Guia 15

Agustin Muñoz Gonzalez

3/6/2020

Preparamos el entorno

```
# rm(list=ls())
library(ggplot2)
```

1. En este ejercicio estudiaremos el comportamiento de la suma Sn y del promedio $\bar{X}n$ de variables X_1,. . . , X_n independientes e idénticamente distribuidas (i.i.d.) donde Xi~N(μ, σ^2). A través de los correspondientes histogramas analizaremos el comportamiento de la distribución de la suma y del promedio $\bar{X}n$, a medida que n va aumentando. Para ello, fijado n, generaremos datos correspondientes a una muestra X 1 ,. . . , X n i.i.d. de variables aleatorias distribuidas como X, con una distribución N (5, 4) y luego calcularemos la suma y el promedio de cada conjunto de datos. Repetimos este procedimiento N rep = 1000 veces. A partir de las N rep = 1000 replicaciones realizaremos un histograma con las sumas y los promedios generados, para obtener una aproximación de la densidad de S n y de $\bar{X}n$.

Notas previas:

Voy a definir una función que simule los Nrep experimentos de generar n variables aleatorias N(5,4). Se me ocurren dos formas de hacer esto:

-Una forma es con un for (1:Nrep) y en cada iteración genero n v.a. N(5,4). Es decir, cada iteración es un experimento. -Otra es con un for (1:n) y en cada iteración genero los Nrep experimentos. Es decir, cada iteración es la simulación de una variable.

La defino para que devuelva un data.frame donde cada fila representa un experimento (ncol=Nrep) y cada columna una variable (nrow=n).

```
var.gen=function(n,Nrep){
   tabla=c()
   for(i in (1:n)){
      tabla=cbind(tabla,rnorm(Nrep,5,4))
   }
   data.frame(tabla)
}
```

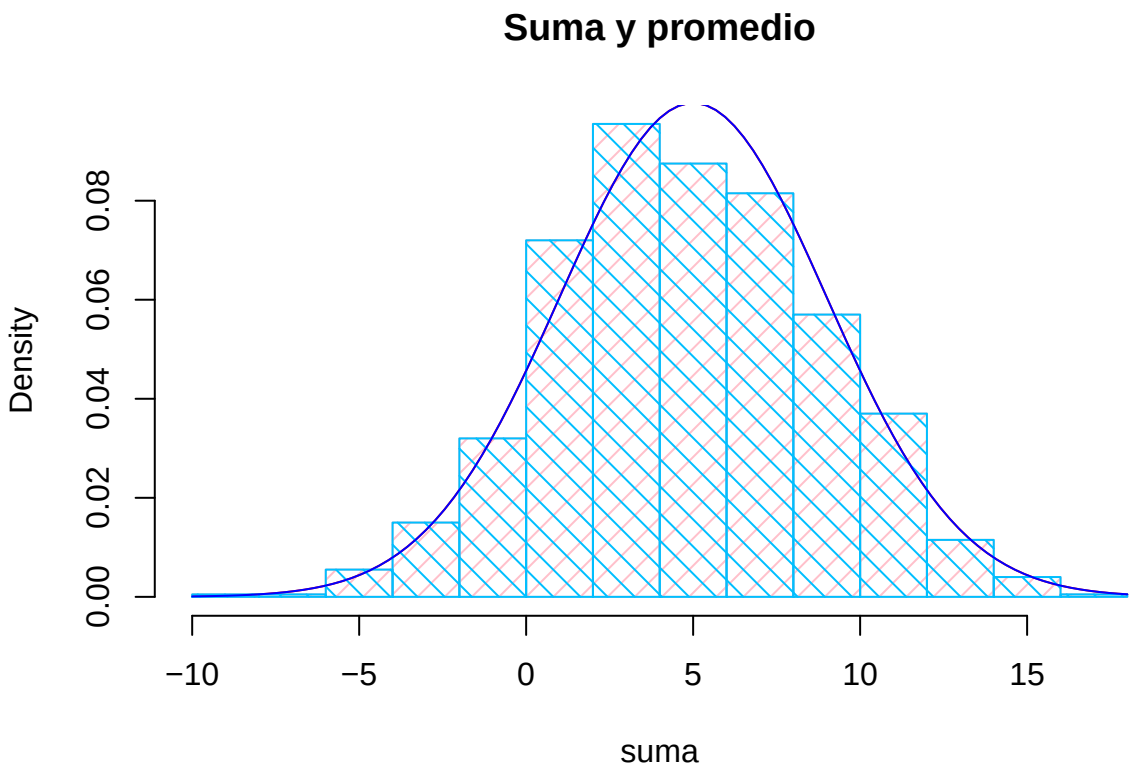
a) Consideremos n = 1: la variable coincide con la suma y el promedio. Generamos entonces N rep = 1000 datos correspondientes a X ~ N (5, 4) y luego hacemos los histogramas de suma y promedio. ¿A qué densidad se parece el histograma obtenido?

Resolución:

Grafiquemos primero con hist.

```
data=var.gen(1,1000)
# como los experimentos son las filas suma y producto es
# sumar y promediar cada fila y por eso apply(-,1,-)
suma=apply(data,1,sum)
```

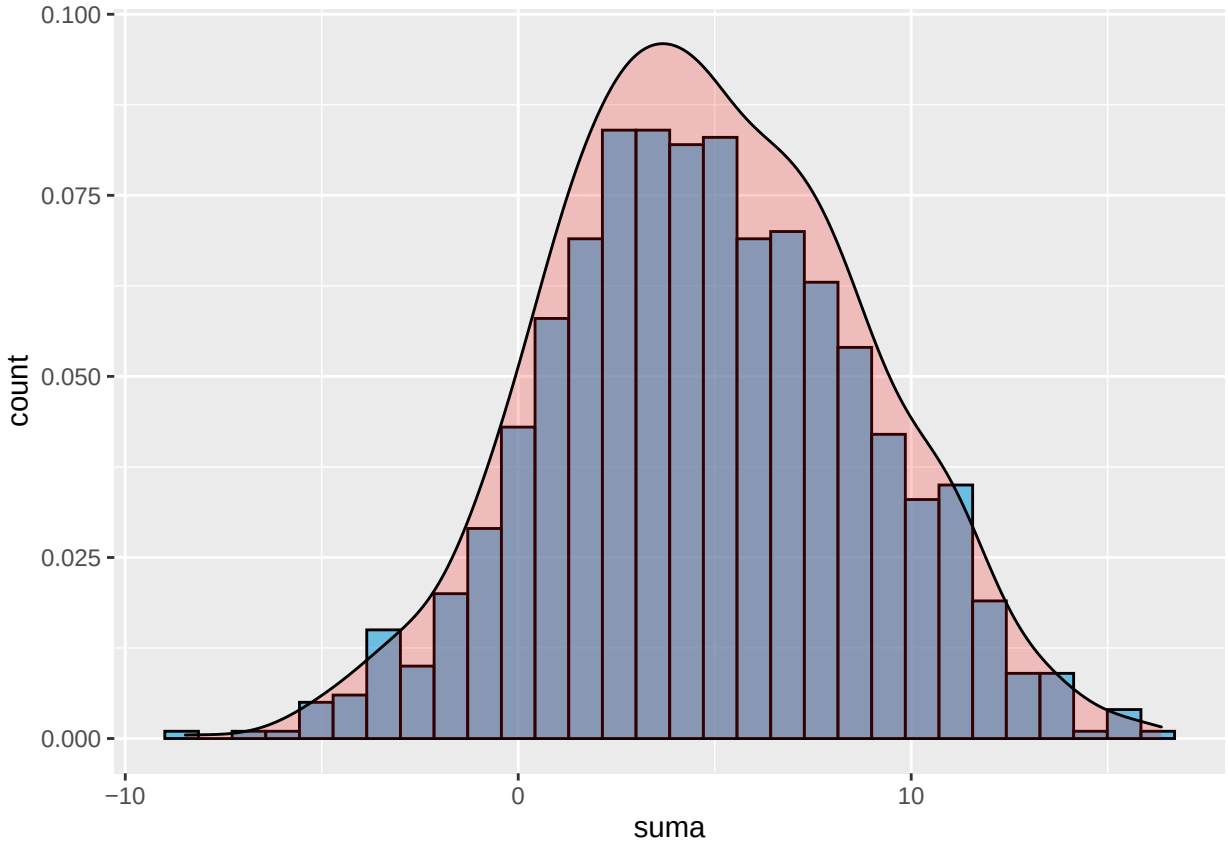
```r
prom=apply(data,1,mean)
hist(suma,freq=F,col='pink',main='Suma y promedio',
     density=10,angle=45)
curve(dnorm(x,5,4),add=T,col='red')
hist(prom,freq=F,col='DeepSkyBlue1',main='Promedio',add=T,
     density=10,angle=135)
curve(dnorm(x,5,4),add=T,col='blue')
```



Jueguemos un poco con ggplot.

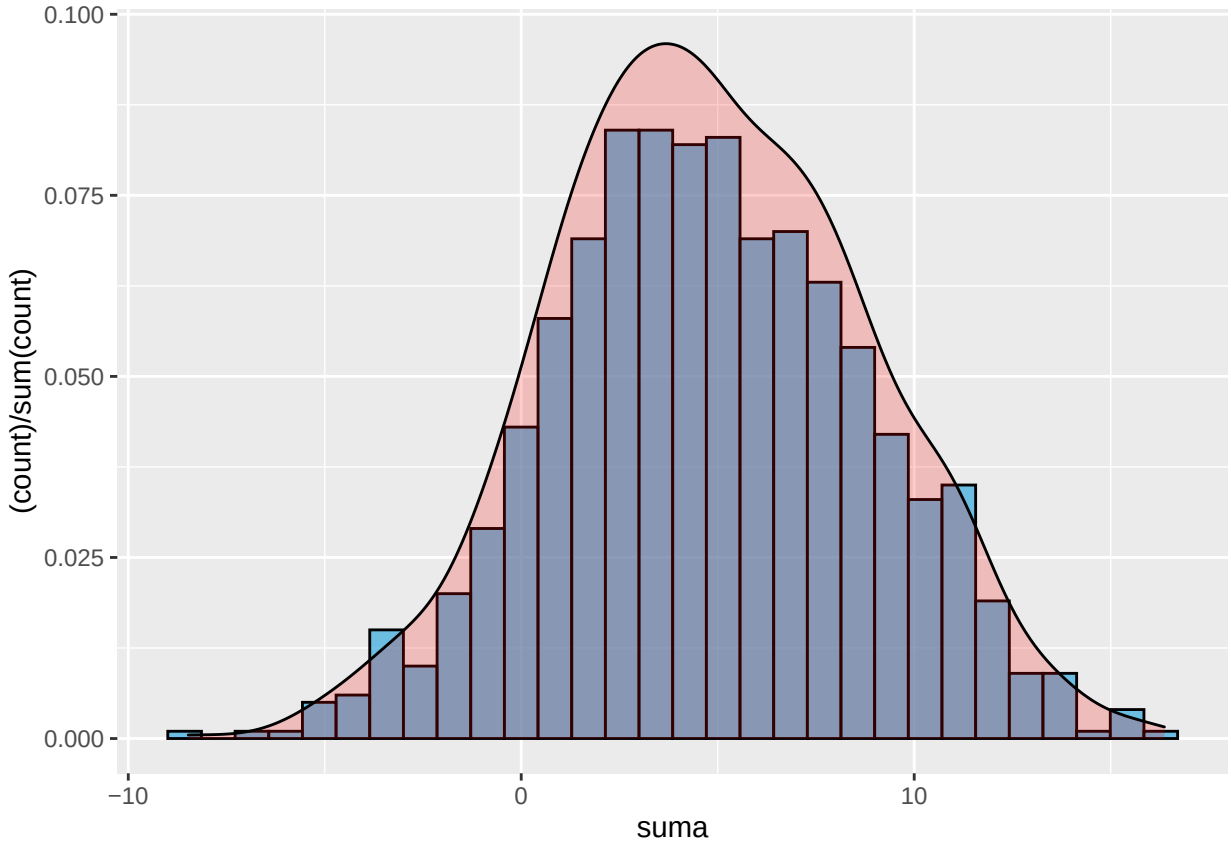
```r
ggplot()+
   geom_histogram(aes(x=suma,weight=1/1000),alpha=0.7, fill="#33AADE", color="black") +
   # alpha establece la opacidad de la capa
   # fill (color de relleno) está dado es la rgb specification
   # i.e. "#RRGGBB" donde RR, GG, BB está en hexadecimal
   geom_density(aes(x=suma),alpha=0.2, fill="red")
```

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



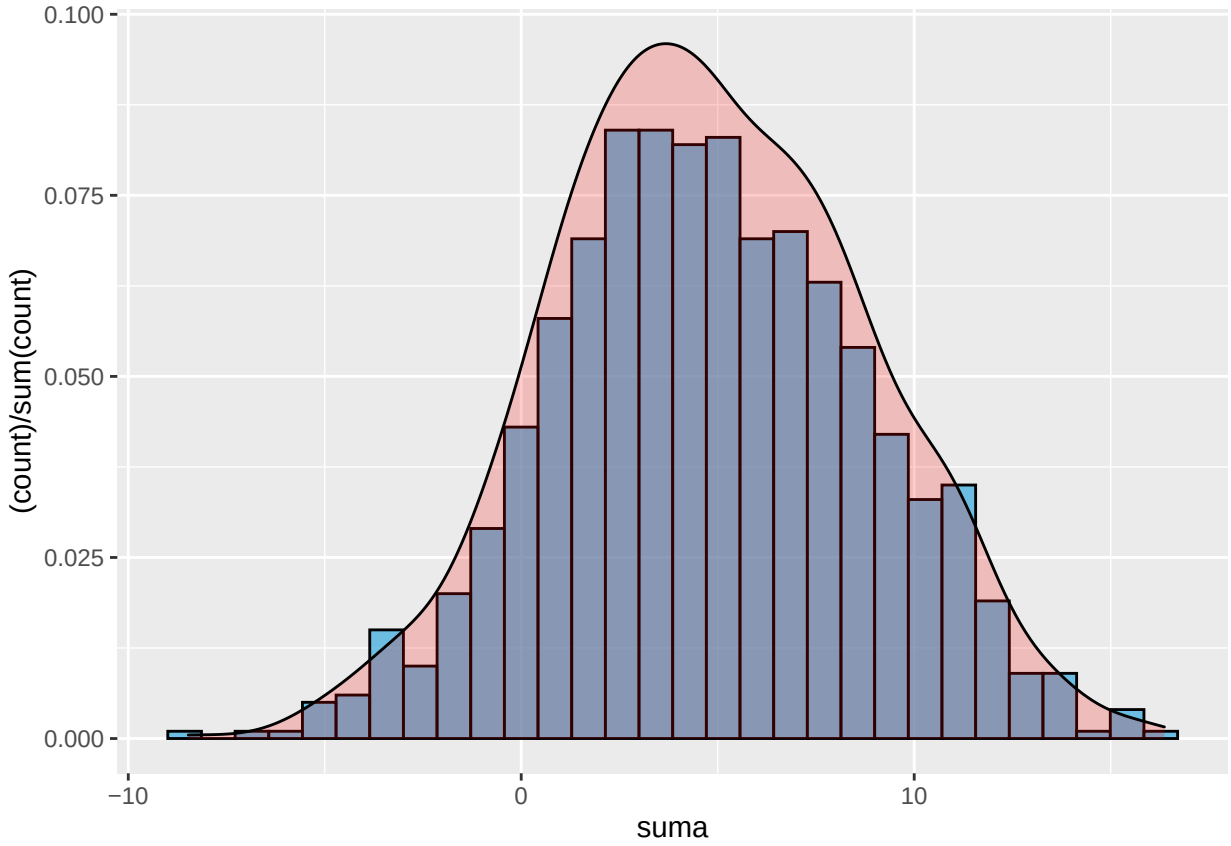

```
# Además le agrego aes(weight=1/1000) para que en el eje y muestra
# freq relativ, i.e. divido cada valor de y por Nrep=1000
ggplot()+
   geom_histogram(aes(x=suma,y = (..count..)/sum(..count..)),alpha=0.7, fill="#33AADE", color="black") +
   geom_density(aes(x=suma),alpha=0.2, fill="red")
```

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




```
# otra forma es aclararle que quiero el eje y normalizado
ggplot()+
   geom_histogram(aes(x=suma,y = (..count..)/sum(..count..)),alpha=0.7, fill="#33AADE", color="black") +
   geom_density(aes(x=suma),alpha=0.2, fill="red")
```

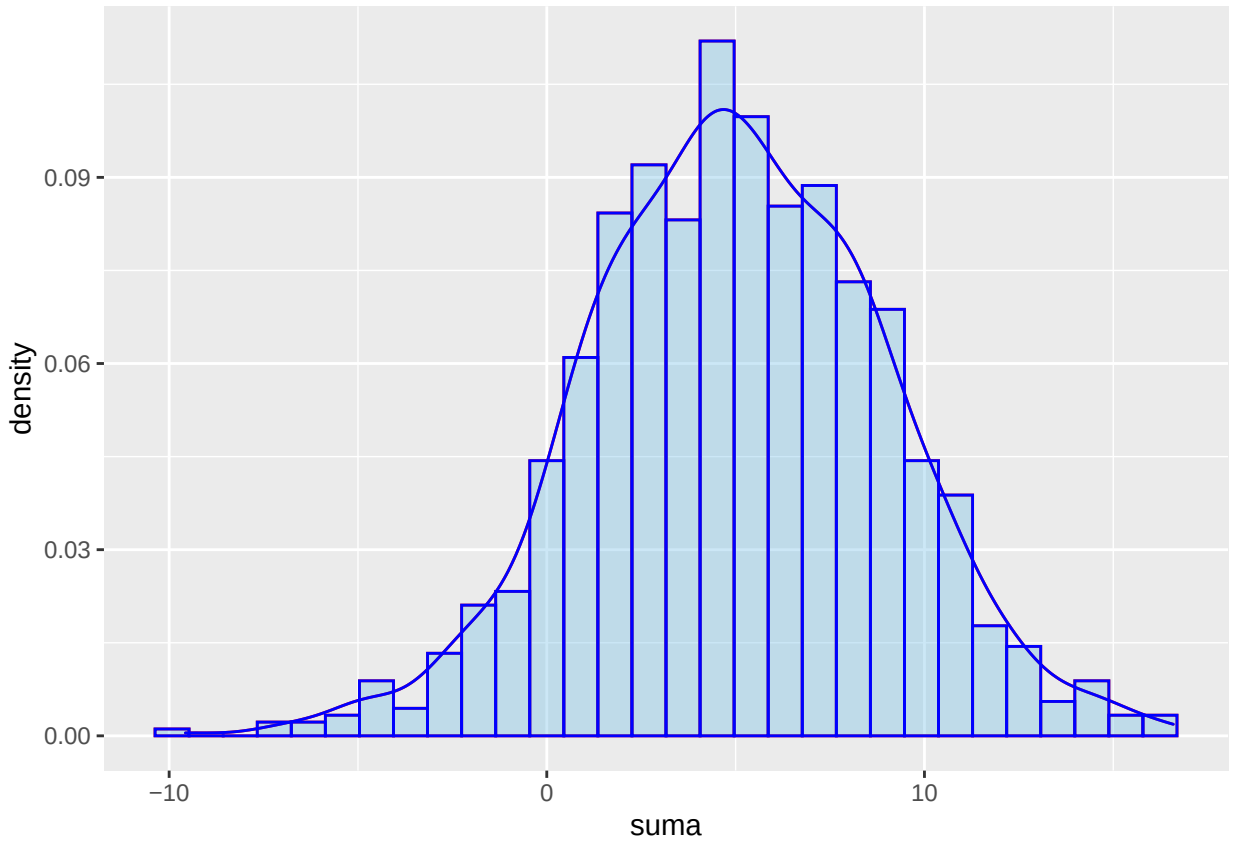
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Ahora sí grafiquemos ambos histogramas juntos.

```
data=var.gen(1,1000)
suma=apply(data,1,sum)
prom=apply(data,1,mean)
ggplot() +
   geom_histogram(aes(x=suma,y = ..density..), alpha=0.2, fill="pink",
                  col='red',position = 'identity') +
   geom_density(aes(x=suma),alpha=0.2, col='red')+
   geom_histogram(aes(x=prom,y = ..density..), alpha=0.2, fill="DeepSkyBlue1",
                  col='blue',position = 'identity')+
   geom_density(aes(x=prom),alpha=0.2, col='blue')
```

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

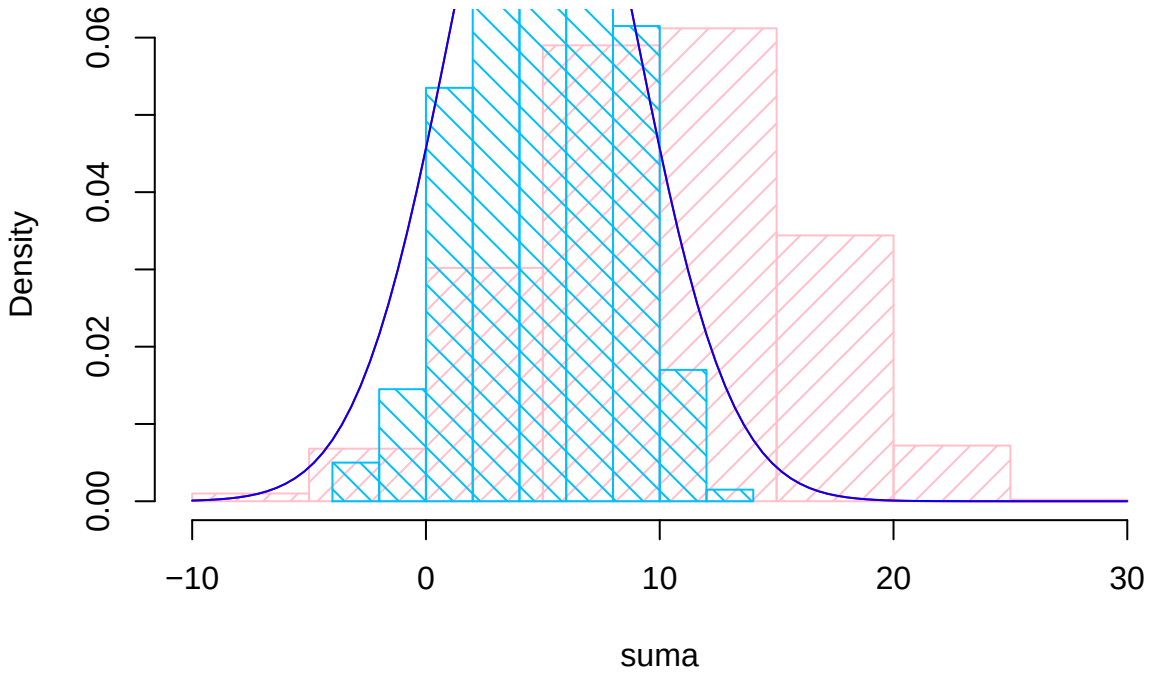


b) Consideramos n = 2 variables aleatorias X 1 y X 2 independientes con distribución N (5, 4), la suma y el promedio de ambas, es decir, S 2 = X 1 + X 2 y $\bar{X}2$ = (X 1 + X 2)/2. Generamos n = 2 datos (independientes) correspondientes a una variable aleatoria con distribución N (5, 4) y computamos suma y promedio. Replicamos N rep = 1000 veces y realizamos los histogramas de S n y X n a partir de los N rep = 1000 sumas y promedios obtenidos. ¿Qué características tienen estos histogramas? Superponerle a cada histograma la densidad que te parezca adecuada.

Resolución:

Con hist.

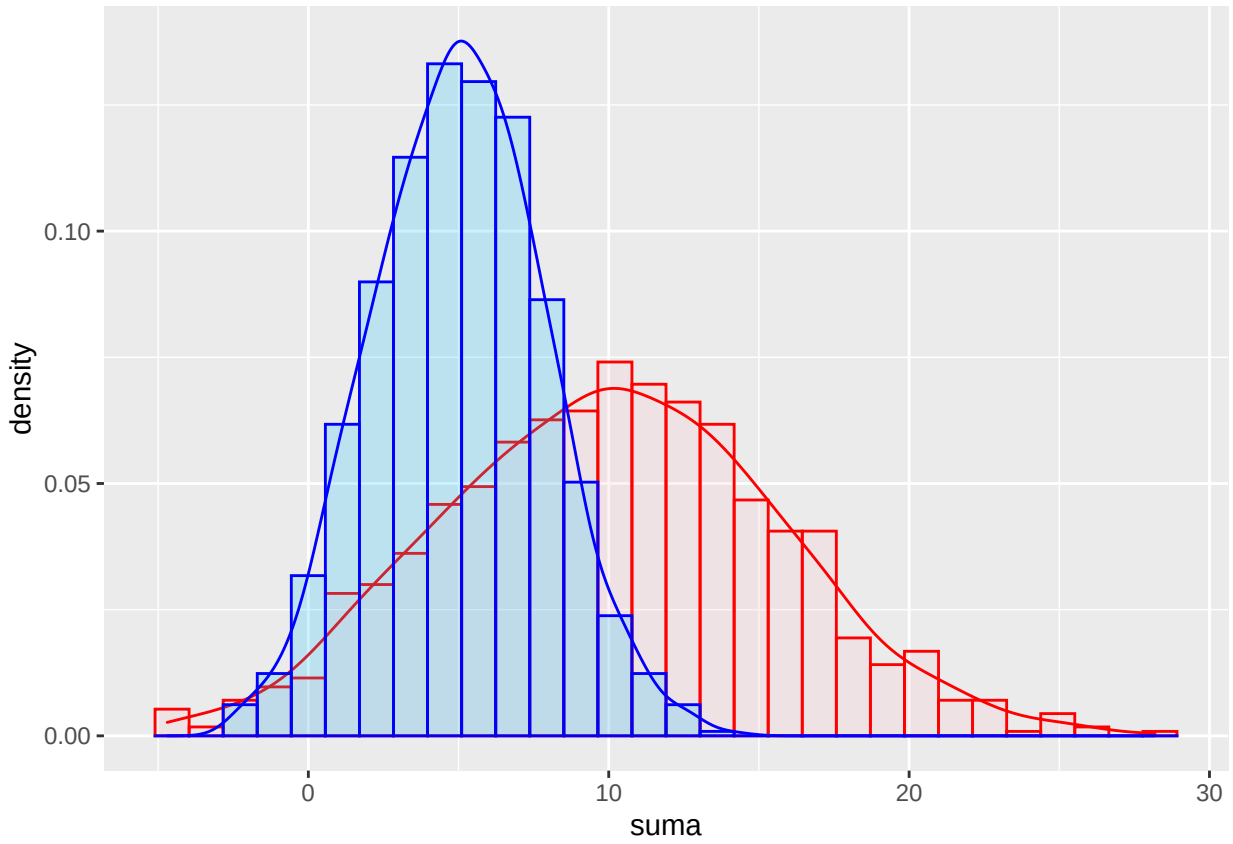
```
data=var.gen(2,1000)
# como los experimentos son las filas suma y producto es
# sumar y promediar cada fila y por eso apply(-,1,-)
suma=apply(data,1,sum)
prom=apply(data,1,mean)
hist(suma,freq=F,col='pink',main='Suma y promedio',
     density=10,angle=45)
curve(dnorm(x,5,4),add=T,col='red')
hist(prom,freq=F,col='DeepSkyBlue1',main='Promedio',add=T,
     density=10,angle=135)
curve(dnorm(x,5,4),add=T,col='blue')
```



Con ggplot.

```
data=var.gen(2,1000)
suma=apply(data,1,sum)
prom=apply(data,1,mean)
ggplot() +
   geom_histogram(aes(x=suma,y = ..density..), alpha=0.2, fill="pink",
                  col='red',position = 'identity') +
   geom_density(aes(x=suma),alpha=0.2, col='red')+
   geom_histogram(aes(x=prom,y = ..density..), alpha=0.2, fill="DeepSkyBlue1",
                  col='blue',position = 'identity')+
   geom_density(aes(x=prom),alpha=0.2, col='blue')
```

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



c) Repetir el ítem anterior usando ahora n = 5, 10, 25.

Resolución:

Graficaremos todos los casos de suma en un mismo plot y todos los casos de promedio en otro plot. Y despues haremos un gif para cada caso que muestre la transición a medida que tomamos mas variables, es decir a medida que n se mueve en c(1,2,5,10,25).

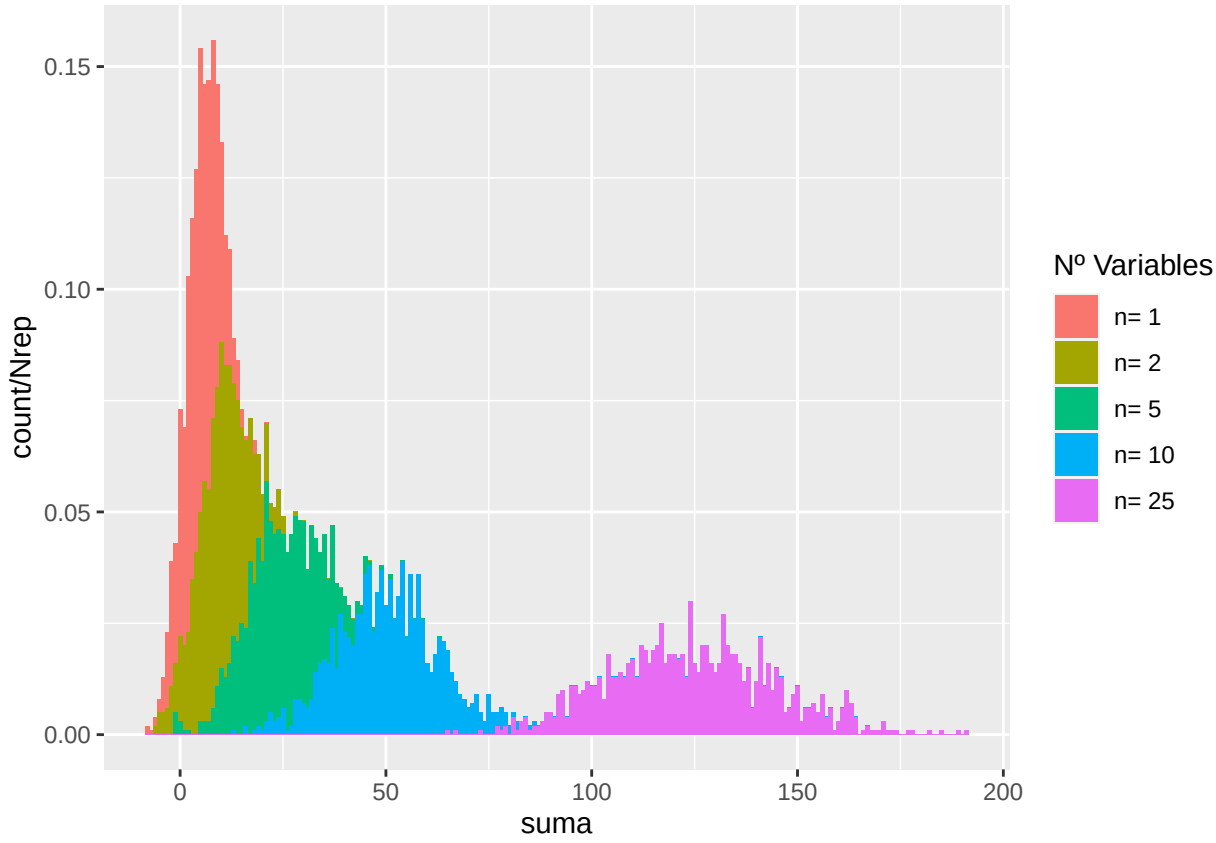
Primero organizo los datos en dos data.frames verticales.

```
enes=c(1,2,5,10,25)
Nrep=1000
sumyprod=function(n,Nrep,modo){
   data=var.gen(n,Nrep)
   if(modo==0){
      data.frame('suma'=apply(data,1,sum))
   }else if(modo==1){
      data.frame('promedio'=apply(data,1,mean))
   }else{ 'Ingrese modo=0 para obtener las sumas o modo=1 para obtener los promedios'}
}
# creo un data.frame con sum y prod de los enes casos
datos_suma=as.data.frame(lapply(enes,sumyprod,Nrep,modo=0))
datos_promedio=as.data.frame(lapply(enes,sumyprod,Nrep,modo=1))
# transformo los datos en un data.frame de 2 columnas donde
# col 1 = datos, col 2 = names
datos_suma=stack(datos_suma)[1]
colnames(datos_suma)='suma'
datos_promedio=stack(datos_promedio)[1]
colnames(datos_promedio)='promedio'
```

```r
e<-vector()
# Ahora le agrego otra col a los datos con numeros en 1:length(enes)
# que van a representar los distintos experimentos para cada valor de ene.
for (i in 1:length(enes)){
 e<-c(e,rep(i,Nrep))
}
datos_suma$ene=datos_promedio$ene=e
# Todo esto es para tener una categoría en la cual moverme para hacer el gif
# i.e. transition...(ene)
```

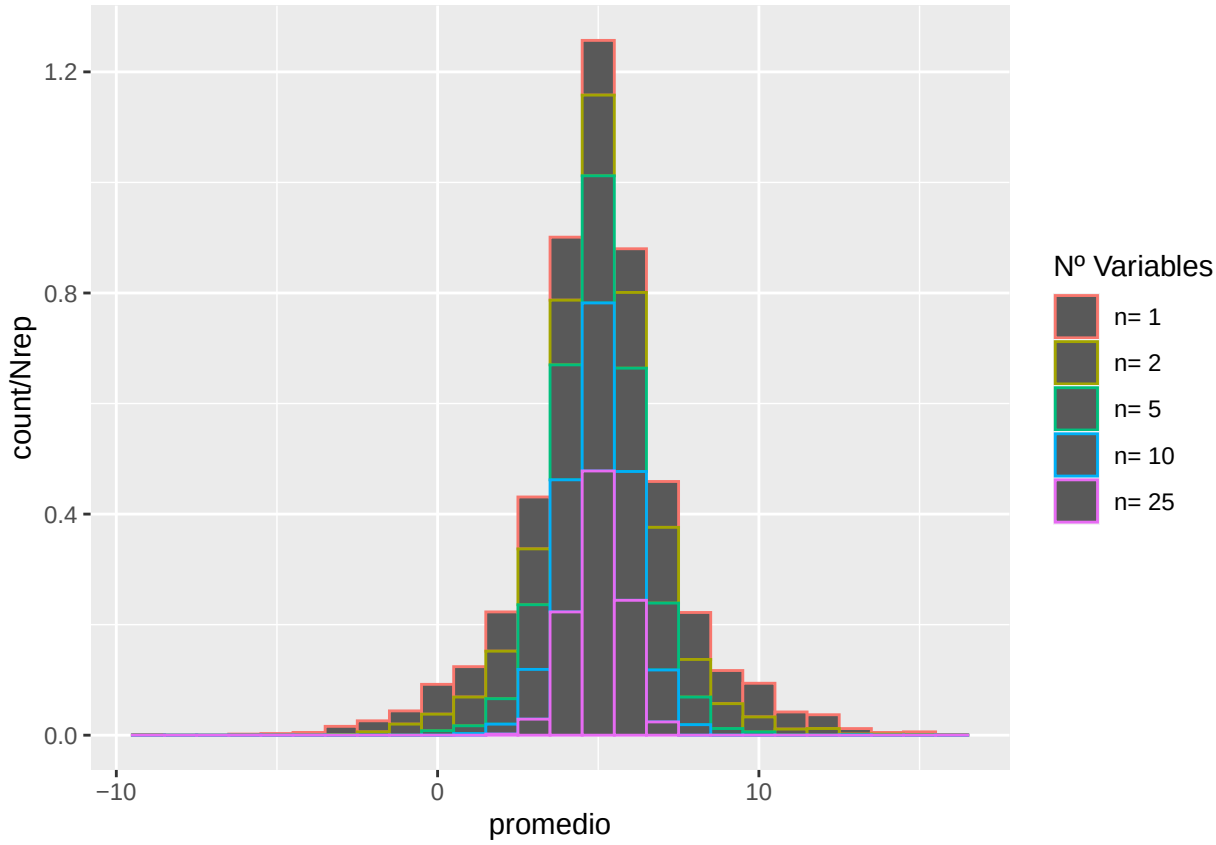
Grafiquemos ambos en dos plots.

```r
library(tidyr)
# tengo que cargar este paquete para poder usar %>%
datos_suma %>%
   ggplot(aes(fill=as.factor(ene)))+
   geom_histogram(aes(x=suma,y=stat(count)/Nrep),
                  binwidth=1 )+
   scale_fill_discrete(name="Nº Variables",
                       breaks=1:5,
                       labels=paste("n=", enes))
```




```r
datos_promedio %>%
   ggplot(aes(color=as.factor(ene)))+
   geom_histogram(aes(x=promedio,y=stat(count)/Nrep),
                  binwidth=1 )+
   scale_color_discrete(name="Nº Variables",
                       breaks=1:5,
```

```
                    labels=paste("n=", enes))
```



Dejo el programa para el gif.

```
library(gganimate)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
# dplyr provee filter()
# datos_suma$ene=as.integer(datos_suma$ene)
# datos_suma$ene=as.character(datos_suma$ene)
# datos_suma$ene=as.factor(datos_suma$ene)
anim_sum <- datos_suma %>%
   filter(ene<3) %>%
   ggplot(aes(color=ene))+
   # Notar que obtenemos distintas cosas si ponemos
   # fill=ene o fill=as.factor(ene)
   # (o color=ene o color=as.factor(ene))
   # si lo tomamos como factor muestra colores
```

```
   # pero no dibuja la transición sino cada estado
   geom_histogram(aes(x=suma,y=stat(count)/Nrep),
                  binwidth=1 )+
   # para que sea suave supuesatmente tengo que agregar
   # aes(...,group=suma) pero r consume toda la ram
   transition_states(ene, transition_length = 1, state_length = 1)+
    labs(title = "n= {closest_state}")
   # closest_state es la variable que va moviendo transition_states
   # en nuestro caso mueve ene
# Con transition_states le decimos qué parámetro ir variando
# y con transition_length = 1, state_length = 1 le decimos que varíe 1
# animate(anim_sum,
#         width = 400, height = 400,
#         nframes = 480, fps = 24)
# anim_save("suma.gif", anim_sum)
anim_prom <- datos_promedio %>%
   ggplot()+
   geom_histogram(aes(x=promedio,y=stat(count)/Nrep,color=as.factor(ene)),
                  binwidth=1 )+
   transition_states(ene, transition_length = 1, state_length = 1)
# animate(anim_prom,
#         width = 400, height = 400,
#         nframes = 480, fps = 24)
# anim_save("prom.gif", anim_prom)
```

VER COMO HAGO QUE LA TRANSICIÓN NO SEA DE A ESTADOS SINO ALGO CONTINUO PERO CON ESOS MISMOS COLORES!

d) ¿Qué se observa a medida que n crece?

Rta:

A medida que n crece observamos que la suma toma una forma acampanada (acorde a LGN) y que el promedio se concentra en la media de los datos (acorde al TCL).

MORALEJA: uno tiene una muestra de datos que son la realizacion de ciertas variables aleatorias que sabemos que son iid, ie son realizaciones de muestras aleatorias entones la distribucion la aproximamos con la frecuencia relativa. Es decir dadas $(X_i)_{i\geq 1}$ F queremos estimar $F(t) = P(X_i \leq t)$ y la idea es considerar $Y_i = I_{X_i \leq t} = \chi_{(-\infty,t]}(X_i)$, entonces Y_i $\mathcal{B}(1,p)$ con $p = E(Y_i) = P(Y_i = 1) = P(X_i \leq t) = F(t)$. Luego por LGN

$$\overline{Y_n} = \frac{1}{n}\sum Y_i \xrightarrow[n\to\infty]{} E(Y_1) = F(t).$$

Si definimos $\widehat{F}(t) = \overline{Y_n} = \frac{1}{n}\sum \chi_{(-\infty,t]}(X_i)$.

2. Generaremos muestras aleatorias de una distribución $\mathcal{U}(0,1)$, de tamaño cada vez mayor, y calcularemos los promedios muestrales. Comprobaremos si estos promedios muestrales se acercan a algún valor cuando el tamaño muestral crece y en tal caso, identicicaremos el valor límite.

a) Consideremos la distribución $\mathcal{U}(0,2)$. Indicar cuál es el valor verdadero de la media μ.

Resolución:

El valor de la media de una función uniforme $\mathcal{U}(a,b)$ sabemos que es $\frac{a+b}{2}$. Entonces la esperanza en nuestro caso es 1.

b) Generamos una muestra de tamaño N rep = 1000 y para cada n entre 1 y 1000 calculemos el promedio de las primeras n observaciones. Realizamos un scatterplot de n vs. los promedios obtenidos. Incluimos una linea horizontal en el valor de y correspondiente a la verdadera media μ.

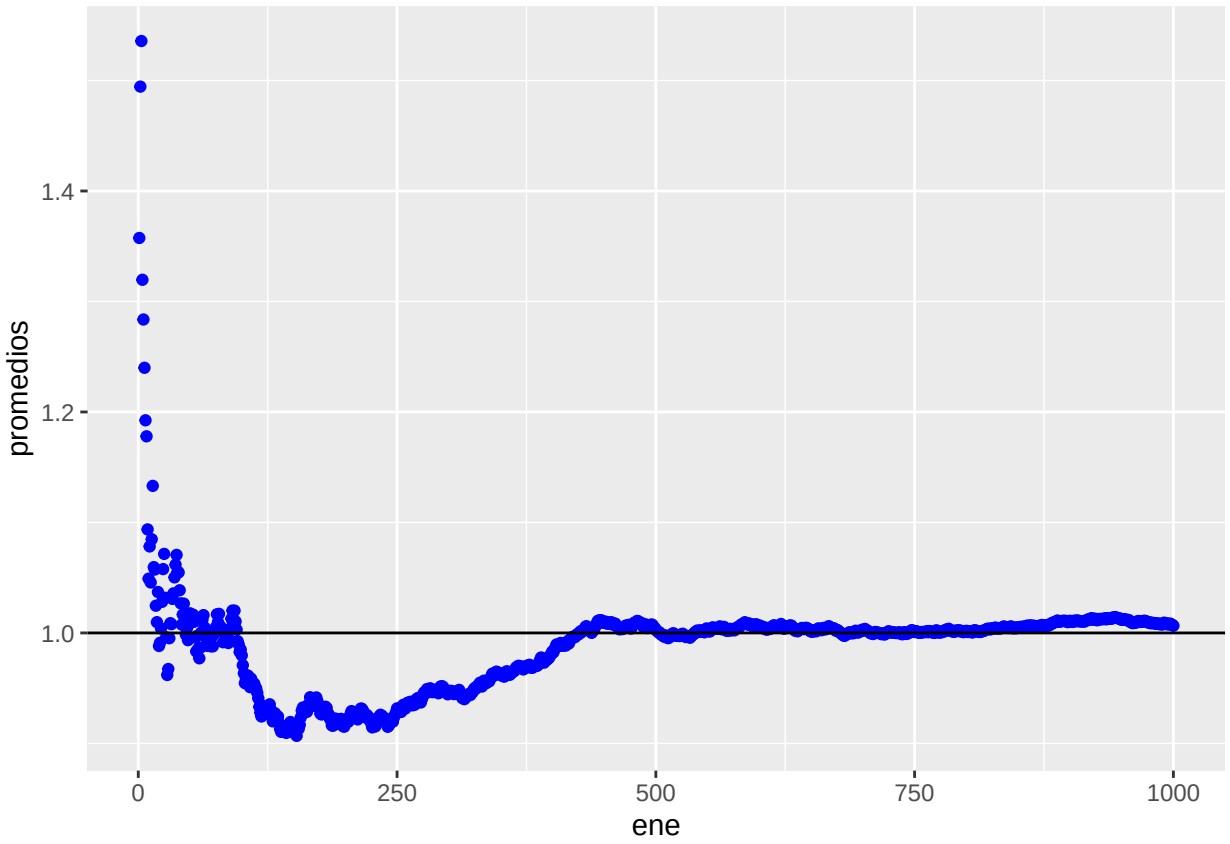
Resolución:

Organizamos la informacion en un dataframe vertical.

```
muestra=runif(1000,0,2)
ene=1000
promedios=c()
for(i in 1:ene){
   promedios=c(promedios,mean(muestra[1:i]))
}
promedios=as.data.frame(promedios)
promedios$ene=1:1000
```

Ploteamos.

```
promedios %>%
   ggplot()+
   geom_point(aes(x=ene,y=promedios),
                  colour ='blue')+
   geom_hline(aes(yintercept=1), col='black')
```



Hagamos el gif.

```
anim_unif=promedios %>%
   ggplot()+
   geom_point(aes(x=ene,y=promedios, group = seq_along(ene)),
                  colour ='blue')+
   geom_hline(aes(yintercept=1), col='black')+
   transition_reveal(ene)
# para que se vea el trazo de lo que va dibujando en la transición usamos
# transition_reveal junto con aes(...,group = seq_along(ene))
# si queremos mostrar cada estado usamos simplemente transition_states.
# animate(anim_unif,
#         width = 400, height = 400,
#         nframes = 480, fps = 24)
```

c) Repetir comenzando con otra semilla y superponer el nuevo gráfico utilizando un color diferente. Comparars los gráficos obtenidos. ¿Qué se puede concluir?

Resolución:

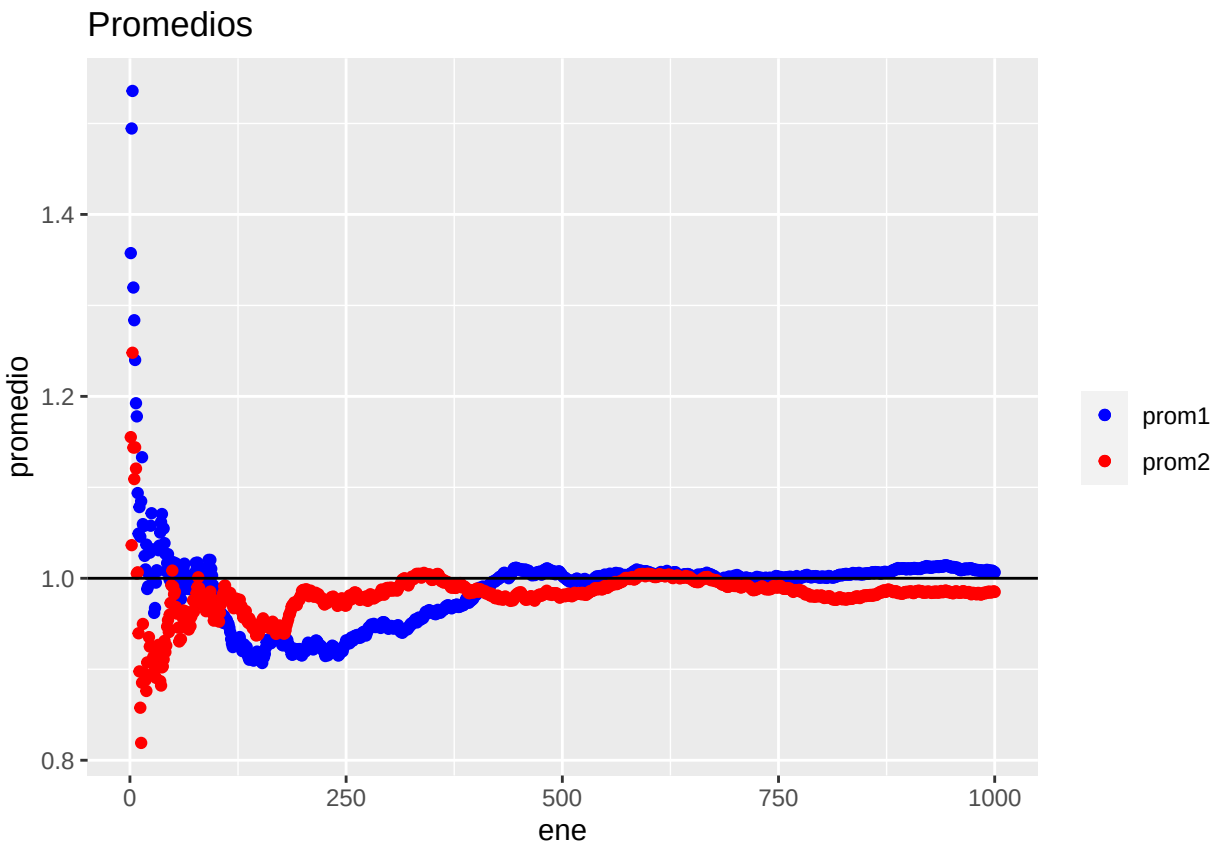
Generamos otros datos.

```
set.seed(8888)
muestra_2=runif(1000,0,2)
promedios_2=c()
for(i in 1:ene){
   promedios_2=c(promedios_2,mean(muestra_2[1:i]))
}
promedios_2=as.data.frame(promedios_2)
```

```
promedios_2$ene=1:1000
```

Ploteamos.

```
prom_total=as.data.frame(cbind('promedios'=promedios$promedios,'promedios_2'=promedios_2$promedios_2))
prom_total$ene=1:1000
prom_total %>%
   ggplot()+
   geom_point(aes(x=ene,y=promedios,
                  colour ='prom1'))+
   geom_point(aes(x=ene,y=promedios_2,
                  colour ='prom2'))+
   geom_hline(aes(yintercept=1), col='black')+
   scale_colour_manual("",
                       breaks = c("prom1", "prom2"),
                       values = c("blue", "red")) +
  xlab("ene") +
  scale_y_continuous("promedio") +
  labs(title="Promedios")
```



Hagamos el gif.

```
anim_total= prom_total %>%
   ggplot()+
   geom_point(aes(x=ene,y=promedios,group = seq_along(ene),
                  colour ='prom1'))+
   geom_point(aes(x=ene,y=promedios_2,group = seq_along(ene),
                  colour ='prom2'))+
```

```
   geom_hline(aes(yintercept=1), col='black')+
   transition_reveal(ene)+
   scale_colour_manual("",
                       breaks = c("prom1", "prom2"),
                       values = c("blue", "red")) +
  xlab("ene") +
  scale_y_continuous("promedio") +
  labs(title="Promedios")
# animate(anim_total,
#         width = 400, height = 400,
#         nframes = 480, fps = 24)
# anim_save("unif prom.gif", anim_total)
```