para los estimadores definidos en (1), respectivamente.

Resolución:

Notar que el estimador $2 * \overline{X_n}$ tiene sentido en tanto que $\overline{X_n} \xrightarrow[n\to\infty]{} \frac{\theta}{2}$, y $\max(X_1, \ldots, X_n)$ tiene sentido ya que θ es el largo de la mesa, entonces la máxima longitud que genere el mono va a ser lo mas parecido a θ.

```
est1=function(datos){
  2*mean(datos)
}
est2=function(datos){
  max(datos)
}
```

2. Calcule el valor de los estimadores est1 y est2 en los datos

1.17 1.75 0.28 2.56 2.36 0.36 1.82 0.24 1.17 1.86

Resolución:

Calculemos los dos estimadores de los 5 datos.

```
datos=c(1.17, 1.75, 0.28, 2.56, 2.36, 0.36, 1.82, 0.24, 1.17, 1.86)
est1(datos)
```

```
## [1] 2.714
```

```
est2(datos)
```

```
## [1] 2.56
```

3. Calcule el valor de los estimadores est1 y est2 en los datos

0.66 0.07 0.62 0.65 1.33 0.40 1.17 1.11 2.01 2.98

Resolución:

Calculemos los dos estimadores de los 5 datos.

```
datos=c(0.66, 0.07, 0.62, 0.65 ,1.33 ,0.40 ,1.17, 1.11, 2.01, 2.98)
est1(datos)
```

```
## [1] 2.2
```

```
est2(datos)
```

```
## [1] 2.98
```

Simulación 1. A lo largo de esta simulación generaremos variables con distribución uniforme en el intervalo [0, 3]. Es decir, trabajaremos con v.a. (X i) i.i.d., $X_i \sim \mathcal{U}[0, \theta]$ con $\theta = 3$.

4. Realice histogramas para emular la distribución de cada uno de los estimadores con n = 5, n = 30, n = 50, haciendo N rep = 1000 replicaciones. Comente las principales características que observa en los gráficos. Diría usted que la distribución de $\widehat{\theta}$ (est1) es aproximadamente normal? Diría usted que la distribución de $\tilde{\theta}$ (est2) es aproximadamente normal?

Resolución:

Defino fc simuladora de datos.

```
# simulo Nrep experimentos con n v.a. iiid
# y pongo cada resultado en una fila distinta
var.gen=function(n,Nrep,theta){
   tabla=c()
   for(i in (1:n)){
      tabla=cbind(tabla,runif(Nrep,0,theta))
   }
   data.frame(tabla)
}
```

Generamos los datos.

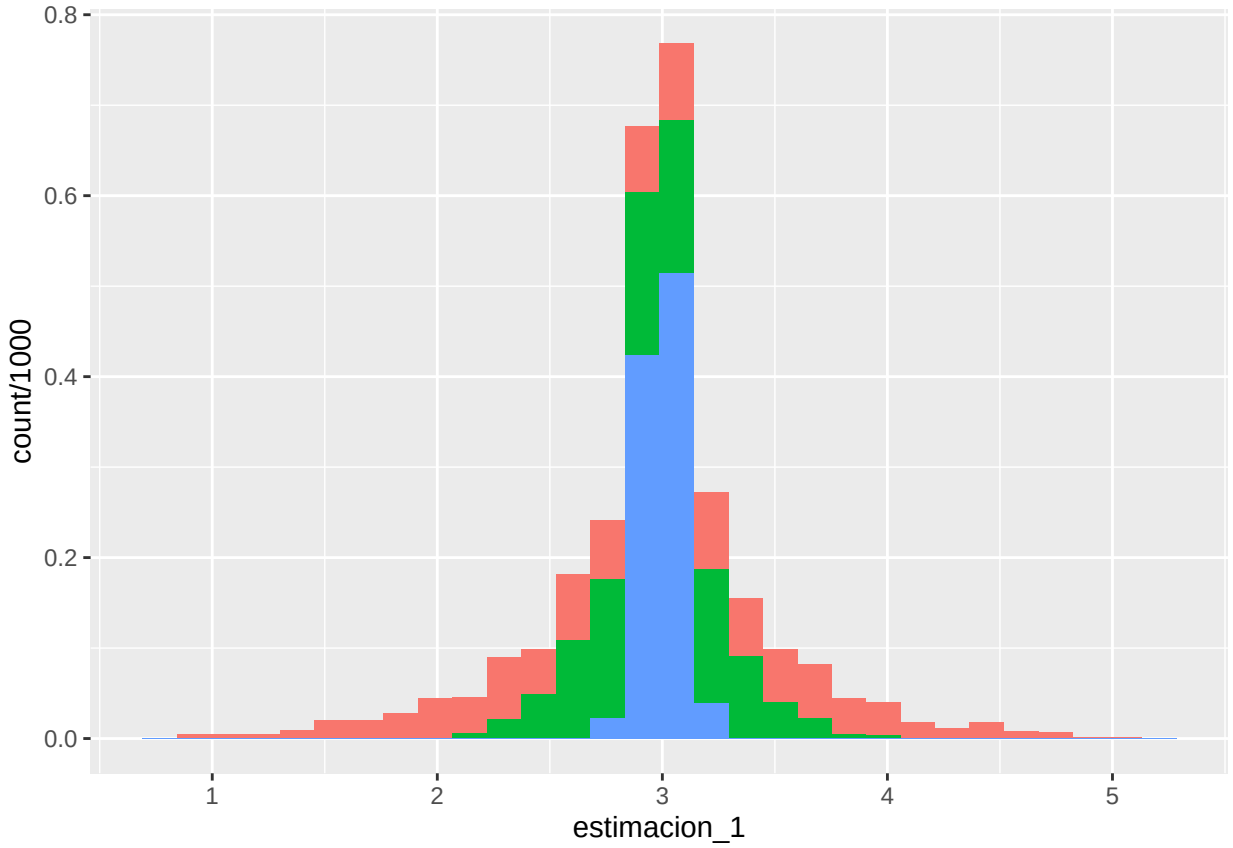
```
Nrep=1000
theta=3
enes=c(5,30,500)
estimacion_1=estimacion_2=matrix()
for(i in enes){
  # simulo Nrep experimentos con n v.a. iiid
  # y pongo cada resultado en una fila distinta
  muestra=var.gen(i,Nrep,theta)
  # le aplico est1 a cada fila
  # i.e. estimo con el primer estimador los datos
  # obtenidos en cada experimento (cada fila)
  aux_1=data.frame(apply(muestra,1,est1))
  names(aux_1)=paste('est1_',i)
  estimacion_1=cbind(estimacion_1,aux_1)
  # le aplico est2 a cada fila
  aux_2=data.frame(apply(muestra,1,est2))
```

```
  names(aux_2)=paste('est2_',i)
  estimacion_2=cbind(estimacion_2,aux_2)
  # estimaciones=cbind(estimaciones,estimacion_1,estimacion_2)
}
# le saco la columna 0 que tiene NAs
estimacion_1=estimacion_1[,2:dim(estimacion_1)[2]]
estimacion_2=estimacion_2[,2:dim(estimacion_2)[2]]
# verticalizamos los datos y les ponemos nombre
estimacion_1=stack(estimacion_1)[1]
names(estimacion_1)="estimacion_1"
estimacion_2=stack(estimacion_2)[1]
names(estimacion_2)="estimacion_2"
# le pongo una variable de filas para la transicion del gif
estimacion_1$filas=1:dim(estimacion_1)[1]
estimacion_2$filas=1:dim(estimacion_2)[1]
```

Ploteamos est1.

```
# nose como colorear usando filas, asi que agrego estados
estimacion_1$estado=as.factor(rep(c(5,30,500),rep(1000,3)))
estimacion_1 %>%
  ggplot(aes(fill=estado))+
  geom_histogram(aes(x=estimacion_1,y=stat(count)/1000))+
   scale_fill_discrete(name="Nº Variables",
                       breaks=1:3,
                       labels=paste("n=", enes))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Progresión est1 filtrando datos (filas) y que el plot entre y se vaya desvaneciendose.

```
# Como puse los datos verticalmente tenemos
# n=5 corresponden las filas 1:1000
# n=30 corresponden las filas 1000:2000
# n=500 corresponden las filas 2000:3000
anim_1=estimacion_1 %>%
  ggplot()+
  geom_histogram(aes(x=estimacion_1,y=stat(count)/1000),
                 color='blue',fill='grey')+
  transition_filter(
    transition_length = 3,
    filter_length = 1,
    1 <= filas & filas <= 1000 ,
    1000 <= filas & filas <= 2000,
    2000 <= filas
  )+
  ggtitle(
    'Progresión est1',
    subtitle = '{closest_expression}'
  ) +
  enter_fade()+
  exit_fade()
  # exit_fly(y_loc = 0)
# animate(anim_1,
#         width = 400, height = 400,
#         nframes = 480, fps = 24)
```

```
# anim_save("est1.gif", anim_1)
```

Progresión est1 filtrando estados y que el plot se mueva entre los distintos graficos.

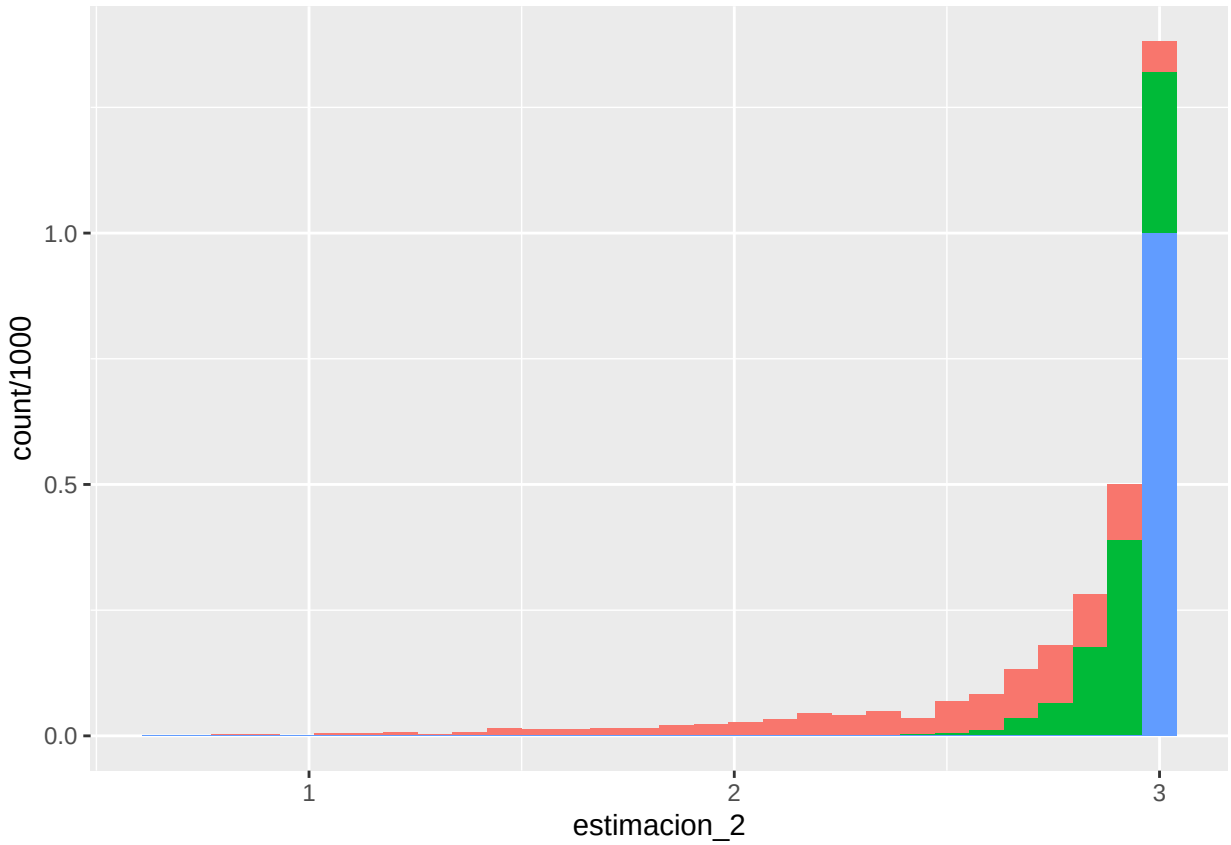
```
anim_1=estimacion_1 %>%
  ggplot()+
  geom_histogram(aes(x=estimacion_1,y=stat(count)/1000),
                 color='blue',fill='grey')+
  transition_states(estado, transition_length = 1, state_length = 1)+
  ggtitle(
    'Progresión est1',
    subtitle = 'n={closest_state}'
  )
# animate(anim_1,
#         width = 400, height = 400,
#         nframes = 480, fps = 24)
# anim_save("est1_2.gif", anim_1)
```

CÓMO PONGO DE SUBTITULO N IN C(5,30,500) Y QUE SE VAYA MOVIENDO CON EL FILTRO?

Ploteamos est2.

```
# nose como colorear usando filas, asi que agrego estados
estimacion_2$estado=as.factor(rep(c(5,30,500),rep(1000,3)))
estimacion_2 %>%
  ggplot(aes(fill=estado))+
  geom_histogram(aes(x=estimacion_2,y=stat(count)/1000))+
   scale_fill_discrete(name="Nº Variables",
                       breaks=1:3,
                       labels=paste("n=", enes))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Progresión est2 filtrando datos (filas) y que el plot entre y se vaya desvaneciendose.

```
# Como puse los datos verticalmente tenemos
# n=5 corresponden las filas 1:1000
# n=30 corresponden las filas 1000:2000
# n=500 corresponden las filas 2000:3000
anim_2=estimacion_2 %>%
  ggplot()+
  geom_histogram(aes(x=estimacion_2,y=stat(count)/1000),
                 color='blue',fill='grey')+
  transition_filter(
    transition_length = 3,
    filter_length = 1,
    1 <= filas & filas <= 1000 ,
    1000 <= filas & filas <= 2000,
    2000 <= filas
  )+
  ggtitle(
    'Progresión est2',
    subtitle = '{closest_expression}'
  ) +
  enter_fade()+
  exit_fade()
  # exit_fly(y_loc = 0)
# animate(anim_2,
#         width = 400, height = 400,
#         nframes = 480, fps = 24)
```

```
# anim_save("est2.gif", anim_2)
```

Progresión est1 filtrando estados y que el plot se mueva entre los distintos graficos.

```
anim_2=estimacion_2 %>%
  ggplot()+
  geom_histogram(aes(x=estimacion_2,y=stat(count)/1000),
                 color='blue',fill='grey')+
  transition_states(estado, transition_length = 1, state_length = 1)+
  ggtitle(
    'Progresión est2',
    subtitle = 'n={closest_state}'
  )
# animate(anim_1,
#         width = 400, height = 400,
#         nframes = 480, fps = 24)
# anim_save("est2_2.gif", anim_2)
```

Observamos en los gráficos que:

- La distribución de est1 es aproximadamente normal y a medida que n aumenta (simulamos mas datos) el estimador se concentra en el valor $2 * E(\mathcal{U}[0,3]) = 3$.
- La distribución est2 no es normal sino que se parece a una función cuadrática y a medida que n aumenta se concentra en 3.

Recuerde que el error cuadrático medio de un estimador $\widehat{\theta}_n = \widehat{\theta}_n(X_1, \ldots, X_n)$ está dado por

$$ECM = E(\widehat{\theta}_n - \theta)^2.$$

Para obtener el ECM necesitamos conocer la distribución de $\widehat{\theta}_n$. Sin embargo, cuando simulamos y generamos datos, podemos estimar el ECM con su versión empírica (ECME) haciendo

$$ECME = \frac{1}{N_{rep}} \sum_{k=1}^{N_{rep}} (\widehat{\theta}_{n,k} - \theta)^2,$$

siendo que $\widehat{\theta}_{n,k}$ es la estimación obtenida en la k-ésima replicación.

5. Presente en una tabla el error cuadrático medio empírico de los estimadores $\widehat{\theta}_n$ y $\tilde{\theta}$ para muestras de tamaño n = 5, n = 30, n = 50, n = 100 y 500, utilizando N rep = 1000 replicaciones en cada caso. Qué estimador elegiría?

Resolución:

Definimos la función ECME con entradas n, Nrep y estimador (que puede tomar los vlaores 1 para est1 y 2 para est2). La idea sera ejecutar var.gen que devuelve un data frame con Nrep filas y n columnas donde **cada fila es la simulación de un experimento con n v.a iid** $\mathcal{U}[0,3]$. Entonces ECME es calcular la estimacion correspondiente a cada fila, restarle θ en cada coord, elevar al cuadrado y calcularle la media al vector obtenido.

```
# con parámetros
ECME_param=function(n,Nrep,estimador){
  theta=3
  muestra=var.gen(n,Nrep,theta)
  if(estimador==1){
    estimacion=apply(muestra,1,est1)
    mean((estimacion-theta)^2)
  }else if(estimador==2){
```

```
    estimacion=apply(muestra,1,est2)
    mean((estimacion-theta)^2)
  }else{'Ingrese un estimador adecuado (1 ó 2)'}
}
# sin parametros
ECME=function(datos){
  theta=3
  mean((datos-theta)^2)
}
```

Generamos los datos necesarios para el plot.

```
# error_est1=error_est2=c()
# for(i in c(5,30,50,100,500)){
#   # genero datos
#   muestra=var.gen(i,1000,theta=3)
#   # calculo las estimaciones
#   estimacion_1=apply(muestra,1,est1)
#   estimacion_2=apply(muestra,1,est2)
#   # calculo errores y los guardo
#   error_est1=c(error_est1,ECME(estimacion_1))
#   error_est2=c(error_est2,ECME(estimacion_2))
# }
tabla_errores=function(enes,Nrep){
  error_est1=error_est2=c()
  for(i in enes){
    # genero datos
    muestra=var.gen(i,Nrep,theta=3)
    # calculo las estimaciones
    estimacion_1=apply(muestra,1,est1)
    estimacion_2=apply(muestra,1,est2)
    # calculo errores y los guardo
    error_est1=c(error_est1,ECME(estimacion_1))
    error_est2=c(error_est2,ECME(estimacion_2))
  }
  errores=data.frame(rbind(error_est1,error_est2))
  names(errores)=enes
  errores
}
```

UNA OBSERVACION: ACA ESTOY GENERANDO NUEVOS DATOS CON VAR.GEN, O SEA QUE ESTA TABLA NO ES LA TABLA DE ERRORES DE LOS PLOT QUE HICE ANTES SINO QUE SIMULE NUEVOS DATOS! EN TODO CASO LO QUE HAY QUE HACER ES USAR LOS DATOS QUE YA SIMULE PARA HACER LOS PLOTS DE EST1 Y EST2.

Muestro la tabla.

```
# genero los datos
datos_plot=tabla_errores(enes,1000)
datos_plot
```

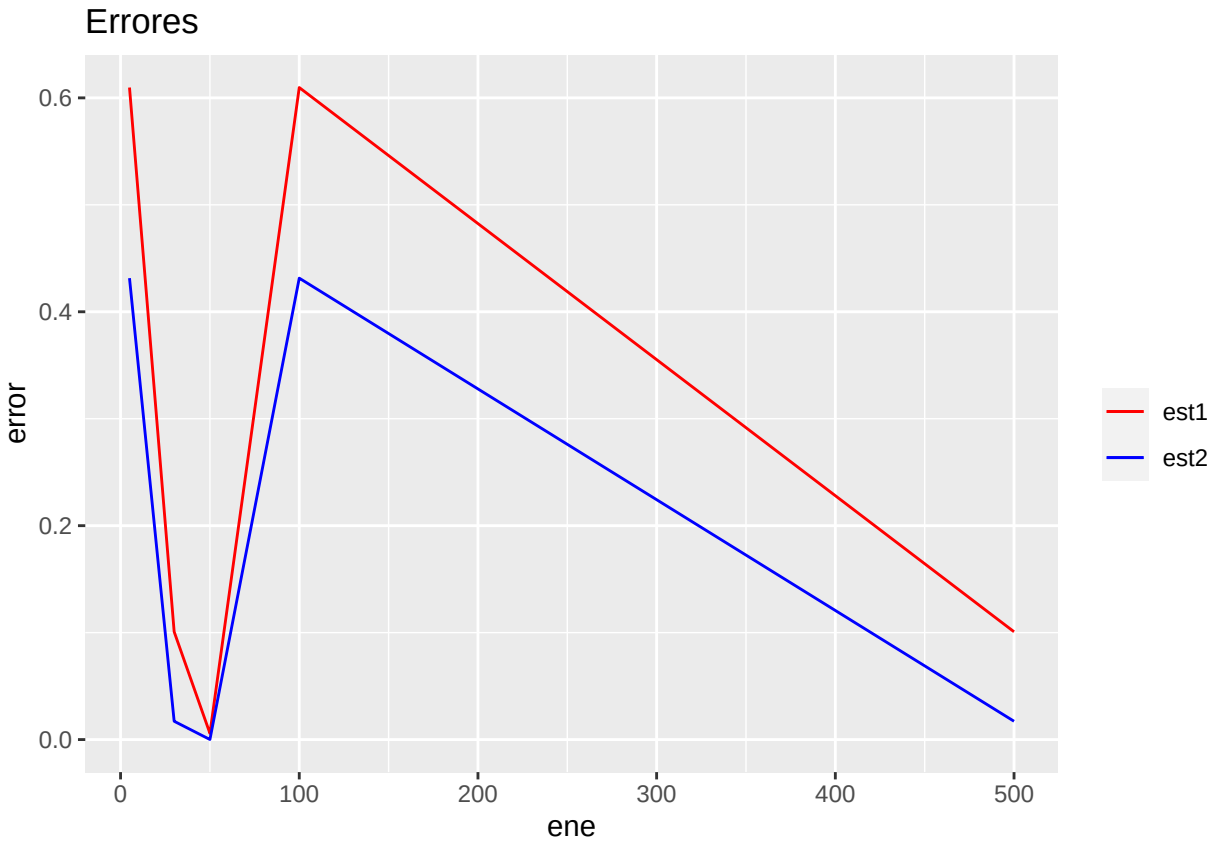
```
##                    5         30          500
## error_est1 0.6096695 0.10081826 5.810599e-03
## error_est2 0.4314447 0.01712239 6.720396e-05
```

Ploteamos.

```r
enes=c(5,30,50,100,500)
# organizo los datos en columnas
aux=stack(datos_plot)
datos_plot=data.frame(cbind('estimador_1'=aux[seq(1,dim(aux)[1],2),1],
                  'estimador_2'=aux[seq(2,dim(aux)[1],2),1],
                  'ene'=enes))
```

```
## Warning in cbind(estimador_1 = aux[seq(1, dim(aux)[1], 2), 1], estimador_2 =
## aux[seq(2, : number of rows of result is not a multiple of vector length (arg 1)
```

```r
ggplot(datos_plot)+
  geom_line(aes(x=ene,y=estimador_1,col='est1'))+
  geom_line(aes(x=ene,y=estimador_2,col='est2'))+
  scale_colour_manual("",
                      breaks = c("est1", "est2"),
                      values = c("red", "blue")) +
  xlab("ene") +
  scale_y_continuous("error") +
  labs(title="Errores")
```



Dado que el est2 tiene menos error, elegiría ese.

2 ¿A medida?

Sea (X i) i>1 una muestra aleatoria con distribucin F . Denotemos con X a un ele- mento con misma distribucin que X i . Asuma que estamos interesados en estimar el la probabilidad de que X sea mayor a uno: $\theta(F) := P_F(X > 1)$.

Estimador 1:

1.1 Proponga un estimador $\widehat{\theta}_n$ consistente para $\theta(F) = P_F(X > 1)$.

Resolución:

Proponemos la frecuencia relativa como estimador, es decir, $\widehat{\theta}_n = \frac{1}{n}\sum_i I_{X_i>1}$. Por la LGN sabemos que la frecuencia relativa es un estimador válido pero además es consistenete porque $E(\frac{1}{n}\sum_i I_{X_i>1}) = E(I_{X_i>1}) = P(X > 1)$.

1.2 Implemente una funcin est1 que tenga por argumento un conjunto de datos (x 1 , . . . , x n) muestra y devuelva el valor de la estimación obtenida utilizando $\widehat{\theta}_n$.

Resolución:

```
est1=function(datos){
  mean(datos>1)
}
```

1.3 Calcule el valor de $\widehat{\theta}_n$ en el siguiente conjunto de datos: 12.23 6.37 6.10 0.70 3.48 2.82 9.55 2.21 0.72 9.09.

Resolución:

```
datos=c(12.23, 6.37, 6.10, 0.70 ,3.48 ,2.82, 9.55, 2.21, 0.72 ,9.09)
est1(datos)
```

```
## [1] 0.8
```

Mundo Exponencial: Calentando motores

1.4 Sea X una variabe aleatoria con distribución F , exponencial de parámetro $\lambda = 0.2$: $X \sim \mathcal{E}(0.2)$. Indique el valor de E(X), V(X), P(X > 1) para $X \sim \mathcal{E}(0.2)$.

1.5 Sea ahora X una variabe aleatoria con distribución F pertenecinete a la familia exponencial: es decir, $X \sim \mathcal{E}(\lambda)$ con λ DESCONOCIDO. Exprese cada uno de los siguientes objetos en función de λ: E(X), V(X), P(X > 1) cuando $X \sim \mathcal{E}(\lambda)$.

Resolución:

Resolvemos sólo el item 1.5 que incluye al 1.4.

Recordemos que una variable aleatoria $X \sim \mathcal{E}(\lambda)$ tiene función de densidad

$$f_X(x) = \begin{cases} \lambda e^{-\lambda x} & \text{para } x \geq 0 \\ 0 & \text{c.c.} \end{cases}$$

y función de distribución

$$F_X(x) = P(X \leq x) = \begin{cases} 0 & \text{para } x < 0 \\ 1 - e^{-\lambda x} & \text{para } x \geq 0 \end{cases}$$

Luego

$$\begin{aligned} E(X) &= \int_{-\infty}^{\infty} s f_X(s) ds = \int_0^{\infty} s\lambda e^{-\lambda s} \\ &= \lambda[s(\frac{-1}{\lambda}e^{-\lambda s}) - \int_0^{\infty} 1(\frac{-1}{\lambda}e^{-\lambda s})ds] \\ &= (-se^{-\lambda s} - \frac{-1}{\lambda}e^{-\lambda s})|_0^{\infty} = \frac{1}{\lambda}; \\ E(X^2) &= \int_{-\infty}^{\infty} s^2 f_X(s) ds = \int_0^{\infty} s^2\lambda e^{-\lambda s} \\ &= \lambda[s^2(\frac{-1}{\lambda}e^{-\lambda s}) - \int_0^{\infty} 2s(\frac{-1}{\lambda}e^{-\lambda s})ds] \\ &= -s^2 e^{-\lambda s}|_0^{\infty} + 2\int_0^{\infty} s(e^{-\lambda s})ds = 2\frac{E(X)}{\lambda} = \frac{2}{\lambda^2}; \\ V(X) &= E(X^2) - E(X)^2 = \frac{2}{\lambda^2} - \frac{1}{\lambda^2} = \frac{1}{\lambda^2}; \\ P(X > 1) &= 1 - F_X(1) = 1 - (1 - e^{-\lambda}) = e^{-\lambda}. \end{aligned}$$

Mundo Exponencial: Haciendo Estadística

Sean (X i) i>1 i.i.d., con misma distribucion que X. Asuma ahora que F pertenece a la familia exponencial; es decir, $X \sim \mathcal{E}(\lambda)$, con λ DESCONOCIDO.

1.6 Proponga un nuevo estimador $\tilde{\theta}_n$ consistente para $\theta(F) = P_F(X > 1)$ bajo este nuevo escenario. Es decir, defina $\tilde{\theta}_n = f_n(X_1, \ldots, X_n)$ de forma tal que

$$\tilde{\theta}_n = f_n(X_1, \ldots, X_n) \xrightarrow{p} e^{-\lambda} \text{ cuando } X_i \sim \mathcal{E}(\lambda), \forall \lambda > 0$$

Resolución:

Sabemos que $\bar{X}_n \xrightarrow{p} E(X) = \frac{1}{\lambda}$, entonces proponemos $\tilde{\theta}_n = e^{-\frac{1}{\bar{X}_n}}$.

1.7 Implemente una funcion est2 que tenga por argumento un conjunto de datos (x 1 , . . . , x n) muestra y devuelva el valor de la estimación obtenida utilizando $\tilde{\theta}_n$.

Resolución:

```
est2=function(datos){
  exp(-1/mean(datos))
}
```

1.8 Calcule el valor de $\tilde{\theta}_n$ en el siguiente conjunto de datos: 12.23 6.37 6.10 0.70 3.48 2.82 9.55 2.21 0.72 9.09.

Resolución:

```
datos=c(12.23, 6.37 ,6.10 ,0.70, 3.48, 2.82 ,9.55 ,2.21 ,0.72 ,9.09)
est2(datos)
```

```
## [1] 0.8288443
```

Simulacin 1:.

A lo largo de esta simulacin generaremos variables con distribucin exponencial de paramtro $\lambda = 0.2$.

1.9 Indique cual es el veradero valor que estamos queriendo estimar: $\theta_0 = P(X > 1)$, siendo $X \sim \mathcal{E}(0.2)$.

Resolución:

El verdadero valor que estamos queriendo simular es

$$P(X > 1) = 1 - F_E(1) = 1 - (1 - e^{-0.2}) = e^{-0.2}.$$

```
exp(-0.2)
```

```
## [1] 0.8187308
```

1.10 Genere una muestras de tamaño n=50 y calcule cada uno de los estimadores.

Resolución:

```
muestra=rexp(50,rate=0.2)
est1(muestra)
```

```
## [1] 0.86
```

```
est2(muestra)
```

```
## [1] 0.7927675
```

1.11 Genere N rep= 1000 muestras de tamao n=50 y guarde los valores de cada uno de los dos estimadores calculados en cada uno de los N rep = 1000 conjuntos de datos.

Resolución:

Defino fc simuladora de datos.

```
# simulo Nrep experimentos con n v.a. iid
# y pongo cada resultado en una fila distinta
var.gen.exp=function(n,Nrep,lambda){
   tabla=c()
   for(i in (1:n)){
      tabla=cbind(tabla,rexp(Nrep,rate=lambda))
   }
   data.frame(tabla)
}
```

Generamos los datos.

```
Nrep=1000
lambda=0.2
enes=c(50)
estimacion_1=estimacion_2=matrix()
for(i in enes){
  # simulo Nrep experimentos con n v.a. iid E(lambda)
  # y pongo cada resultado en una fila distinta
  muestra=var.gen.exp(i,Nrep,lambda)
  # le aplico est1 a cada fila
  # i.e. estimo con el primer estimador los datos
  # obtenidos en cada experimento (cada fila)
  aux_1=data.frame(apply(muestra,1,est1))
  names(aux_1)=paste('est1_',i)
  estimacion_1=cbind(estimacion_1,aux_1)
```

```
  # le aplico est2 a cada fila
  aux_2=data.frame(apply(muestra,1,est2))
  names(aux_2)=paste('est2_',i)
  estimacion_2=cbind(estimacion_2,aux_2)
  # estimaciones=cbind(estimaciones,estimacion_1,estimacion_2)
}
# le saco la columna 0 que tiene NAs
# y como tiene 1 sola fila lo transformo en data.frame
# sino lo toma como un vector
estimacion_1=data.frame('estimacion_1'=estimacion_1[,2:dim(estimacion_1)[2]])
estimacion_2=data.frame('estimacion_2'=estimacion_2[,2:dim(estimacion_2)[2]])
# # verticalizamos los datos y les ponemos nombre
# estimacion_1=stack(estimacion_1)[1]
# names(estimacion_1)="estimacion_1"
# estimacion_2=stack(estimacion_2)[1]
# names(estimacion_2)="estimacion_2"
# le pongo una variable de filas para la transicion del gif
estimacion_1$estado=as.factor(rep(enes,rep(Nrep,length(enes))))
estimacion_2$estado=as.factor(rep(enes,rep(Nrep,length(enes))))
```

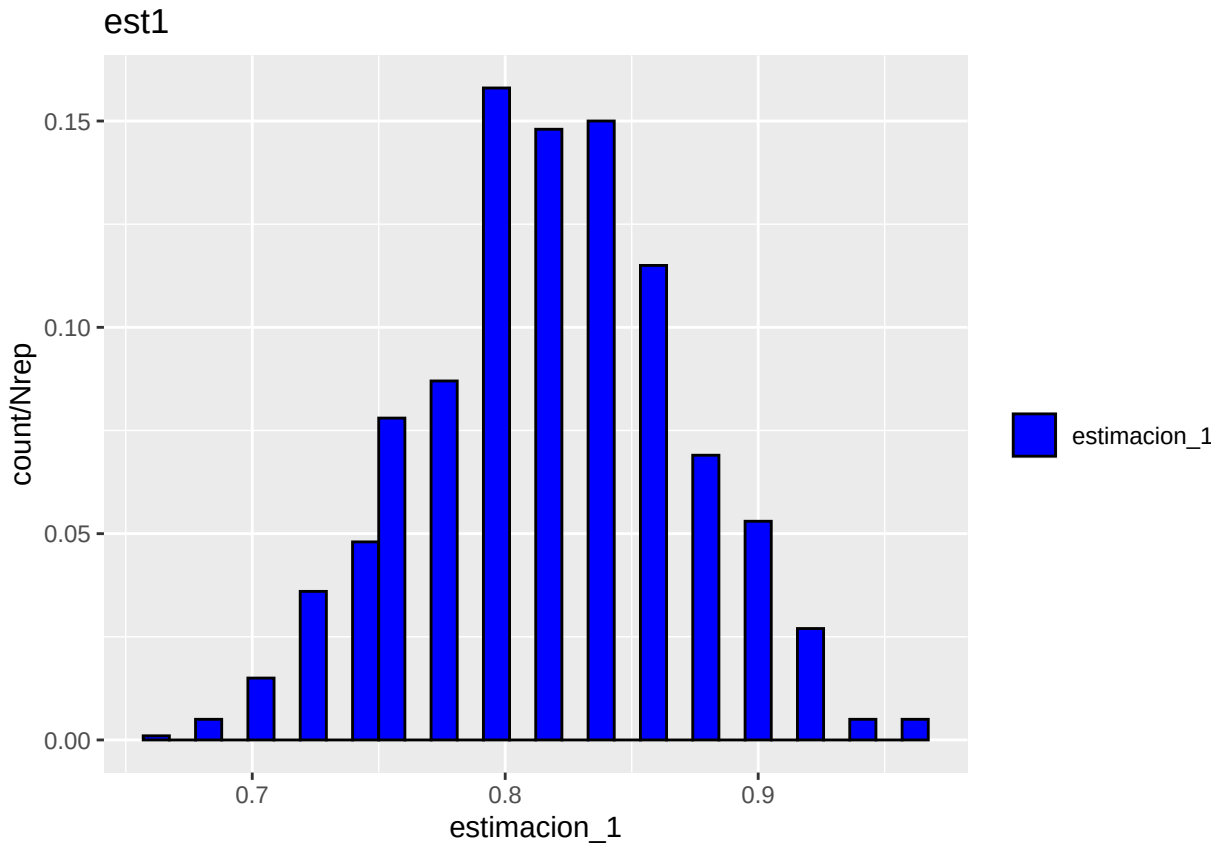
1.12 Realize un histograma de cada uno de los estimadores propuestos con los val- ores obtenidos en el item anterior. Comente los gráficos realizados. Indique que etimador prefiere en este escenario y explique a que atribuye sus bondades.

Resolución:

Graficamos cada plot por separado.

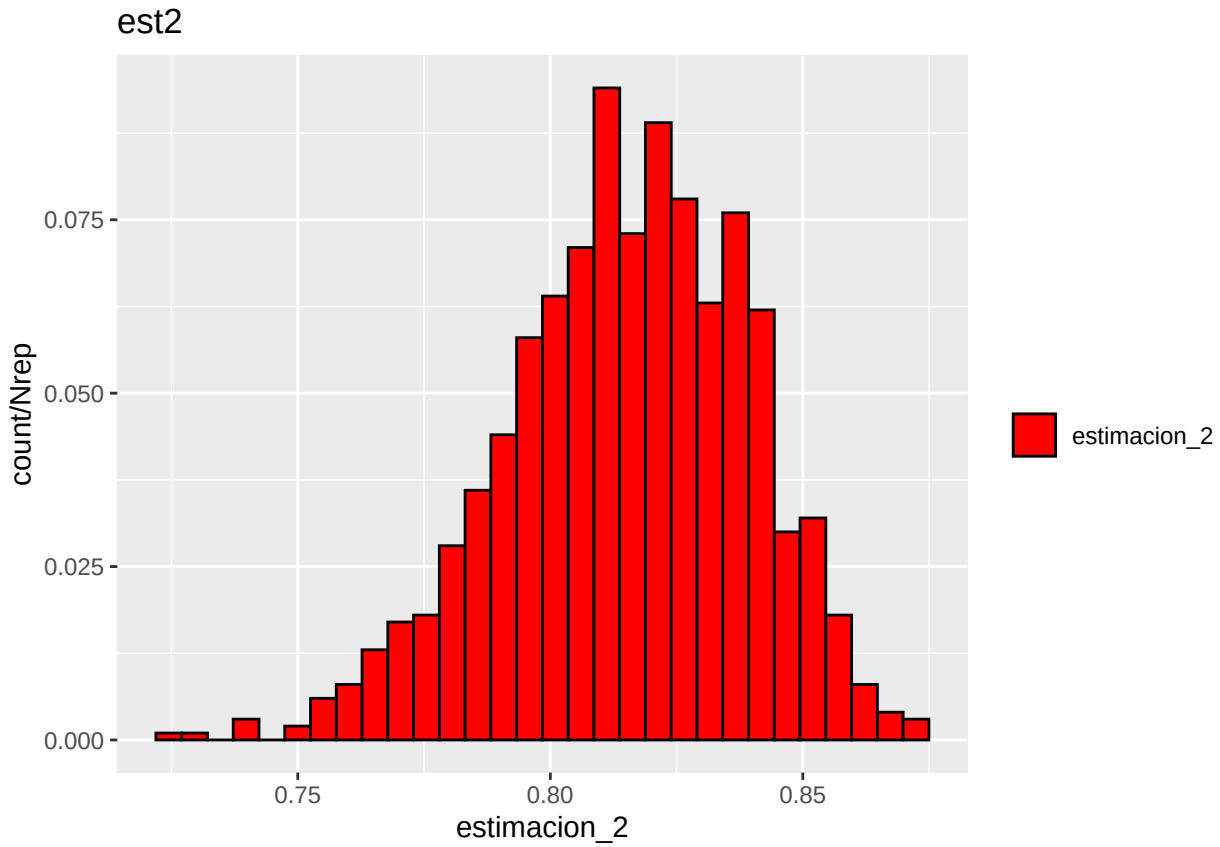
```
estimacion_1 %>%
  ggplot()+
  geom_histogram(aes(x=estimacion_1,y=stat(count)/Nrep,fill='estimacion_1'),color='black')+
   scale_fill_manual("",
                     breaks = c("estimacion_1"),
                     values = c("blue")) +
  labs(title="est1")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
estimacion_2 %>%
  ggplot()+
  geom_histogram(aes(x=estimacion_2,y=stat(count)/Nrep,fill='estimacion_2'),color='black')+
   scale_fill_manual("",
                     breaks = c("estimacion_2"),
                     values = c("red")) +
  labs(title="est2")
```

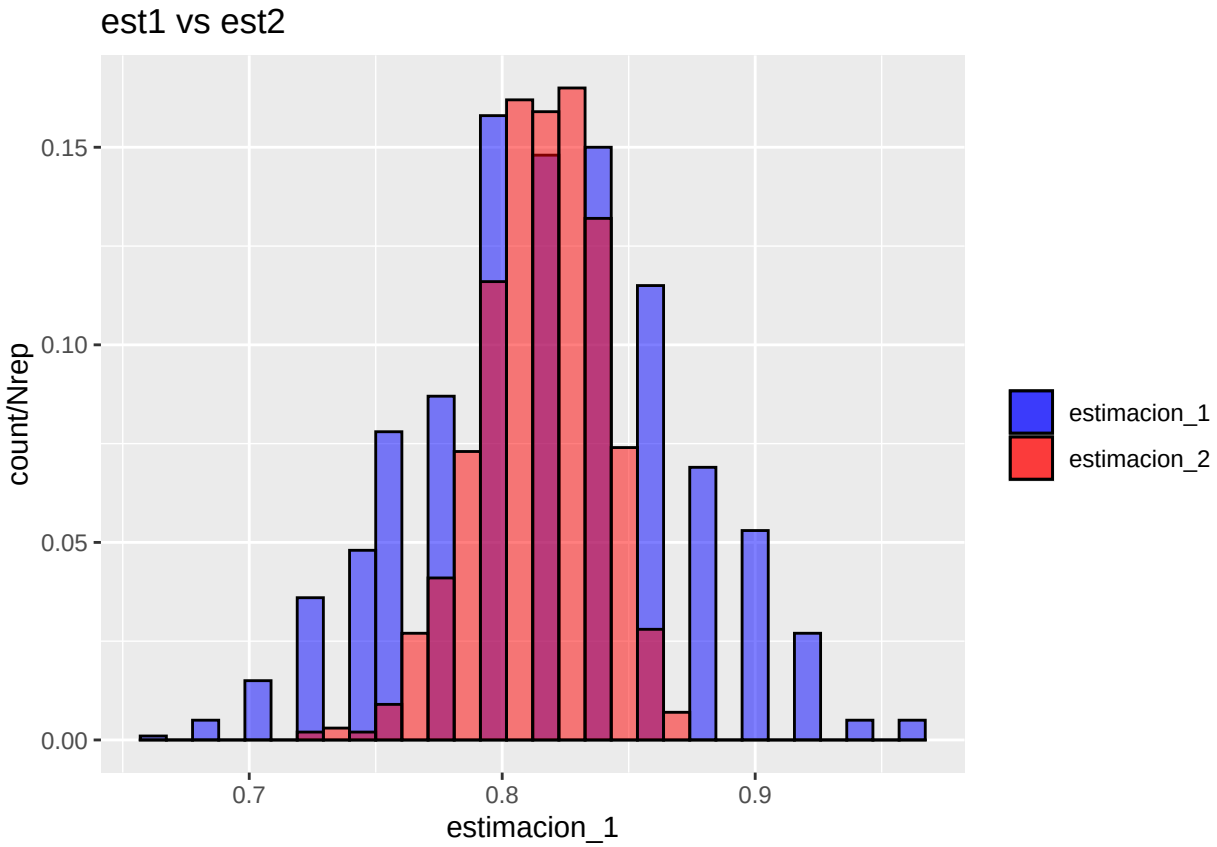
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Ahora los ploteamos juntos.

```
datos_plot=cbind('estimacion_1'=estimacion_1[,1],estimacion_2)
datos_plot %>%
  ggplot()+
  geom_histogram(aes(x=estimacion_1,y=stat(count)/Nrep,fill='estimacion_1'),color='black',alpha=0.5)+
  geom_histogram(aes(x=estimacion_2,y=stat(count)/Nrep,fill='estimacion_2'),color='black',alpha=0.5)+
  scale_fill_manual("",
                    breaks = c("estimacion_1", "estimacion_2"),
                    values = c("blue", "red")) +
  labs(title="est1 vs est2")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



1.13 Represente en una tabla el error cuadrtico medio (estimado) de los estimadores $\widehat{\theta}_n$ y $\tilde{\theta}_n$ para muestras de tamao n=150, n=200, n=500 y n=1000, utilizando Nrep=1000 replicaciones en cada caso. Qué estimador prefiere bajo este escenario?

Resolución:

Primero defino la función de error cuadrático medio empírico para este contexto.

```
ECME_exp=function(datos){
  lambda=0.2
  mean((datos-exp(-lambda))^2)
}
```

Generamos los datos necesarios para el plot.

```
tabla_errores_exp=function(enes,Nrep){
  error_est1=error_est2=c()
  for(i in enes){
    # genero datos
    muestra=var.gen.exp(i,Nrep,lambda=0.2)
    # calculo las estimaciones
    estimacion_1=apply(muestra,1,est1)
    estimacion_2=apply(muestra,1,est2)
    # calculo errores y los guardo
    error_est1=c(error_est1,ECME_exp(estimacion_1))
    error_est2=c(error_est2,ECME_exp(estimacion_2))
  }
  errores=data.frame(rbind(error_est1,error_est2))
  names(errores)=enes
```

```
  errores
}
```

UNA OBSERVACION: ACA ESTOY GENERANDO NUEVOS DATOS CON VAR.GEN.EXP, O SEA QUE ESTA TABLA NO ES LA TABLA DE ERRORES DE LOS PLOT QUE HICE ANTES SINO QUE SIMULE NUEVOS DATOS! EN TODO CASO LO QUE HAY QUE HACER ES USAR LOS DATOS QUE YA SIMULÉ PARA HACER LOS PLOTS DE EST1 Y EST2.

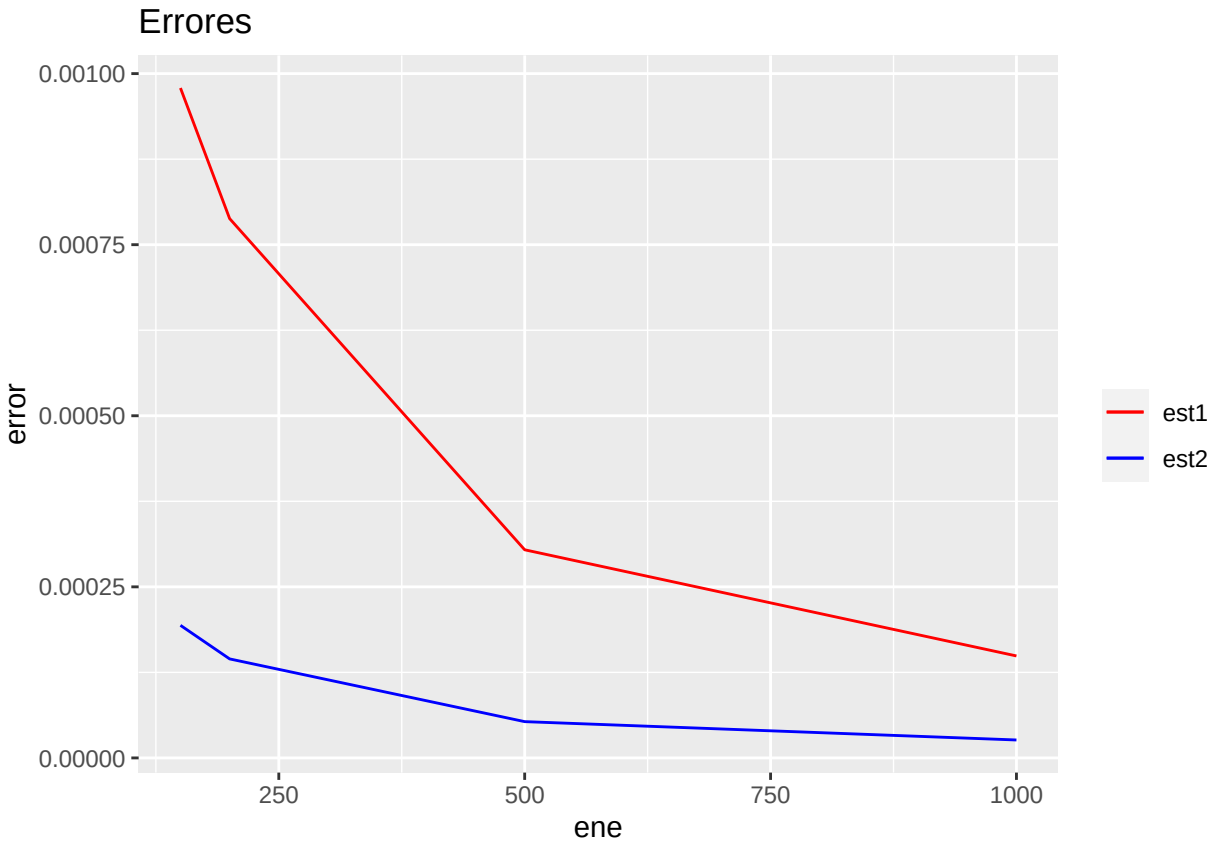
Muestro la tabla de errores.

```
# genero los datos
enes=c(150,200,500,1000)
Nrep=1000
datos_plot=tabla_errores_exp(enes,Nrep)
datos_plot
```

```
##                      150          200          500         1000
## error_est1 0.0009789718 0.0007881194 0.0003041492 1.489916e-04
## error_est2 0.0001937418 0.0001446675 0.0000530189 2.619342e-05
```

Ploteamos.

```
# organizo los datos en columnas
aux=stack(datos_plot)
datos_plot=data.frame(cbind('estimador_1'=aux[seq(1,dim(aux)[1],2),1],
                 'estimador_2'=aux[seq(2,dim(aux)[1],2),1],
                 'ene'=enes))
ggplot(datos_plot)+
  geom_line(aes(x=ene,y=estimador_1,col='est1'))+
  geom_line(aes(x=ene,y=estimador_2,col='est2'))+
  scale_colour_manual("",
                      breaks = c("est1", "est2"),
                      values = c("red", "blue")) +
  xlab("ene") +
  scale_y_continuous("error") +
  labs(title="Errores")
```


Errores
0.00100
0.00075
0.00050
0.00025
0.00000
error
250
500
750
1000
ene
est1
est2

Mundo Normal: Ojo al Piojo!

Considere ahora variables aleatorias X i i.i.d. con distribucin normal de media $\mu = 1/0.2$ y $\sigma^2 = 1/0.2^2$.

1.14 Calcule la probabilidad de que X i supere el valor 1: P(X i > 1)

Resolución:

Recordemos que la distribución norma tiene densidad

$$f_X(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}},$$

y función de distribución dada por

$$F_X(x) = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{x} e^{-\frac{(u-\mu)^2}{2\sigma^2}} du.$$

Luego

$$\begin{aligned} P(X_i > 1) &= 1 - F_X(1) \\ &= 1 - \frac{0.2}{\sqrt{2\pi}} \int_{-\infty}^{1} e^{-\frac{(u-1/0.2)^2}{2*(1/0.2^2)}} du \\ &= 1 - \frac{0.2}{\sqrt{2\pi}} \int_{-\infty}^{1} e^{-\frac{(0.2u-1)^2}{2}} du \end{aligned}$$

1.15 Calcule el valor de cada uno de los siguientes lmites:

$$\lim_{n\to\infty} \widehat{\theta}_n(X_1, \ldots, X_n), \ \lim_{n\to\infty} \tilde{\theta}_n(X_1, \ldots, X_n);$$

Resolución:

$$\begin{aligned} \lim_{n\to\infty} \widehat{\theta}_n(X_1, \ldots, X_n) &= \lim_{n\to\infty} \frac{1}{n} \sum_i I_{X_i>1} \\ &\stackrel{p}{=} E(I_{X_i>1}) = P(X > 1); \\ \lim_{n\to\infty} \tilde{\theta}_n(X_1, \ldots, X_n) &= \lim_{n\to\infty} e^{-\frac{1}{X_n}} \\ &\stackrel{p}{=} e^{-\frac{1}{\lambda}} = e^{-\lambda}. \end{aligned}$$

1.16 Proponga un nuevo estimador $\theta_n^* = \theta_n^*(X_1, \ldots, X_n)$ para $\theta(F) = P_F(X_i > 1)$, asumiendo ahora que F pertenece a la normal: $X_i \sim \mathcal{N}(\mu, \sigma^2)$.

Resolución:

Quiero $P(X > 1) = 1 - P(X < 1) = 1 - F(1) = 1 - \phi(\frac{X-\mu}{\sigma})$ entonces propongo $1 - pnorm(1, \tilde{\mu}, \tilde{\sigma})$, con $\tilde{\mu} = mean(datos) = \overline{X_n}, \tilde{\sigma} = sd(datos)$ los valores de los parámetros estimados a partir de los datos.

Simulacion 2:

A lo largo de esta simulacin generaremos variables con distribucin normal de media $\mu = 1/0.2$ y $\sigma^2 = 1/0.2^2$. Represente en una tabla el error cuadratico medio (estimado) de los estimadores $\widehat{\theta}_n$, $\tilde{\theta}_n$ y θ_n^* para muestras de tamaño n=150, n=200, n=500 y n=1000, utilizando Nrep=1000 replicaciones en cada caso. Analice los resultados obtenidos y explique que estimador elegira bajo este escenario.

Resolución:

Definimos primero la función error cuadrático medio empírico para este contexto.

```
ECME_norm=function(datos){
  mu=1/0.2
  sd=1/0.2
  valor_a_estimar=1-pnorm(1,mu,sd)
  mean((datos-valor_a_estimar)^2)
}
```

Definimos ahora una función que simule Nrep experimentos de n v.a. iid N(1/0.2,1/0.2^2), el nuevo estimador $est3 = 1 - pnorm(1, \tilde{\mu}, \tilde{\sigma})$ y la función que arma la tabla de errores.

```
# simulo Nrep experimentos con n v.a. iid
# y pongo cada resultado en una fila distinta
var.gen.norm=function(n,Nrep,mu,sd){
   tabla=c()
   for(i in (1:n)){
      tabla=cbind(tabla,rnorm(Nrep,mean=mu,sd))
   }
   data.frame(tabla)
}
####################################
est3=function(datos){
  1-pnorm(1,mean(datos),sd(datos))
}
####################################
tabla_errores_norm=function(enes,Nrep){
  error_est1=error_est2=error_est3=c()
  for(i in enes){
    # genero datos
    muestra=var.gen.norm(i,Nrep,mu=1/0.2,sd=1/0.2)
    # calculo las estimaciones
    estimacion_1=apply(muestra,1,est1)
    estimacion_2=apply(muestra,1,est2)
    estimacion_3=apply(muestra,1,est3)
    # calculo errores y los guardo
    error_est1=c(error_est1,ECME_norm(estimacion_1))
    error_est2=c(error_est2,ECME_norm(estimacion_2))
    error_est3=c(error_est3,ECME_norm(estimacion_3))
  }
  errores=data.frame(rbind(error_est1,error_est2,error_est3))
  names(errores)=enes
  errores
}
```

UNA OBSERVACION: ACA ESTOY GENERANDO NUEVOS DATOS CON VAR.GEN.EXP, O SEA QUE ESTA TABLA NO ES LA TABLA DE ERRORES DE LOS PLOT QUE HICE ANTES SINO QUE SIMULE NUEVOS DATOS! EN TODO CASO LO QUE HAY QUE HACER ES USAR LOS DATOS QUE YA SIMULÉ PARA HACER LOS PLOTS DE EST1 Y EST2.

Generamos los datos necesarios para el plot y muestro la tabla de errores.

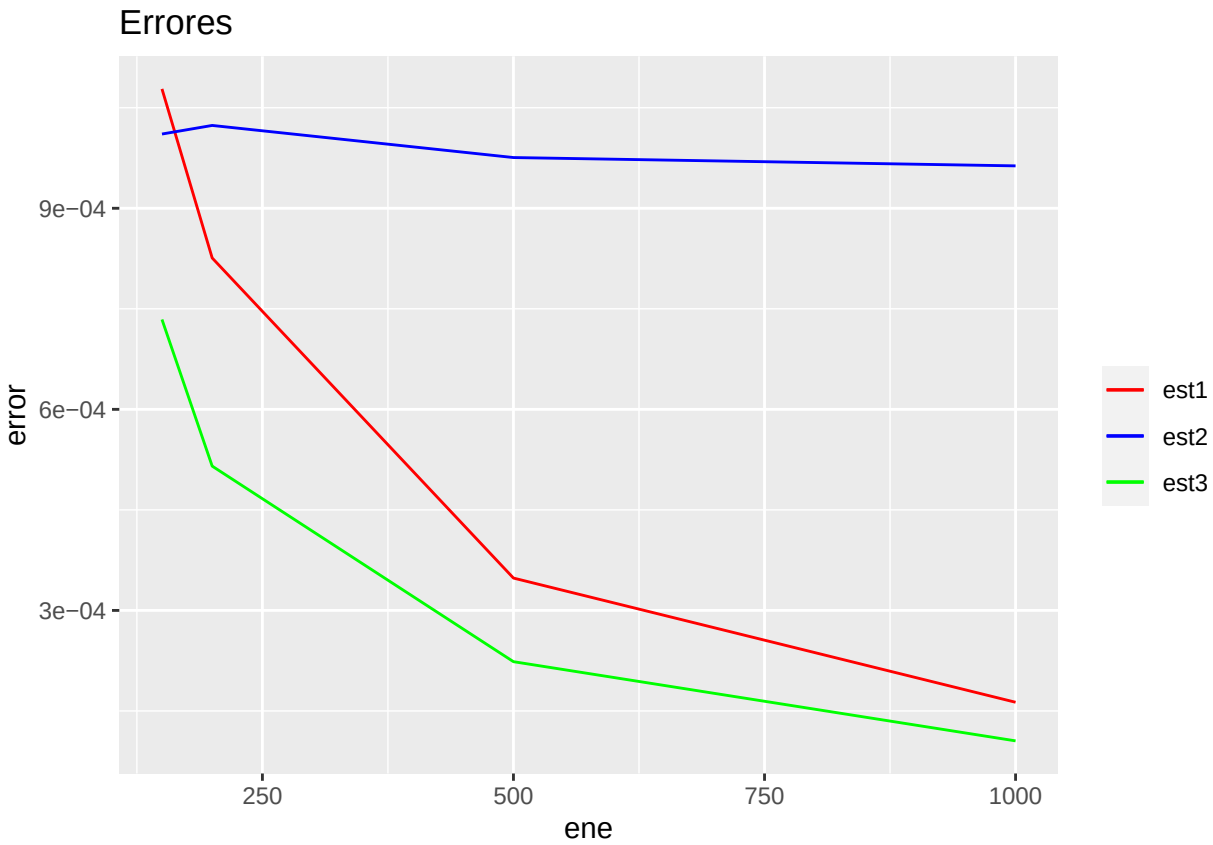
```
# genero los datos
enes=c(150,200,500,1000)
Nrep=1000
```

```
datos_plot=tabla_errores_norm(enes,Nrep)
datos_plot
```

```
##                      150          200          500         1000
## error_est1 0.0010781101 0.0008258432 0.0003482689 0.0001629706
## error_est2 0.0010107819 0.0010236009 0.0009756848 0.0009632562
## error_est3 0.0007341122 0.0005153070 0.0002235517 0.0001053464
```

Ploteamos.

```
# organizo los datos en columnas
aux=stack(datos_plot)
datos_plot=data.frame(cbind('estimador_1'=aux[seq(1,dim(aux)[1],3),1],
                 'estimador_2'=aux[seq(2,dim(aux)[1],3),1],
                 'estimador_3'=aux[seq(3,dim(aux)[1],3),1],
                 'ene'=enes))
ggplot(datos_plot)+
  geom_line(aes(x=ene,y=estimador_1,col='est1'))+
  geom_line(aes(x=ene,y=estimador_2,col='est2'))+
  geom_line(aes(x=ene,y=estimador_3,col='est3'))+
  scale_colour_manual("",
                      breaks = c("est1", "est2", "est3"),
                      values = c("red", "blue", "green")) +
  xlab("ene") +
  scale_y_continuous("error") +
  labs(title="Errores")
```



Claramente el mejor estimador es el est3.