

Función normal segunda parte

Chaparro, Analía G.

2023-09-19

Valores aleatorios de una distribución normal

Supongamos que tenemos una distribución normal con media 0 y desvío 1. La variable aleatoria, toma infinitos valores entre $(-\infty; \infty)$.

Vamos a tomar 10 valores de las variables aleatoria en forma aleatoria.

```
## rnorm(10, 0, 1)

## [1] -1.30581678 -1.04049827 -1.62372189 0.06015882 -1.95006594 0.95823976
## [7] 0.15504951 1.20741650 -1.24202005 -0.37144765

n <- 10 #tamaño de la muestra
muestra <- rnorm(n, 0, 1)
muestra

## [1] -0.9344891 -0.4363571 2.6268225 0.4512500 0.9270267 -0.4003894
## [7] -0.2302232 -1.6729304 0.7862206 -1.2727375

# Convertimos la muestra en un dataframe
muestra <- data.frame(muestra)

muestra <- muestra %>%
  arrange(muestra)
muestra

##      muestra
## 1 -1.6729304
## 2 -1.2727375
## 3 -0.9344891
## 4 -0.4363571
## 5 -0.4003894
## 6 -0.2302232
## 7 0.4512500
## 8 0.7862206
## 9 0.9270267
## 10 2.6268225
```

Para cada valor en la muestra vamos a calcular su probabilidad con pnorm.

Para ello, al dataframe le agregamos una columna que se llame probabilidad. Sugerencia, utilizar la función mutate().

```
data <- muestra %>%
  mutate(Probabilidad = pnorm(muestra, 0, 1))
data

##      muestra Probabilidad
## 1 -1.6729304 0.0471705
## 2 -1.2727375 0.1015556
## 3 -0.9344891 0.1750258
## 4 -0.4363571 0.3312888
## 5 -0.4003894 0.3444349
## 6 -0.2302232 0.4089592
## 7 0.4512500 0.6740953
## 8 0.7862206 0.7841309
## 9 0.9270267 0.8230437
## 10 2.6268225 0.9956907
```

Calculamos la densidad para cada punto y se lo agregamos al dataframe

```
data <- data %>%
  mutate(Densidad = dnorm(muestra, 0, 1))
data

##      muestra Probabilidad Densidad
## 1 -1.6729304 0.0471705 0.09844212
## 2 -1.2727375 0.1015556 0.17748506
## 3 -0.9344891 0.1750258 0.25779941
## 4 -0.4363571 0.3312888 0.36271340
## 5 -0.4003894 0.3444349 0.36821275
## 6 -0.2302232 0.4089592 0.38850863
## 7 0.4512500 0.6740953 0.36032393
## 8 0.7862206 0.7841309 0.29287483
## 9 0.9270267 0.8230437 0.25959623
## 10 2.6268225 0.9956907 0.01266343
```

Mostramos en tabla el dataframe

```
data %>%
  kbl() %>%
  kable_styling(full_width = F)
```

	muestra	Probabilidad	Densidad
	-1.6729304	0.0471705	0.0984421
	-1.2727375	0.1015556	0.1774851
	-0.9344891	0.1750258	0.2577994
	-0.4363571	0.3312888	0.3627134
	-0.4003894	0.3444349	0.3682127
	-0.2302232	0.4089592	0.3885086
	0.4512500	0.6740953	0.3603239
	0.7862206	0.7841309	0.2928748
	0.9270267	0.8230437	0.2595962
	2.6268225	0.9956907	0.0126634

Calculamos el desvío y la media de la muestra

```
x_aya <- mean(data$muestra)
s_d <- sd(data$muestra)

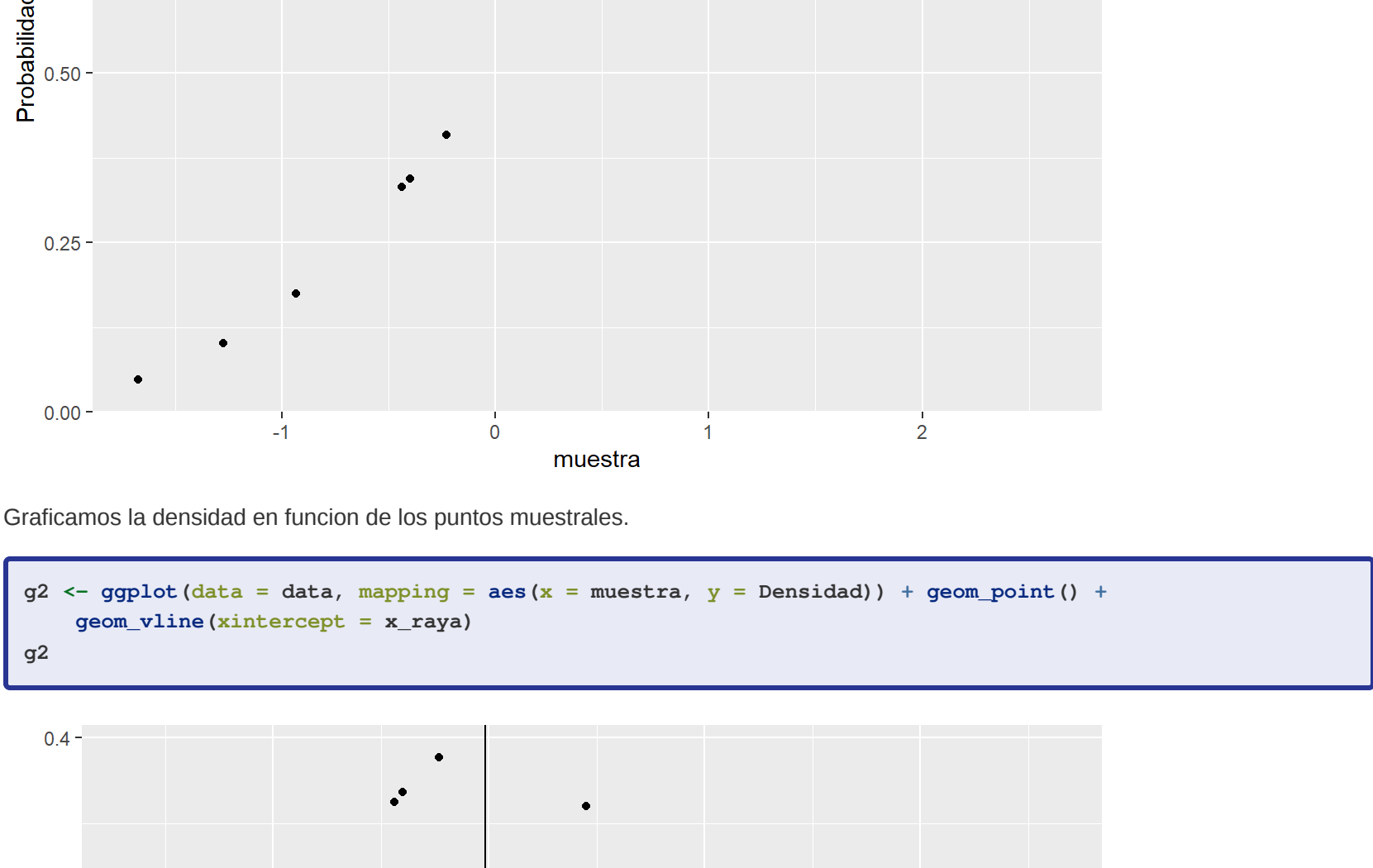
Lo presentamos en un cuadro de salida
```

```
resumen_data <- data.frame(Tamaño_muestral = n, Media_muestral = x_aya, Desvio_muestral = s_d)
resumen_data <- resumen_data %>%
  mutate(Numero_muestra = nrow(resumen_data))
# reordeno
resumen_data <- resumen_data %>%
  select(Numero_muestra, Tamaño_muestral, Media_muestral, Desvio_muestral)
resumen_data %>%
  kbl() %>%
  kable_styling(full_width = F)
```

Numero_muestra	Tamaño_muestral	Media_muestral	Desvio_muestral
1	10	-0.0155807	1.259385

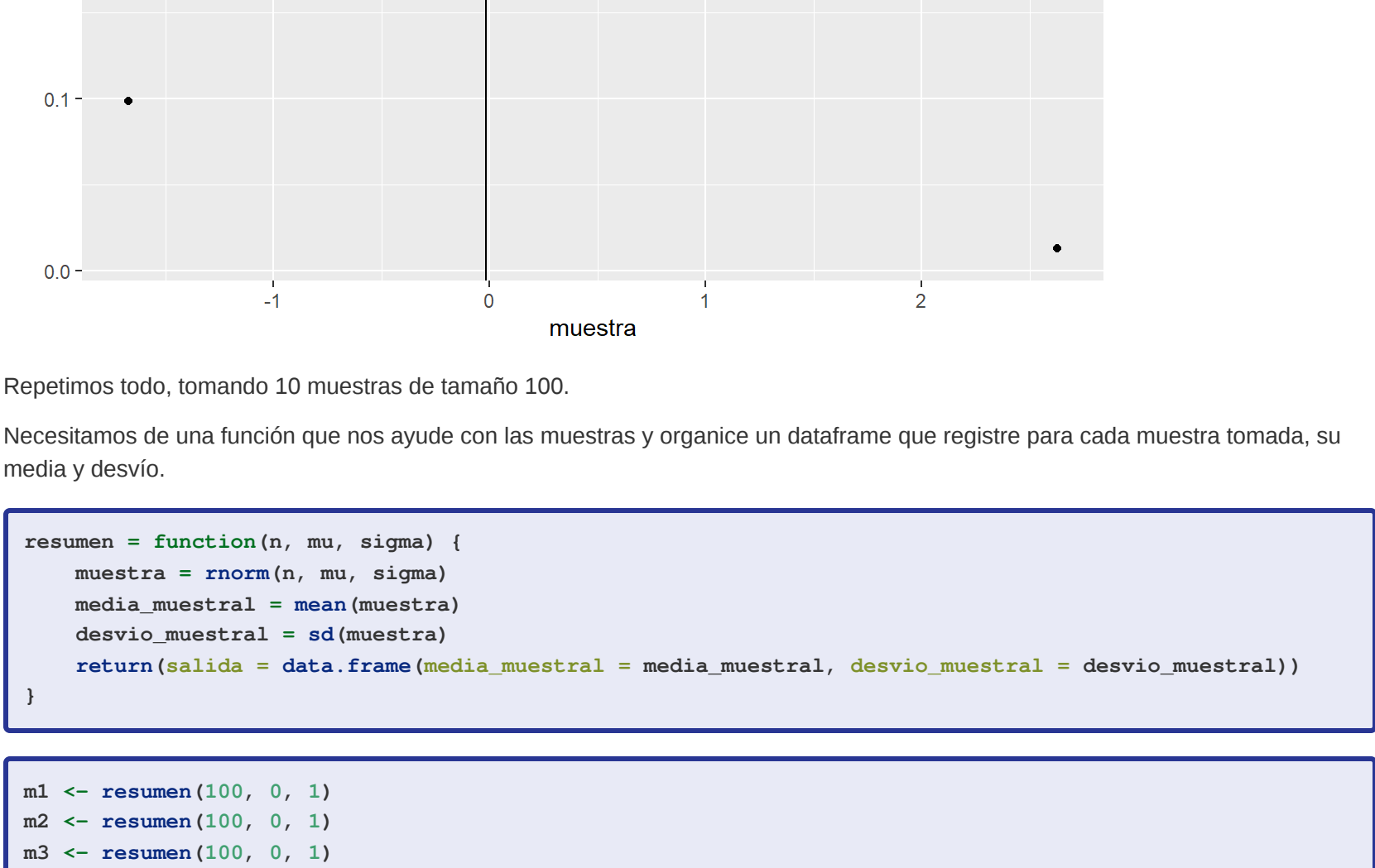
Grificamos la probabilidad vs los cuantiles

```
g1 <- ggplot(data = data, mapping = aes(x = muestra, y = Probabilidad)) + geom_point() +
  geom_hline(yintercept = 1)
g1
```



Grificamos la densidad en funcion de los puntos muestrales.

```
g2 <- ggplot(data = data, mapping = aes(x = muestra, y = Densidad)) + geom_point() +
  geom_vline(xintercept = x_aya)
g2
```



Repetimos todo, tomando 10 muestras de tamaño 100.

Necesitamos de una función que nos ayude con las muestras y organice un dataframe que registre para cada muestra tomada, su media y desvío.

```
resumen = function(n, mu, sigma) {
  muestra = rnorm(n, mu, sigma)
  media_muestral = mean(muestra)
  desvio_muestral = sd(muestra)
  return(salida = data.frame(media_muestral = media_muestral, desvio_muestral = desvio_muestral))
}

m1 <- resumen(100, 0, 1)
m2 <- resumen(100, 0, 1)
m3 <- resumen(100, 0, 1)
m4 <- resumen(100, 0, 1)
m5 <- resumen(100, 0, 1)
m6 <- resumen(100, 0, 1)
m7 <- resumen(100, 0, 1)
m8 <- resumen(100, 0, 1)
m9 <- resumen(100, 0, 1)
m10 <- resumen(100, 0, 1)
res <- rbind(m1, m2, m3, m4, m5, m6, m7, m8, m9, m10)
res <- res %>%
  mutate(muestra_numero = 1:nrow(res)) %>%
  dplyr::arrange(media_muestral)

res %>%
  kbl() %>%
  kable_styling(full_width = F)
```

media_muestral	desvio_muestral	muestra_numero
-0.1258346	1.1142442	9
-0.0926124	1.0379415	7
-0.0642554	0.9879867	6
-0.0577403	1.0097423	4
-0.0341990	0.8576478	8
-0.0153370	1.0696174	10
-0.0127470	1.0217628	2
0.0123323	1.0971268	5
0.0327007	0.9185687	3
0.1691035	0.9320647	1

Grificamos un histograma de las medias...

```
g <- ggplot(data = res, mapping = aes(x = media_muestral)) + geom_histogram(aes(y = ..density..),
  colour = 1, fill = "lightblue") + geom_density(lwd = 1, colour = 4)
g
```

Warning: The dot-dot notation ('..density..') was deprecated in ggplot2 3.4.0.
i Please use 'after_stat(density)' instead.
This warning is displayed once every 8 hours.
Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was generated.



Qué pasa si queremos repetir este proceso 100 veces? tomar 100 muestras de tamaño 10 o de tamaño 100 o de tamaño 1000?

Simulación de una variable aleatoria especial

Definamos una función que haga esta simulación...

```
# Definimos la función que realiza las iteraciones
simulacion <- function(I, n, mu, sigma) {
  # En primer lugar, definimos la función resumen, dentro de la función
  # generalizadora

  resumen = function(n, mu, sigma) {
    muestra = rnorm(n, mu, sigma)
    media_muestral = mean(muestra)
    desvio_muestral = sd(muestra)
    return(salida = data.frame(media_muestral = media_muestral, desvio_muestral = desvio_muestral))
  }

  # Creamos un Dataframe vacío para almacenar los resultados
  resultado <- data.frame()
  # Creamos un ciclo for que en cada iteración, genere una muestra y calcule
  for (i in 1:I) {
    # Llamamos a la función resumen y obtenemos el dataframe en cada
    # iteracion
    df <- resumen(n, mu, sigma)
    # Unificamos el Dataframe actual con el dataframe vacío llamado
    # resultado
    resultado <- rbind(resultado, df)
  }

  # Reiniciamos los índices del Dataframe resultado
  rownames(resultado) <- NULL
  # Agregamos una columna con el tamaño muestral
  resultado <- resultado %>%
    mutate(Muestreo_numero = 1:nrow(resultado))
  # Reordenamos las columnas del dataframe
  resultado <- resultado %>%
    select(Muestreo_numero, media_muestral, desvio_muestral)

  return(resultado)
}
```

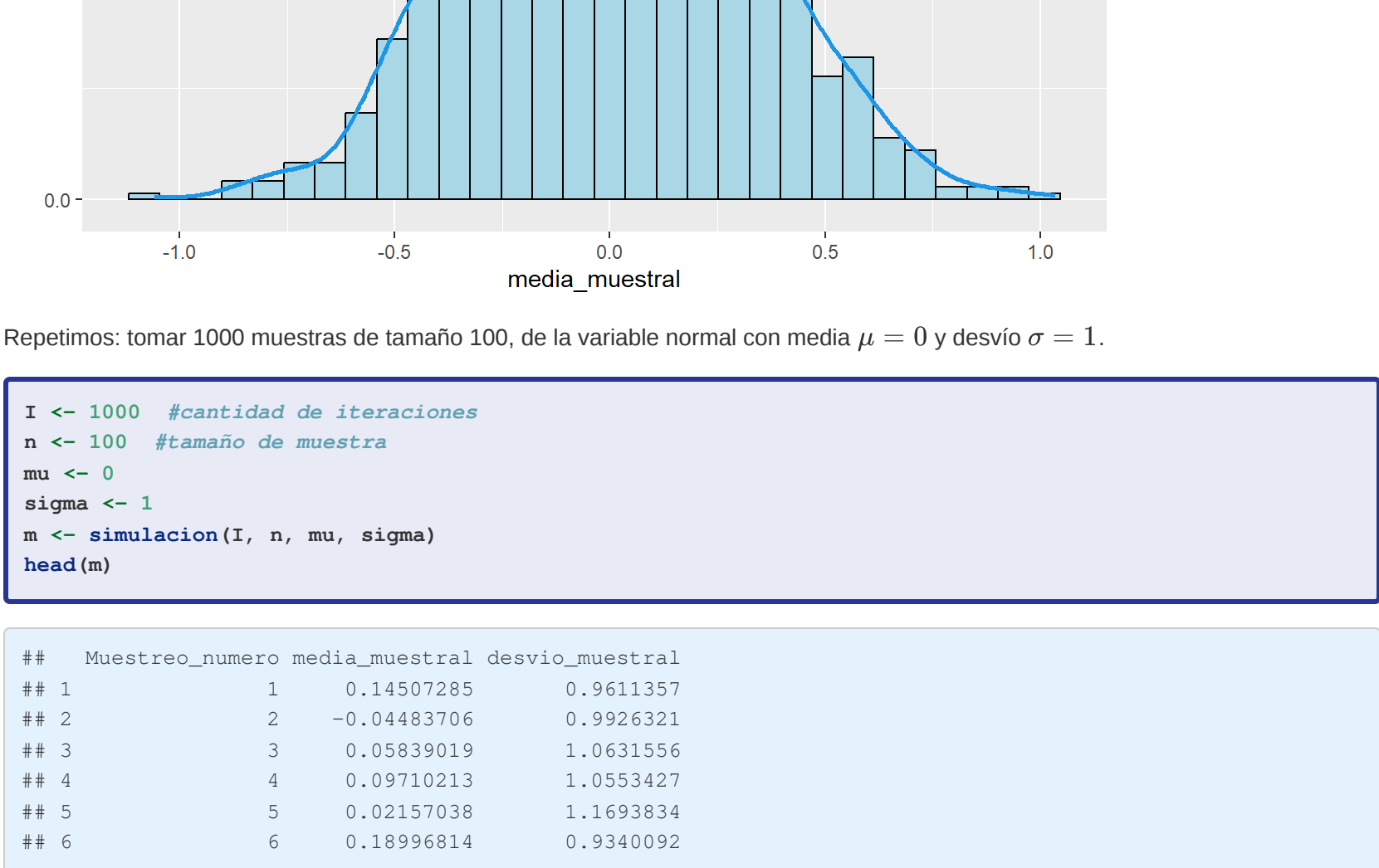
Ahora aplicamos la función generalizadora para tomar 1000 muestras de tamaño 10, de la variable normal con media $\mu = 0$ y desvío $\sigma = 1$

```
I <- 1000 #cantidad de iteraciones
n <- 10 #tamaño de muestra
mu <- 0
sigma <- 1
m <- simulacion(I, n, mu, sigma)
head(m)

##      Muestreo_numero media_muestral desvio_muestral
## 1          1 0.056139761 1.1373697
## 2          2 -0.004132874 0.5748030
## 3          3 -0.082572188 1.0604397
## 4          4 -0.160055668 0.7700694
## 5          5 -0.272317758 0.9570052
## 6          6 0.177191262 0.7704807
```

Grificamos un histograma de las medias...

```
g <- ggplot(data = m, mapping = aes(x = media_muestral)) + geom_histogram(aes(y = ..density..),
  colour = 1, fill = "lightblue") + geom_density(lwd = 1, colour = 4)
g
```



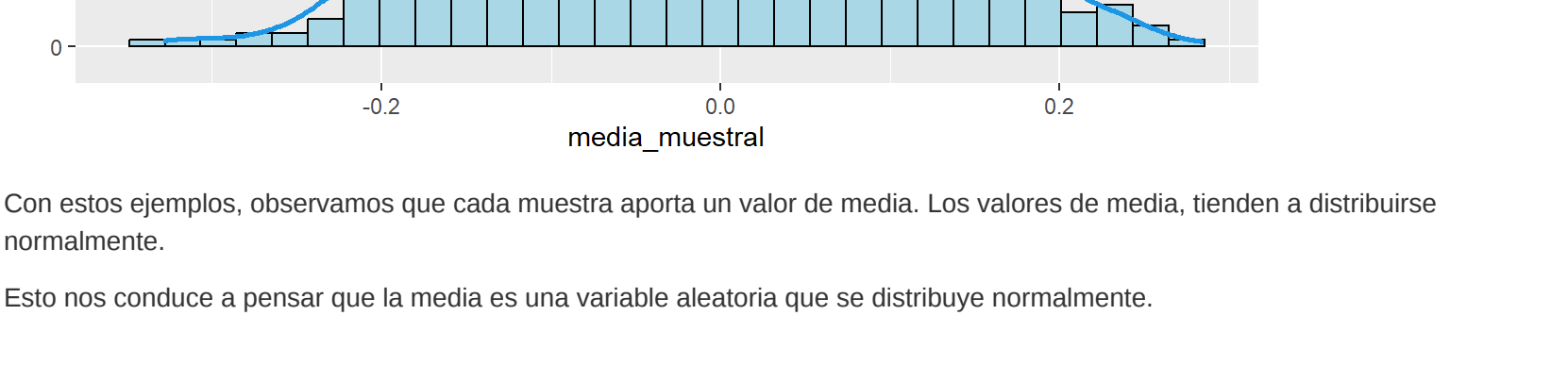
Repetimos: tomar 1000 muestras de tamaño 100, de la variable normal con media $\mu = 0$ y desvío $\sigma = 1$.

```
I <- 1000 #cantidad de iteraciones
n <- 100 #tamaño de muestra
mu <- 0
sigma <- 1
m <- simulacion(I, n, mu, sigma)
head(m)

##      Muestreo_numero media_muestral desvio_muestral
## 1          1 0.14507285 0.9611357
## 2          2 -0.04483706 0.9926321
## 3          3 0.05839019 1.0631556
## 4          4 0.09710213 1.0553427
## 5          5 0.02157038 1.1693834
## 6          6 0.18996814 0.9340092
```

Grificamos un histograma de las medias

```
g <- ggplot(data = m, mapping = aes(x = media_muestral)) + geom_histogram(aes(y = ..density..),
  colour = 1, fill = "lightblue") + geom_density(lwd = 1, colour = 4)
g
```



Con estos ejemplos, observamos que cada muestra aporta un valor de media. Los valores de media, tienden a distribuirse normalmente.

Esto nos conduce a pensar que la media es una variable aleatoria que se distribuye normalmente.