SESSION 4 SUMMARY

Victor Miguel Terrón Macias

21/1/2021

SESSION 4. ALGUNAS DISTRIBUCIONES, TEOREMA CENTRAL DEL LÍMITE Y CONTRASTE DE HIPOTESIS

CONCEPTOS DE ESTADÍSTICA INFERENCIAL BÁSICOS

DISTRIBUCIÓN BINOMIAL

Se define como un experimento con las siguientes características: * Consiste en un número fijo, n, de pruebas idénticas. * Cada prueba resulta en uno de dos resultados: éxito S o fracaso F * La probabilidad de éxito en una sola prueba es igual a algún valor p y es la misma de una prueba a otra. La probabilidad de fracaso es igual a $q = 1 - p$ * Las pruebas son independientes * La variable aleatoria (v.a.discreta) de interés es Y, el número de éxitos observado durante las n pruebas.

Se dice que una variable aleatoria Y tiene una distribución binomial basada en n pruebas con probabilidad p de éxito, si y solo si.

$$P_x = \binom{n}{x} p^x q^{m-x}$$

De donde P es probabilidad binomial, de donde x es el numero de veces para obtener un resultado específico en n ensayos, de donde $\binom{n}{x}$, de donde p es la probabilidad de exito en un solo ensayo,q es probabilidad de fallo en un solo ensayo y n es el numero de ensayos. Y donde P que es la posibilidad tiene que cumplir con $0 \leq P \leq 1$

EJEMPLO DE APLICACION DE DISTRIBUCION BINOMIAL

La última novela de un autor ha tenido un gran éxito, hasta el punto de que el 80% de los lectores ya la han leído. Hallar la probabilidad de que en un grupo de 4 amigos que son aficionados a la lectura, 2 hayan leído la novela.

1. Hallar la probabilidad de que una persona haya leído el libro es de 0.8, por lo que la probabilidad de que no lo haya leído es de 0.2. De donde tenemos los siguientes datos: $n = 4, k = 2, p = 0.8, q = 0.2$
2. Hallar la probabilidad de que máximo 2 personas del grupo de 4 amigos hayan leído la novela: tenemos los siguientes datos:

$$P(x \leq 2) = P(x = 0) + P(x = 1) + P(x = 2)$$

Sustituyendo los datos en la fomrula de distribución binomial tenemos lo siguiente:

$$P_{x\leq 2} = \binom{4}{0}(0.8)^0(0.2)^{4-0} + \binom{4}{1}(0.8)^1(0.2)^{4-1} + \binom{4}{2}(0.8)^2(0.2)^{4-2} = 0.1808$$

DISTRIBUCIÓN NORMAL

A una distribución normal se le conoce como distribución Gaussiana o distribucion de Laplace Gauss. Se utiliza csolamente con variables continuas (variables que pueden tomar un numero infinito de valores entre dos valores cualesquiera de una caracteristica).

Su gráfica es en forma de campana y simétrica respecto de un determinado parametro estadístico. Éste tipo de distribución propicia el modelado de numerosos fenómenos naturales, sociales y psicológicos. El centro de la campana es el promedio. La desviación estandar nos indican que tan dispersos o separados están los datos conr especto la media.

Para calcular la probabilidad normal se divide la cantidad de casos favorables entre la cantidad de datos totales, para ello se tuliza la siguiente fórmula:

$$z = \frac{x - \mu}{\sigma}$$

De donde tenemos que x es el valor de la condición y μ es el promedio también conocido como media y σ es la desviación estandar. Manualmente podríamos calcularlo con las tablas de distribución normal.

Las funciones de densidad de probabilidad (de variables aleatorias continuas) cumplen con las siguientes propiedades:

- El área bajo la curva de la función de densidad de probabilidad es igual a 1
- La probabilidad de que X se encuentre en determinado intervalo (a, b) es igual al área bajo la curva entre los puntos a y b.
- $P(X = c) = 0$ para cualquier valor c para el que se encuentre definida la función de densidad.

La funcion de densidad de probabilidad de una variable aleatoria x que se distribuye como normal con media μ y desviacion estandar σ es:

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

DISTRIBUCIÓN t DE STUDENT

La función de densidad t de Student se define como:

En el caso de la distribución t la media $\mu = 0$ y

$$\sigma^2 = \frac{v}{(v-2)}$$

para $v > 2$ respectivamente.

La apariencia general de la distribución t es similar a la de la distribución normal estándar: ambas son simétricas y unimodales, y el valor máximo de la ordenada se alcanza en la media $\mu = 0$. Sin embargo, la distribución t tiene colas más amplias que la normal; esto es, la probabilidad de las colas es mayor que en la distribución normal. A medida que el número de grados de libertad tiende a infinito, la forma límite de la distribución t es la distribución normal estándar.

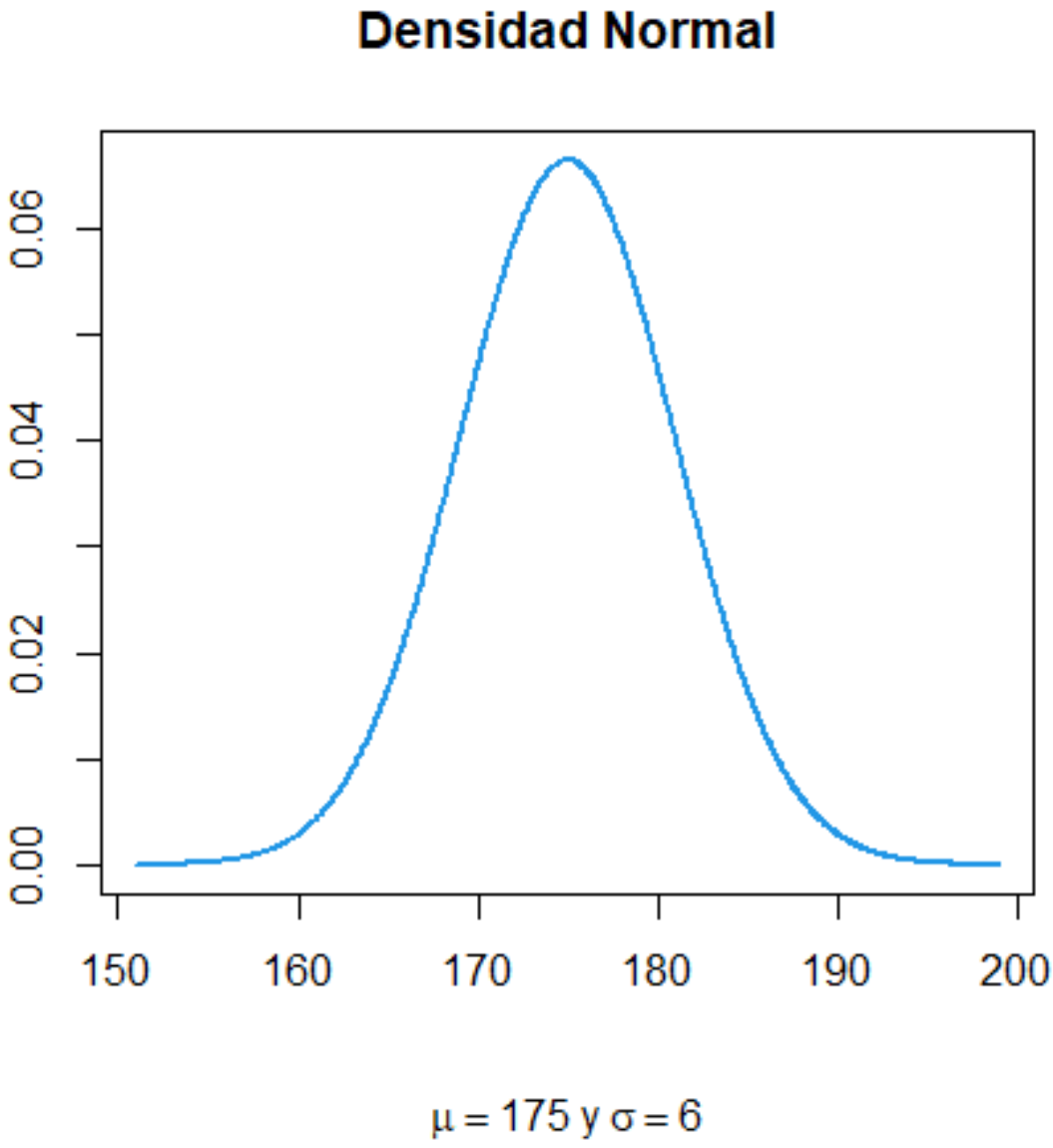


Figure 1: Distribución normal

PROPIEDADES DE LA DISTRIBUCIÓN t DE STUDENT

- Cada curva tiene forma de campana con centro en 0.
- Cada curva t está más dispersa que la curva normal estandar z
- A medida que v aumenta, la distribución de la curva t correspondiente disminuye
- A medida que $v \to \infty$ la secuencia de curvas t se aproxima a la curva normal estandar, por lo que la curva z recibe a veces el nombre de curva t con grados de libertad (gl)$gl = \infty$.

La distribución de la variable aleatoria t está dada por:

$$f(t) = \frac{\Gamma\left(\frac{v+1}{2}\right)}{\Gamma\left(\frac{v}{2}\right) \cdot \sqrt{\pi v \sigma}} \cdot \left(1 + \frac{1}{v}\left(\frac{x-\mu}{\sigma}\right)^2\right)^{-\frac{v+1}{2}}$$

De donde debemos recordar que σ^2 corresponde a varianza y no desviación estandar. Y que v son los grados de libertad.

La formula de la varianza es:

$$\sigma^2 \cdot \frac{v}{v-2}$$

, la moda es $= \mu$

TEOREMA DEL LÍMITE CENTRAL

Sean y_1, y_2,..., y_n variables aleatorias independientes y distribuidas idénticamente con $E(y_i) = y\ vy_i = 2 < \infty$. Definamos:

$$U_n = \frac{\sum_{i=1}^{n} y_i - n\mu}{\sigma\sqrt{n}} = \frac{\bar{y} - \mu}{\frac{\sigma}{\sqrt{n}}}$$

Entonces la función de distribución de U_n converge hacia la función de distribución normal estándar cuando $n \to \infty$. Esto es:

$$P(U_n) = \int_{-\infty}^{u} \frac{1}{\sqrt{2\pi}} \cdot e^{\frac{-t^2}{2}}\, dt$$

Para toda u. Es decir que $\bar{y}$, está distribuida normalmente en forma asintótica con media μ y varianza

$$\frac{\sigma^2}{n}$$

.

El teorema central del límite se puede aplciar a una muestra aleatoria y_1,y_2,...,y_n para cualquier distribución mientras $E(y_i) = \mu$ y $V(y_i) = \sigma^2$ sean finitas y el tamaño muestral sea grande.

ESTIMADORES PUNTUALES INSESGADOS COMUNES

DEFINICIÓN Un estimador es una regla, a menudo expresada como una fórmula, que indica cómo calcular el valor de una estimación con base en las mediciones contenidas en una muestra.

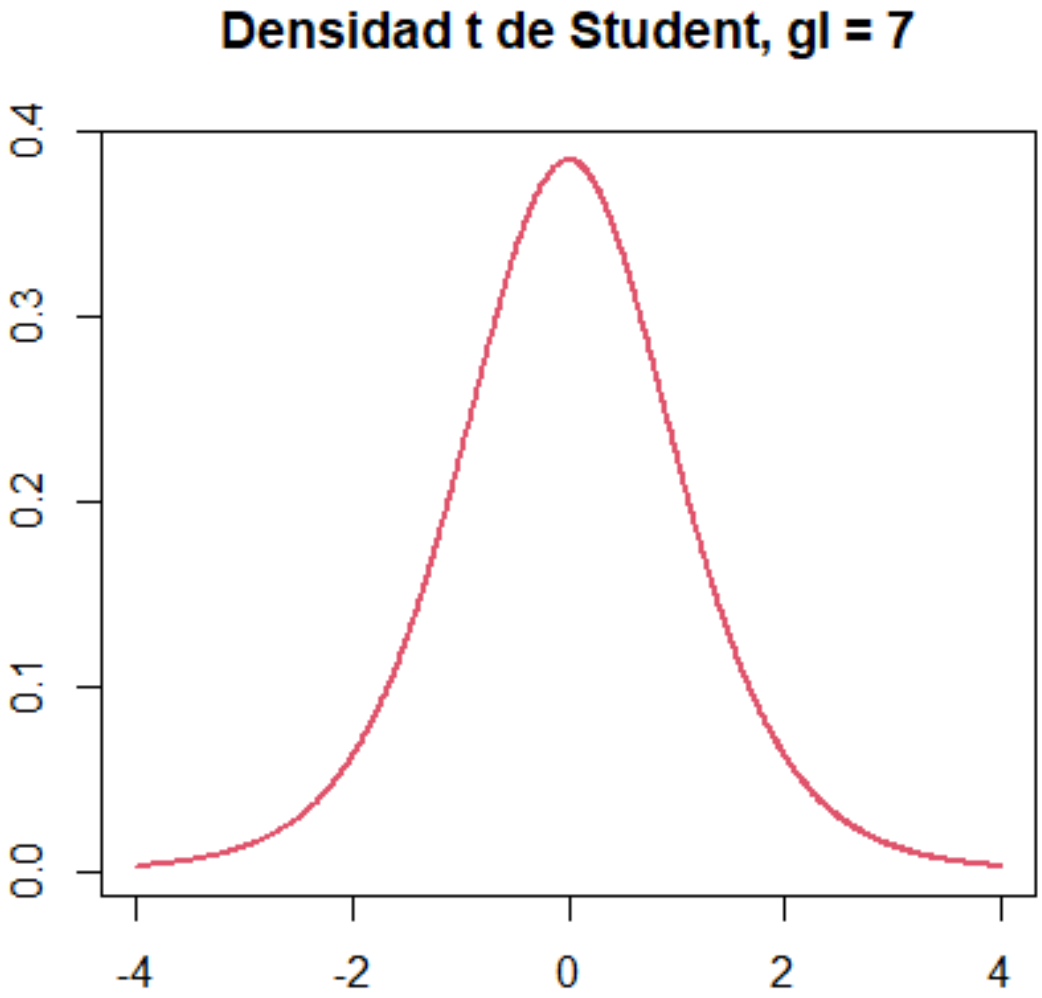


Figure 2: Densidad t de student con 7 GDL

Definición. Si $\hat{\theta}$ es un estimador puntual de un parámetro θ, entonces $\hat{\theta}$ es un estimador insesgado si $E\left(\hat{\theta}\right) = \theta$. Si $E\left(\hat{\theta}\right) \neq \theta$, se dice que $\hat{\theta}$ está sesgado.

Valores esperados y errores estándar de algunos estimadores puntuales comunes

Parámetro objetivo θ	Tamaño(s) muestral(es)	Estimador puntual $\hat{\theta}$	$E(\hat{\theta})$	Error estándar $\sigma_{\hat{\theta}}$
μ	n	$\overline{Y}$	μ	$\frac{\sigma}{\sqrt{n}}$
p	n	$\hat{p} = \frac{Y}{n}$	p	$\sqrt{\frac{pq}{n}}$
$\mu_1 - \mu_2$	n_1 y n_2	$\overline{Y}_1 - \overline{Y}_2$	$\mu_1 - \mu_2$	$\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}$
$p_1 - p_2$	n_1 y n_2	$\hat{p}_1 - \hat{p}_2$	$p_1 - p_2$	$\sqrt{\frac{p_1 q_1}{n_1} + \frac{p_2 q_2}{n_2}}$

σ_1^2 y σ_2^2 son las varianzas de las poblaciones 1 y 2, respectivamente.

Se supone que las dos muestras son independientes.

Figure 3: valores esperados y errores estándar de algunos estimadores puntuales comunes

CONTRASTE DE HIPOTESIS

Los elementos de un contraste de hipotesis son: * Hipótesis nula, H0 * Hipotesis alternativa, Ha * Estadístico de prueba * Región de rechazo

Nota: También llamaremos prueba de hipótesis a un contraste de hipótesis, sin caer en discusiones formales. Buscamos decidirnos por una de las hipótesis y no estamos exentos de cometer errores.

Definición. Se comete un *error tipo I* si H_0 es rechazada cuando H_0 es verdadera. La *probabilidad de un error tipo I* está denotada por α. El valor de α se denomina *nivel* de la prueba.

Se comente un *error tipo II* si H_0 es aceptada cuando H_a es verdadera. La probabilidad de un *error tipo II* está denotada por β.

Error tipo I y tipo II

H_0: No hay embarazo *vs* H_a: Hay embarazo

CONTRASTES COMUNES CON MUESTRAS GRANDES

Suponga que deseamos contrastar un conjunto de hipótesis respecto a un parámetro con base a una muestra aleatoria $Y_1, Y_2, \ldots, Y_n$. En esta sección desarrollaremos procedimientos de contrastes de hipótesis que están basados en un estimador que tiene una distribución muestral normal (aproximadamente) con media y error estándar .

Si 0 es un valor específico de, podemos probar H0: $\theta = 0$ contra Ha: $\theta > 0$. En este caso, las hipótesis nula y alternativa, el estadístico de prueba y la región de rechazo son como sigue:

$$H_0 : \theta = \theta_0 .$$

$$H_a : \theta > \theta_0 .$$

Estadístico de prueba: $\hat{\theta}$.

Región de rechazo: RR = { $\hat{\theta} > k$ }, para alguna selección de k.

El valor real de k en la región de rechazo RR se determina al fijar la probabilidad α de error tipo I (el nivel de la prueba) y escoger k de conformidad. Si H_0 es verdadera, $\hat{\theta}$ tiene una distribución aproximadamente normal con media θ_0 y error estándar $\sigma_{\hat{\theta}}$.

Por tanto, si deseamos una prueba de nivel α,

$$k = \theta_0 + z_\alpha \sigma_{\hat{\theta}}$$

Es la selección apropiada para k [si Z tiene una distribución normal estándar, entonces Z_α es tal que $P(Z > Z_\alpha) = \alpha$].

Figure 4: A

CONTRASTES DE HIPOTESIS DE NIVEL PARA MUESTRAS GRANDES

¿Cómo decidir cuál hipótesis alternativa usar para una prueba? La respuesta depende de la hipótesis que pretendemos apoyar. Si estamos interesados sólo en detectar un aumento en el porcentaje de piezas defectuosas, por ejemplo, debemos localizar la región de rechazo en la cola superior de la distribución normal estándar. Por otra parte, si deseamos detectar un cambio en p ya sea arriba o debajo de p=0.10, debemos localizar la región de rechazo en ambas colas de la distribución normal estándar y emplear una prueba de dos colas.

$H_0 : \theta = \theta_0$.

$H_a : \{\theta > \theta_0$ *(alternativa de cola superior)* . $\theta < \theta_0$ *(alternativa de cola inferior)* . $\theta \neq \theta_0$ *(alternativa de dos colas)* .

Estadístico de prueba:

$$Z = \frac{\hat{\theta} - \theta_0}{\sigma_{\hat{\theta}}}$$

Región de rechazo:

$\{\{z > z_\alpha\}$ *(RR de cola superior)* . $\{z <- z_\alpha\}$ *(RR de cola inferior)* . $\left\{|z| > z_{\frac{\alpha}{2}}\right\}$ *(RR de dos colas)* .

Figure 5: B

Contraste de hipótesis con muestras pequeñas para μ y $\mu_1 - \mu_2$

Supongamos que $Y_1, Y_2, \ldots, Y_n$ denotan una muestra aleatoria de tamaño *n* de una distribución normal con media μ desconocida y varianza σ^2 desconocida. Si $\overline{Y}$ y S denotan la media muestral y la desviación estándar muestral, respectivamente, y si $H_0 : \mu = \mu_0$ es verdadera, entonces

$$T = \frac{\overline{Y} - \mu_0}{S/\sqrt{n}}$$

Tiene una distribución *t* con *n-1* grados de libertad.

Figure 6: C

Contraste de muestra pequeña para μ

Suposiciones: $Y_1, Y_2, \ldots, Y_n$ constituyen una muestra aleatoria de una distribución normal con $E(Y_i) = \mu$.

$H_0: \mu = \mu_0$.

$H_a: \{\mu > \mu_0$ *(alternativa de cola superior)*. $\mu < \mu_0$ *(alternativa de cola inferior)*. $\mu \neq \mu_0$ *(alternativa de dos colas)*.

Estadístico de prueba: $T = \frac{\overline{Y} - \mu_0}{S/\sqrt{n}}$.

Región de rechazo:
$\{t > t_\alpha$ *(RR de cola superior)*. $t < -t_\alpha$ *(RR de cola inferior)*. $|t| > t_{\alpha/2}$ *(RR de dos colar)*.

Figure 7: D

Contrastes con muestras pequeñas para comparar dos medias poblacionales

Suposiciones: muestras independientes de distribuciones normales con $\sigma_1^2 = \sigma_2^2$.

$H_0: \mu_1 - \mu_2 = D_0$. Donde D_0 es un número fijo.

$H_a: \{\mu_1 - \mu_2 > D_0$ *(alternativa de cola superior)*. $\mu_1 - \mu_2 < D_0$ *(alternativa de cola inferior)*.

$\mu_1 - \mu_2 \neq D_0$ *(alternativa de dos colas)*

Estadístico de prueba:

$$T = \frac{\overline{Y}_1 - \overline{Y}_2 - D_0}{S_p\sqrt{\frac{1}{n_1} + \frac{1}{n_2}}}$$

Donde T tiene una distribución t de Student con $n_1 + n_2 - 2$ grados de libertad y

$$S_p = \sqrt{\frac{(n_1 - 1)S_1^2 + (n_2 - 1)S_2^2}{n_1 + n_2 - 2}}$$

Figure 8: E

Estadístico de prueba:

$$T = \frac{\overline{Y}_1 - \overline{Y}_2 - D_0}{S_p\sqrt{\frac{1}{n_1}+\frac{1}{n_2}}}$$

Donde T tiene una distribución t de Student con $n_1 + n_2 - 2$ grados de libertad y

$$S_p = \sqrt{\frac{(n_1-1)S_1^2+(n_2-1)S_2^2}{n_1+n_2-2}}$$

Región de rechazo:
$\{t > t_\alpha$ $(RR\ de\ cola\ superior)$. $t < -t_\alpha$ $(RR\ de\ cola\ inferior)$. $|t| > t_{\alpha/2}$ $(RR\ de\ dos\ colas)$.

Aquí, $P(T > t_\alpha) = \alpha$.

Figure 9: F

WORK

Estudiar algunas distribuciones de probabilidad muy comunes y útiles, obtener estimaciones puntuales con propiedades deseables utilizando algunos estimadores insesgados comunes. Llevar a cabo contrastes de hipótesis que ayuden a tomar decisiones.

En esta sesión estudiaremos temas relacionados con los siguientes puntos:

- Cálculo de probabilidades y cuantiles de las distribuciones binomial, normal y t de Student
- Generación de muestras aleatorias de las distribuciones estudiadas
- Estudio del teorema central del límite mediante simulaciones
- Propiedades de algunos estimadores puntuales insesgados comunes
- Contraste de hipótesis con muestras grandes y pequeñas

EJEMPLO 1 SESION 4. DISTRIBUCIONES BINOMIAL, NORMAL Y T DE STUDENT

OBJETIVO

- Aprender a obtener probabilidades, cuantiles y muestras aleatorias relacionadas con las distribuciones binomial, normal y t de Student
- Intepretar las probabilidades cuando se condieran las gráficas de las funciones de probabilidad y de densidad

REQUISITOS

- Tener `R` y Rstudio instalados
- Haber leído el pre-work

DESARROLLO

```
library(ggplot2) # Utilizaremos estos paquetes para algunas gráficas
library(reshape2)
```

DISTRIBUCION BINOMIAL

En el caso de la **Distribución binomial** En R para calcular valores de las funciones de probabilidad, distribución o cuantiles de la distribución binomial (discreta), usamos las funciones *dbinom*, *pbinom* y *qbinom* respectivamente. Para generar muestras aleatorias de esta distribución utilizamos la función *rbinom*.

Consideremos un experimento binomial con $n = 30$ pruebas idénticas e independientes, en donde la probabilidad de éxito en cada prueba es $p = 0.2$ (parámetros $n = 30$ y $p = 0.2$) 1. Suponga que realiza un examen de opción múltiple con 30 preguntas, en donde cada pregunta tiene 5 posibles respuestas, pero solo una es correcta siempre. Si elige la respuesta al azar en cada pregunta, y estamos interesados en el número de respuestas correctas obtenidas al final ¿Podemos decir que estamos ante un experimento binomial?

FUNCIÓN DE PROBABILIDAD

Para obtener P(X = 20), es decir, la probabilidad de observar 20 éxitos exactamente, en R ejecutamos:

```
dbinom(x = 20, size = 30, prob = 0.2)
```

```
[1] 3.382768e-08
```

Para obtener P($x \leq 20$), es decir, la probabilidad de observar a lo mucho 20 exitos o menos ejecutamos:

```
pbinom(x<=20, size = 30, prob = 0.2)
```

La diferencia entre *dbinom*() y *pbino*() es que *dbinom*() me dice cuál es la probabilidad de que $Pr(X = x)$ mientras que *pbinom*() te calcula la probabilidad de que $Pr(X \leq x)$

Para encontrar el valor más pequeño b tal que P(X <= b) >= 0.35, es decir, el cuantil de orden 0.35, usamos:

CUANTILES

```
qbinom(p = 0.35, size = 30, prob = 0.2) # b = 5

pbinom(q = 4, size = 30, prob = 0.2) # P(X <= 4) = 0.2552 < 0.35
pbinom(q = 5, size = 30, prob = 0.2) # P(X <= 5) = 0.4275 >= 0.35
pbinom(q = 6, size = 30, prob = 0.2) # P(X <= 6) = 0.6070 >= 0.35
```

```
[1] 5
[1] 0.2552333
[1] 0.4275124
[1] 0.6069699
```

MUESTRAS ALEATORIAS

Para obtener una muestra aleatoria de tamaño $n = 1000$, de la distribución binomial con parámetros como especificamos, hacemos

```
set.seed(4857) # Establecemos una semilla,
# para poder reproducir la muestra en el futuro
muestra <- rbinom(n = 1000, size = 30, prob = 0.2)
length(muestra); muestra[1:3]
```

```
[1] 1000
[1] 4 7 8
```

Podemos observar las frecuencias absolutas de los distintos valores obtenidos

```
as.data.frame(table(muestra))
```

```
   muestra Freq
1        0    1
2        1    8
3        2   31
4        3   77
5        4  132
6        5  161
7        6  196
8        7  146
9        8  126
10       9   65
11      10   30
12      11   13
13      12   10
14      13    2
15      15    2
```

También podemos observar las frecuencias relativas:

```
(df1 <- as.data.frame(table(muestra)/length(muestra)))

valg <- as.character(df1$muestra) # distintos valores generados por rbinom
(valg <- as.numeric(valg)) # Convertimos a números
```

```
   muestra  Freq
1        0 0.001
2        1 0.008
3        2 0.031
4        3 0.077
5        4 0.132
6        5 0.161
7        6 0.196
8        7 0.146
9        8 0.126
10       9 0.065
```

```
11      10 0.030
12      11 0.013
13      12 0.010
14      13 0.002
15      15 0.002
 [1]  0  1  2  3  4  5  6  7  8  9 10 11 12 13 15
```

Las frecuencias relativas son muy parecidas a las siguientes probabilidades

```
(v1 <- round(sapply(valg, dbinom, size = 30, p = 0.2), 3))
```

```
 [1] 0.001 0.009 0.034 0.079 0.133 0.172 0.179 0.154 0.111 0.068 0.035 0.016
[13] 0.006 0.002 0.000
```

Combinamos y unimos en un solo dataframe $df1$ y $v1$

```
(df2 <- cbind(df1, v1))
(names(df2) <- c("Exitos", "FR", "Prob"))

(df2 <- melt(df2)) # función del paquete reshape2
```

```
Using Exitos as id variables
```

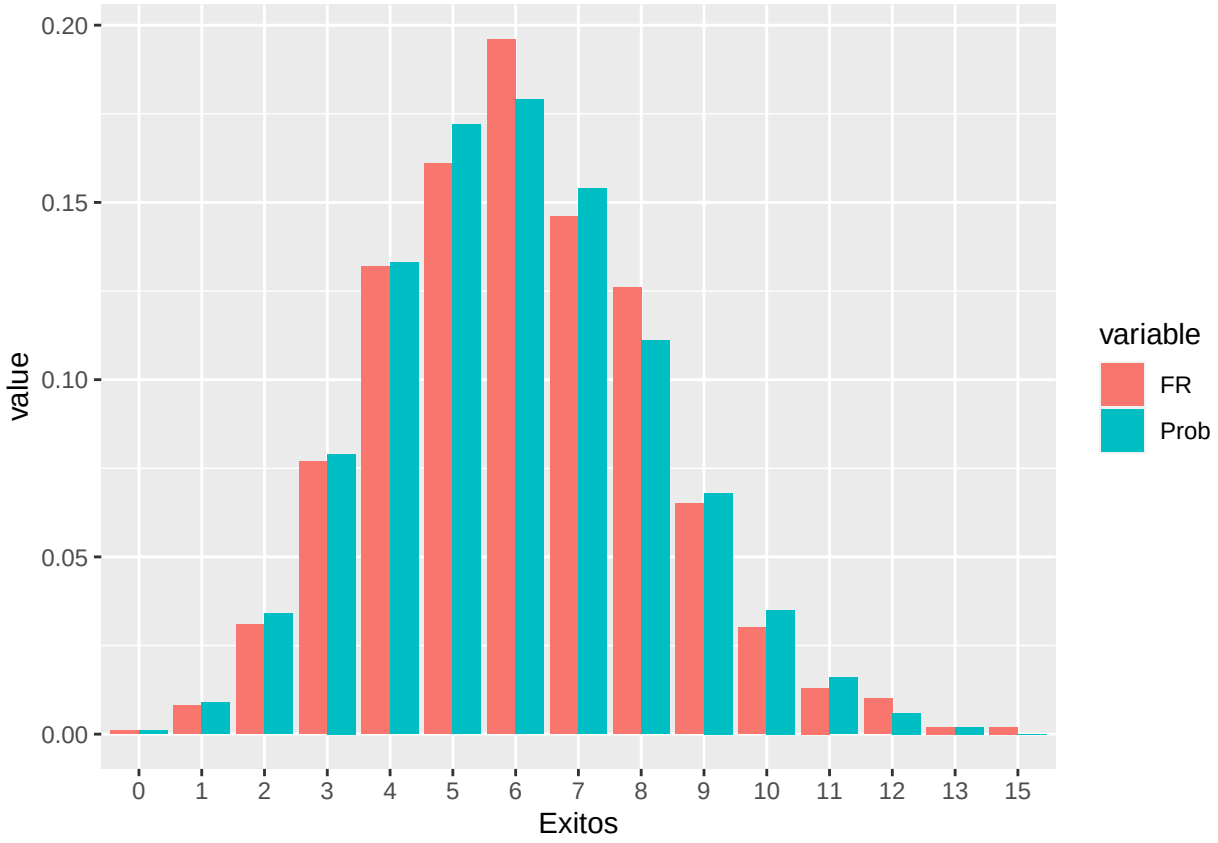
```
   muestra  Freq    v1
1        0 0.001 0.001
2        1 0.008 0.009
3        2 0.031 0.034
4        3 0.077 0.079
5        4 0.132 0.133
6        5 0.161 0.172
7        6 0.196 0.179
8        7 0.146 0.154
9        8 0.126 0.111
10       9 0.065 0.068
11      10 0.030 0.035
12      11 0.013 0.016
13      12 0.010 0.006
14      13 0.002 0.002
15      15 0.002 0.000
[1] "Exitos" "FR"     "Prob"
   Exitos variable value
1       0       FR 0.001
2       1       FR 0.008
3       2       FR 0.031
4       3       FR 0.077
5       4       FR 0.132
6       5       FR 0.161
7       6       FR 0.196
8       7       FR 0.146
9       8       FR 0.126
10      9       FR 0.065
11     10       FR 0.030
```

```
12      11        FR 0.013
13      12        FR 0.010
14      13        FR 0.002
15      15        FR 0.002
16       0      Prob 0.001
17       1      Prob 0.009
18       2      Prob 0.034
19       3      Prob 0.079
20       4      Prob 0.133
21       5      Prob 0.172
22       6      Prob 0.179
23       7      Prob 0.154
24       8      Prob 0.111
25       9      Prob 0.068
26      10      Prob 0.035
27      11      Prob 0.016
28      12      Prob 0.006
29      13      Prob 0.002
30      15      Prob 0.000
```

Melt en cierta forma une los dataframes en uno solo pero basandose en ciertas condiciones

Las frecuencias relativas son muy parecidas a las probabilidades:

```
ggplot(df2, aes(x = Exitos, y = value, fill = variable)) +
  geom_bar (stat="identity", position = "dodge")
```



```r
# Funciones del paquete ggplot2
```

DISTRIBUCIÓN NORMAL

En R para calcular valores de las funciones de densidad, distribución o cuantiles de la distribución normal (continua), usamos las funciones *dnorm*, *pnorm* y *qnorm* respectivamente. Para generar muestras aleatorias de esta distribución utilizamos la función *rnorm*.

Consideremos una variable aleatoria (v.a.) X que se distribuye como normal con media 175 y desviación estándar 6 (parámetros mu = 175 y sigma = 6).

FUNCIÓN DE DENSIDAD

La función de densidad sirve para caracterizar el comportamiento probable de una población en tanto especifíca la posibilidad relativa de que una variable aleatoria continuas X tome un valor cercano a x.

Densidad significa que la suma de todas las areas del histograma deben ser igual a 1. La funcion de densidad es el contorno. Es una línea continua que representa la distribucion de densidad de TODA LA POBLACIÓN.

FUNCION DE DISTRIBUCIÓN

La función de distribución asocia a cada valor de la variable alewatoria la probabilidad acumulada hasta ese valor, por ejemplo: Calcular la función de distribución de probabilidad de las puntuaciones obtenidas al lanzar un dado.

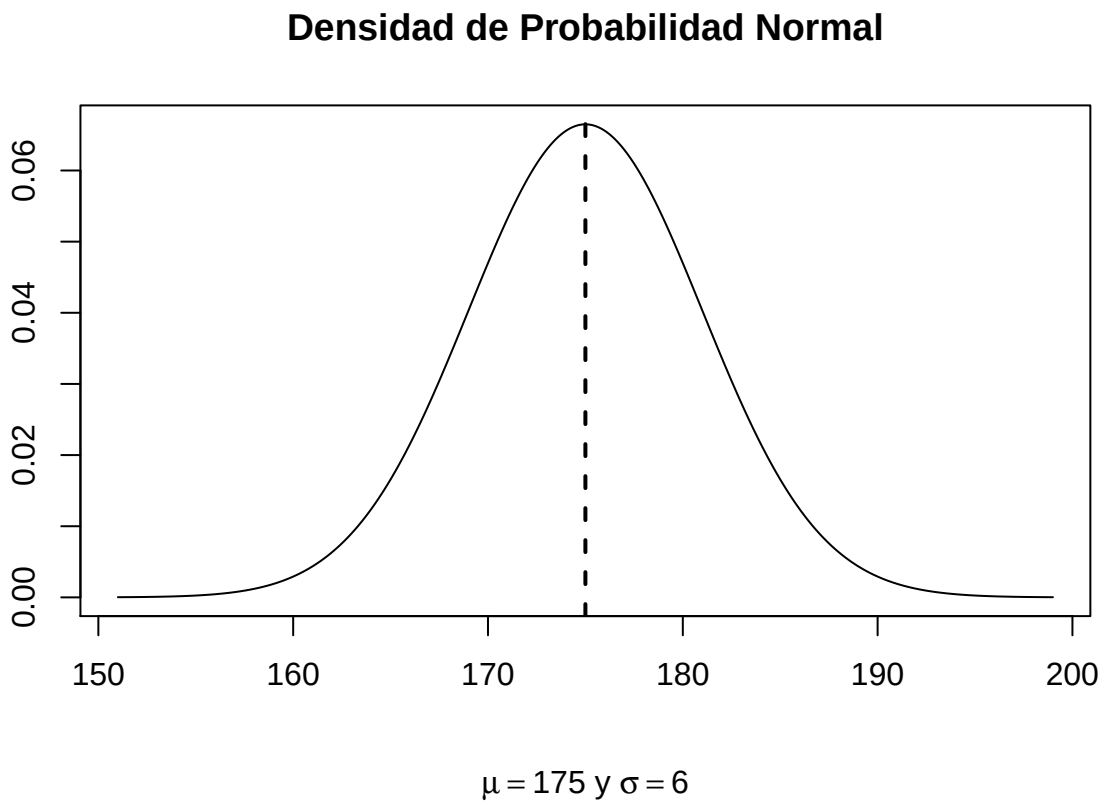
X	P_i
$x < 1$	0
$1 \leq x < 2$	$\frac{1}{6}$
$2 \leq x < 3$	$\frac{2}{6}$
$3 \leq x < 4$	$\frac{3}{6}$
$4 \leq x < 5$	$\frac{4}{6}$
$5 \leq x < 6$	$\frac{5}{6}$
$x \leq 6$	1

Siguiendo con el ejemplo disponible en el work tenemos lo siguiente:

Para obtener $P(x \leq 180)$, es decir, la probabilidad de que X tome un valor menor o igual a 180, ejecutamos:

```r
x <- seq(-4, 4, 0.01)*6 + 175 # Algunos posibles
# valores que puede tomar la v.a.
# X (mínimo: mu-4sigma, máximo: mu+4sigma)
```

```r
y <- dnorm(x, mean = 175, sd = 6) # Valores
# correspondientes de la función de
# densidad de probabilidad
plot(x, y, type = "l", xlab = "", ylab = "")#Te grafica
# en forma de campana los valores de x y y
title(main = "Densidad de Probabilidad Normal",
      sub = expression(paste(mu == 175, " y ", sigma == 6)))#Agrega leyendas
#a gráfico
abline(v = 175, lwd = 2, lty = 2) # La media es 175 grafica la línea
```




```r
pnorm(q = 180, mean = 175, sd = 6)
```

```
[1] 0.7976716
```

```r
par(mfrow = c(2, 2))#ESTABLECE EL LIENZO EN EL PLOT PARA GRAFICAR
#VARIAS EN UN MISMO ESPACIO
plot(x, y, type = "l",
     xlab = "",
     ylab = "")
#AGREGA LEYENDAS AL GRÁFICO
title(main = "Densidad de Probabilidad Normal",
      sub = expression(paste(mu == 175,
                             " y ",
                             sigma == 6)))
```

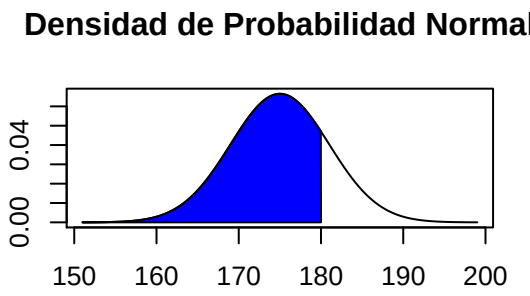
```
polygon(c(min(x),
          x[x<=180],
          180),
        c(0, y[x<=180], 0),
        col="blue")#GRAFICA DENTRO DE LA DISTRIBUCIÓN
#LA PROBABILIDAD DE QUE X TOME VALORES MENORES A 180


#para obtener la probabilidad de que x P(x<=165), es decir, la probabilidad de
#que X tome un valor menor o igual a 165, ejecutamos:

pnorm(q = 165, mean = 175, sd = 6)
```

```
[1] 0.04779035
```

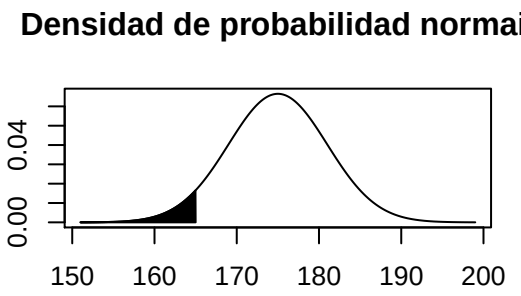
```
plot(x,y,type="l",
     xlab = "",
     ylab = "")
title(main = "Densidad de probabilidad normail",
      sub = expression(paste(mu==175,"y",sigma==6)))
polygon(c(min(x),
          x[x<=165],165),
        c(0,
          y[x<=165],0),
        col = "black")

#para obtener la probabilidad de que x P(x<=165), es decir, la probabilidad de
#que X tome un valor menor o igual a 165, ejecutamos:

pnorm(q = 165, mean = 175, sd = 6)
```

```
[1] 0.04779035
```

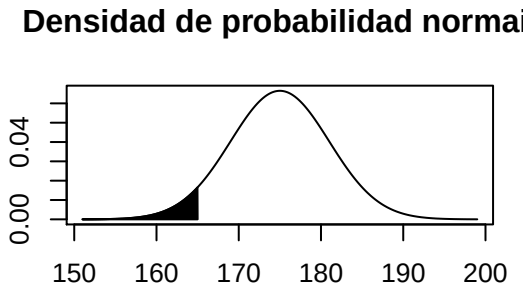
```
plot(x,y,type="l",
     xlab = "",
     ylab = "")
title(main = "Densidad de probabilidad normail",
      sub = expression(paste(mu==175,"y",sigma==6)))
polygon(c(min(x),
          x[x<=165],165),
        c(0,
          y[x<=165],0),
        col = "black")
```



$\mu = 175$ y $\sigma = 6$



$\mu = 175 y \sigma = 6$



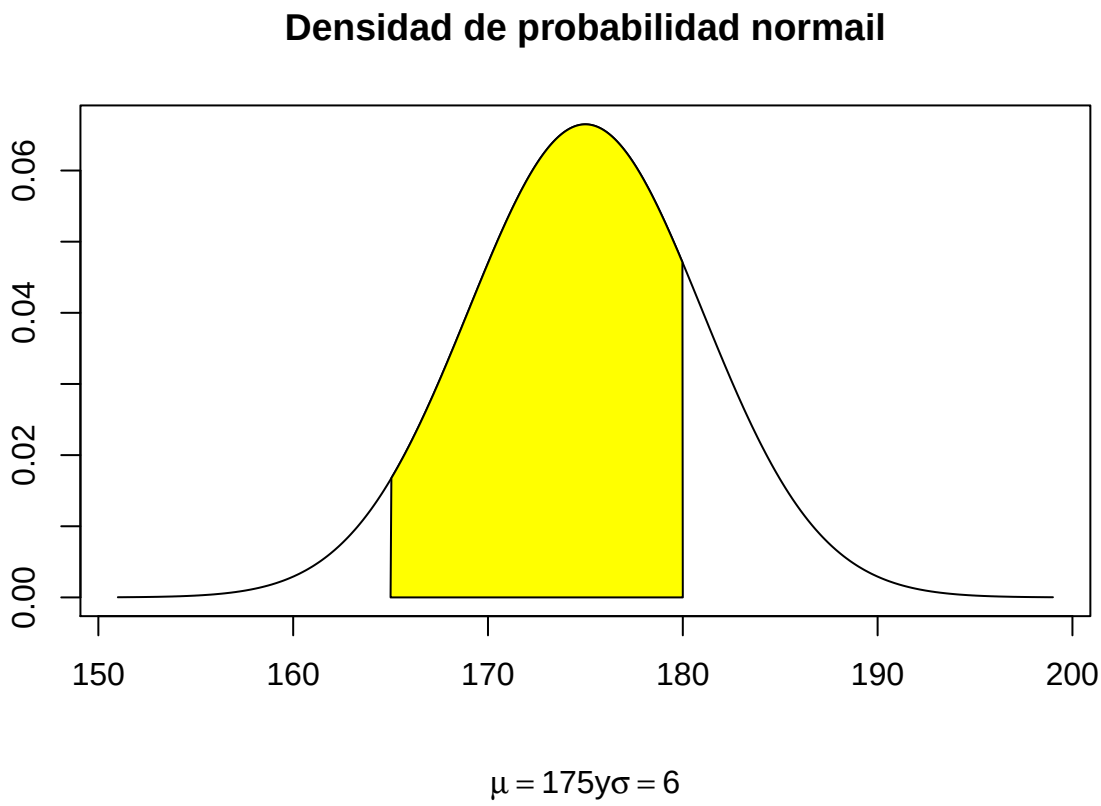
$\mu = 175 y \sigma = 6$

Para obtener la probabilidad de que $P(165 \leq X \leq 180)$ ejecutamos el siguiente comando:

```
#PARA OBTENER P(165 <= X <= 180), es decir, la probabilidad de que X
#tome un valor mayor o igual a 165 y menor o igual a 180, debemos correr

pnorm(q = 180, mean = 175, sd = 6) - pnorm(q = 165, mean = 175, sd = 6)
```

```
[1] 0.7498813
```

```
plot(x,y,type="l",
     xlab = "",
     ylab = "")
title(main = "Densidad de probabilidad normail",
      sub = expression(paste(mu==175,"y",sigma==6)))
polygon(c(165,
          x[x>=165&x<=180],180),
        c(0,
          y[x>=165&x<=180],0),
        col = "yellow")
```

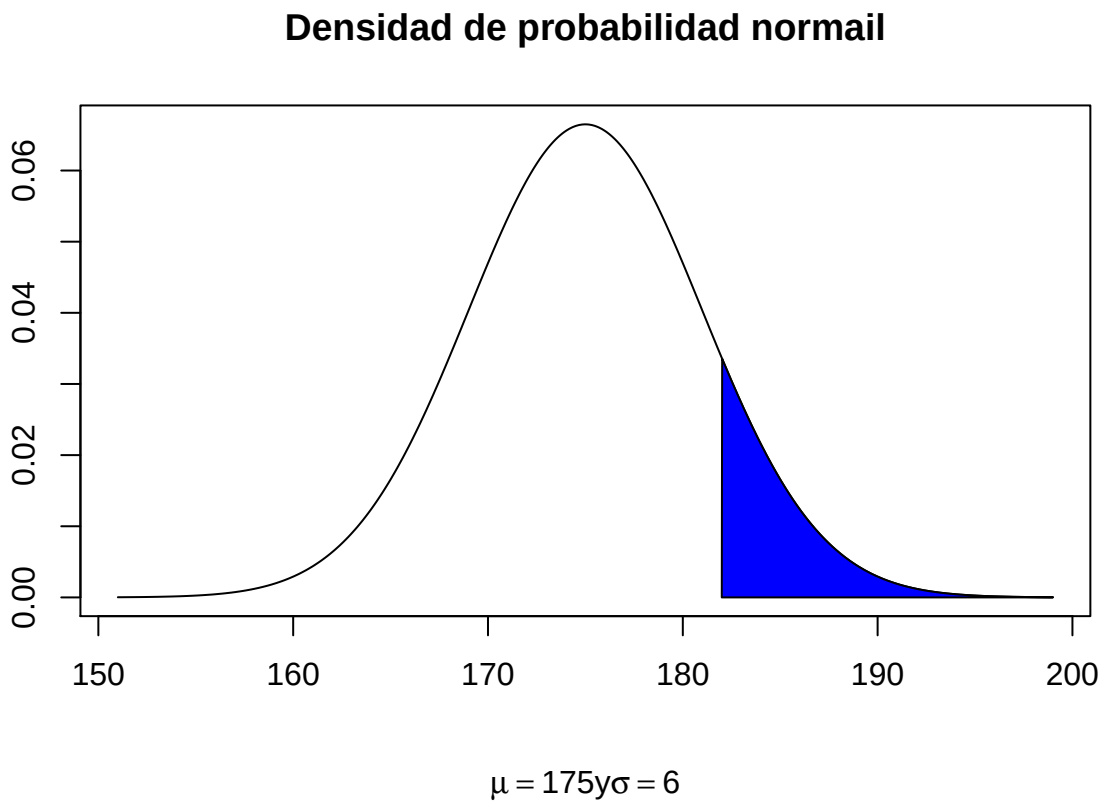


Para obtener $P(X \geq 182)$, es decir, la probabilidad de que X tome un valor mayor o igual a 182, una alternativa es:

```
pnorm(q = 182, mean = 175, sd = 6, lower.tail = FALSE)
```

```
[1] 0.1216725
```

```
#graficando
plot(x,y,type="l",
     xlab = "",
     ylab = "")
title(main = "Densidad de probabilidad normail",
      sub = expression(paste(mu==175,"y",sigma==6)))

polygon(c(182,
          x[x>=182], max(x)),
        c(0,
          y[x>=182], 0),
        col="blue")
```




```
dev.off() # Para mostrar solo una gráfica
```

```
null device
          1
```

CUANTILES

Recordemos que los cuantiles ayudan a determinar porcentajes que superan o no un determinado valor de la variable. Para encontrar el número b, tal que $P(X \leq b) = 0.75$, es decir, el cuantil de orden 0.75, ejecutamos:

```
(b <- qnorm(p = 0.75, mean = 175, sd = 6))
```

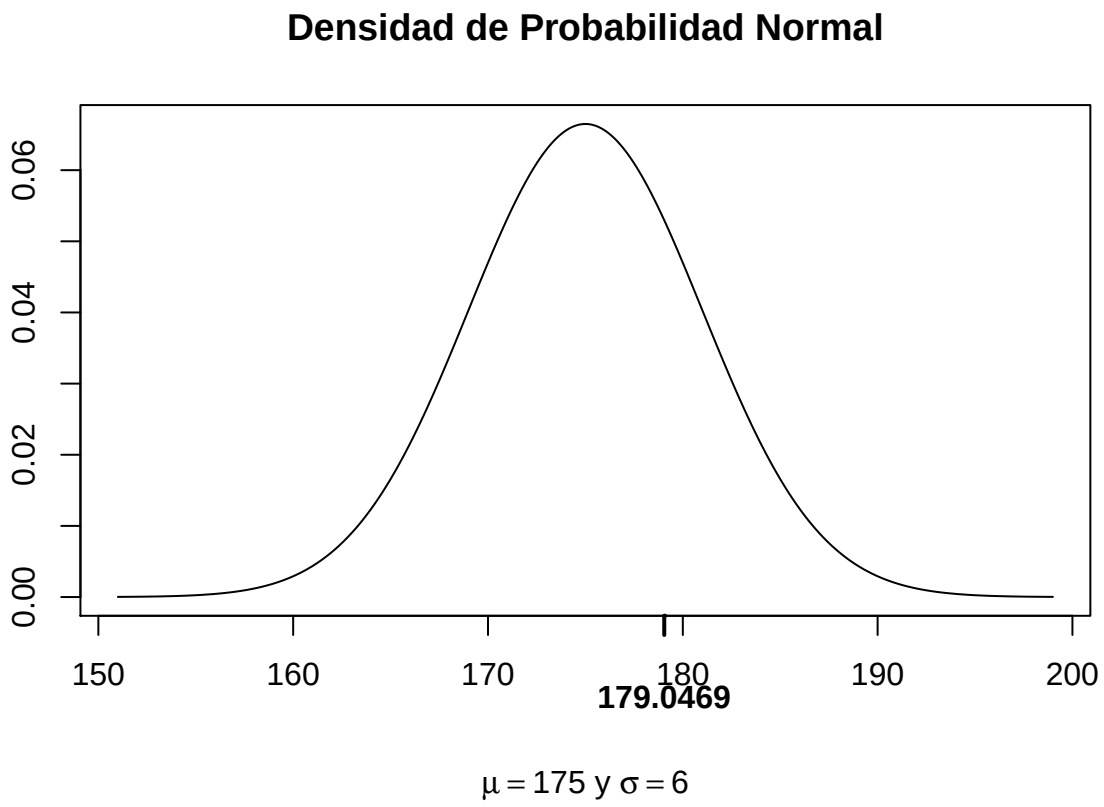
```
[1] 179.0469
```

```
#COMPROBANDO DE FORMA COMUN
pnorm(b, 175, 6)
```

```
[1] 0.75
```

```
# El cuantil se encuentra en el eje de medición (eje horizontal)
plot(x, y,
     type = "l",
```

```
     xlab="",
     ylab="")
title(main = "Densidad de Probabilidad Normal",
      sub = expression(paste(mu == 175, " y ", sigma == 6)))
#COLOCA UNA LÍNEA DONDE SE ENCUENTRA EL CUANTIL
axis(side = 1, at = b, font = 2, padj = 1, lwd = 2)
```




```
# SIDE un entero que especifica donde se colocará la linea
#1 abajo 2 izquierda 3 arriva 4 derecha
#AT es donde se colcoará la linea
#padj ajusta a cada linea, 0 es arriba o derecha, 1 es izq o inferior
```

MUESTRAS ALEATORIAS

Para generar una muestra aleatoria de tamaño n = 1000 de la v.a. X corremos la siguiente instrucción

```
set.seed(7563) # Para poder reproducir la muestra en el futuro
muestra <- rnorm(n = 1000, mean = 175, sd = 6)
length(muestra); mdf <- as.data.frame(muestra)
```

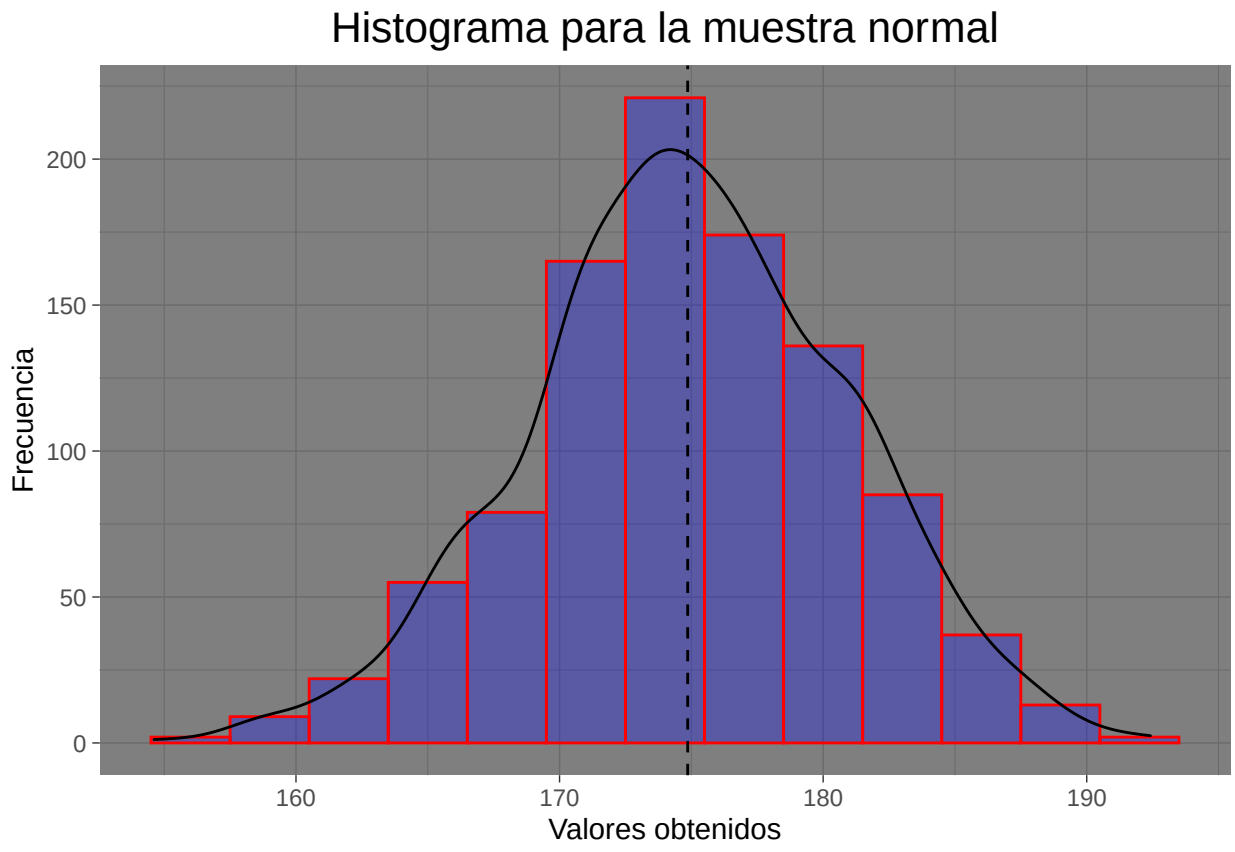
```
[1] 1000
```

```r
tail(mdf)
```

```
      muestra
995  164.7160
996  177.3317
997  156.9474
998  177.1626
999  171.3045
1000 172.4188
```

Observamos que el histograma de la muestra generada tiene forma de campana similar a la densidad de una normal

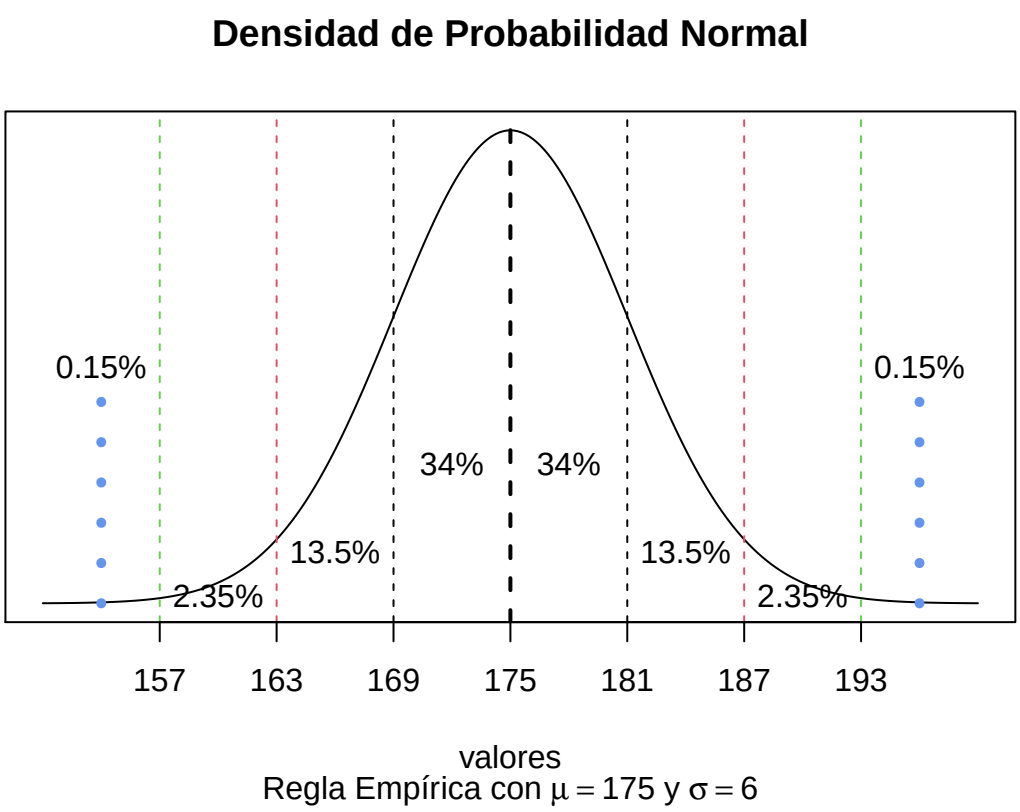
```r
ggplot(mdf, aes(muestra)) +
  geom_histogram(colour = 'red',
                 fill = 'blue',
                 alpha = 0.3, # Intensidad del color fill
                 binwidth = 3) +
  geom_density(aes(y = 3*..count..))+
  geom_vline(xintercept = mean(mdf$muestra), linetype="dashed", color = "black") +
  ggtitle('Histograma para la muestra normal') +
  labs(x = 'Valores obtenidos', y = 'Frecuencia')+
  theme_dark() +
  theme(plot.title = element_text(hjust = 0.5, size = 16))
```



REGLA EMPÍRICA

La regla empírica es una abreviatura utilizada para recordar el porcentaje de valores que se encuentran dentro de una banda alrededor de la media en una distribución normal con un ancho de dos, cuatro y seis veces la desviación típica, respectivamente.

```
mean <- 175; sd <- 6
x <- seq(mean-4*sd, mean+4*sd, 0.01)
y <- dnorm(x, mean, sd)
plot(x, y, type = "l",
     xlab="valores",
     ylab = "",
     xaxt = "n",
     yaxt = "n")
title(main = "Densidad de Probabilidad Normal",
      sub = expression(paste("Regla Empírica con ",
                             mu == 175,
                             " y ",
                             sigma == 6)))
abline(v=mean, lty = 2, lwd = 2)
for(k in c(-3, -2, -1, 1, 2, 3)) abline(v = mean+k*sd, lty = 2, col = abs(k))
ps <- c(mean - 3*sd, mean - 2*sd,
        mean - sd, mean,
        mean + sd,
        mean + 2*sd,
        mean + 3*sd)
axis(side = 1, at = ps)
x0 <- NULL
for(i in 1:length(ps)-1) x0 <- c(x0, (ps[i]+ps[i+1])/2)
y0 <- dnorm(x0, mean, sd)*1/3
text(x = x0, y = y0, labels = c("2.35%", "13.5%", "34%", "34%", "13.5%", "2.35%"))
x1 <- (x[1]+ps[1])/2; y1 <- dnorm(mean, mean, sd)*1/2
xf <- (x[length(x)]+ps[length(ps)])/2; yf <- dnorm(mean, mean, sd)*1/2
text(x = c(x1, xf), y = c(y1, yf), labels = c("0.15%", "0.15%"))
segments(x0 = x1, y0 = 0, x1 = x1, y1 = y1,               # Draw one line as in Example 1
         col = "cornflowerblue",                               # Color of line
         lwd = 5,                                              # Thickness of line
         lty = "dotted")
segments(x0 = xf, y0 = 0, x1 = xf, y1 = yf,
         col = "cornflowerblue",
         lwd = 5,
         lty = "dotted")
```



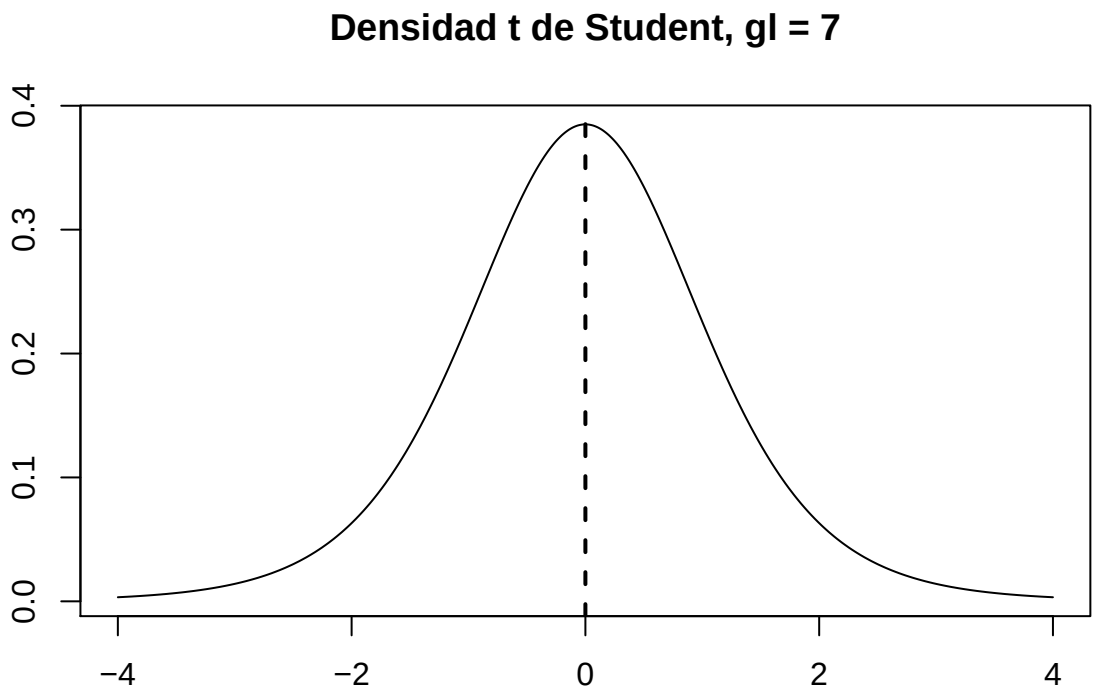
DISTRIBUCIÓN t DE STUDENT

En R para calcular valores de las funciones de densidad, distribución o cuantiles de la distribución t de Student (continua), usamos las funciones dt, pt y qt respectivamente. Para generar muestras aleatorias de esta distribución utilizamos la función rt.

Consideremos una variable aleatoria (v.a.) T que se distribuye como t de Student con 7 grados de libertad (gl) (parámetro gl = 7)

FUNCIÓN DE DENSIDAD

```
x <- seq(-4, 4, 0.01) # Algunos valores que puede tomar la v.a.
# T con 7 gl
y <- dt(x, df = 7) # Valores correspondientes de
# la densidad t de Student con 7 gl
plot(x, y, type = "l", main = "Densidad t de Student, gl = 7", xlab="", ylab="")
abline(v = 0, lwd=2, lty=2)
```



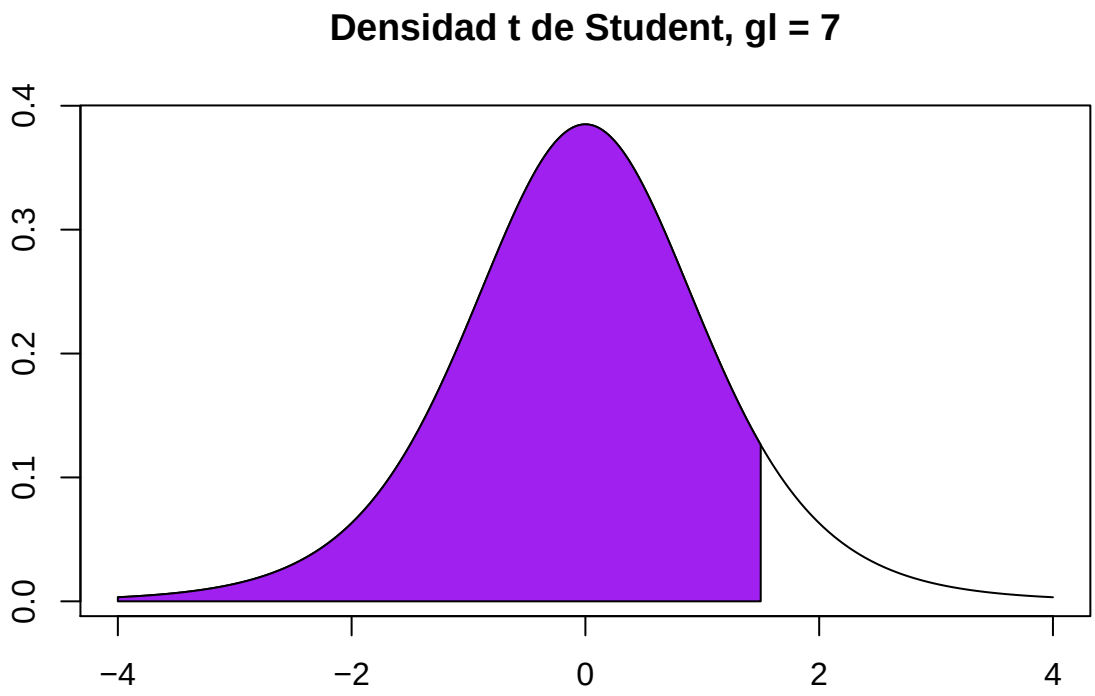
Para encontrar P(T <= 1.5), ejecutamos la siguiente instrucción

```
pt(q = 1.5, df = 7)
```

```
[1] 0.9113508
```

Observemos la región que corresponde a esta probabilidad en la siguiente gráfica

```
plot(x, y,
     type = "l",
     main = "Densidad t de Student, gl = 7",
     xlab="",
     ylab="")
polygon(c(min(x),x[x<=1.5], 1.5),
        c(0, y[x<=1.5], 0),
        col="purple")
```



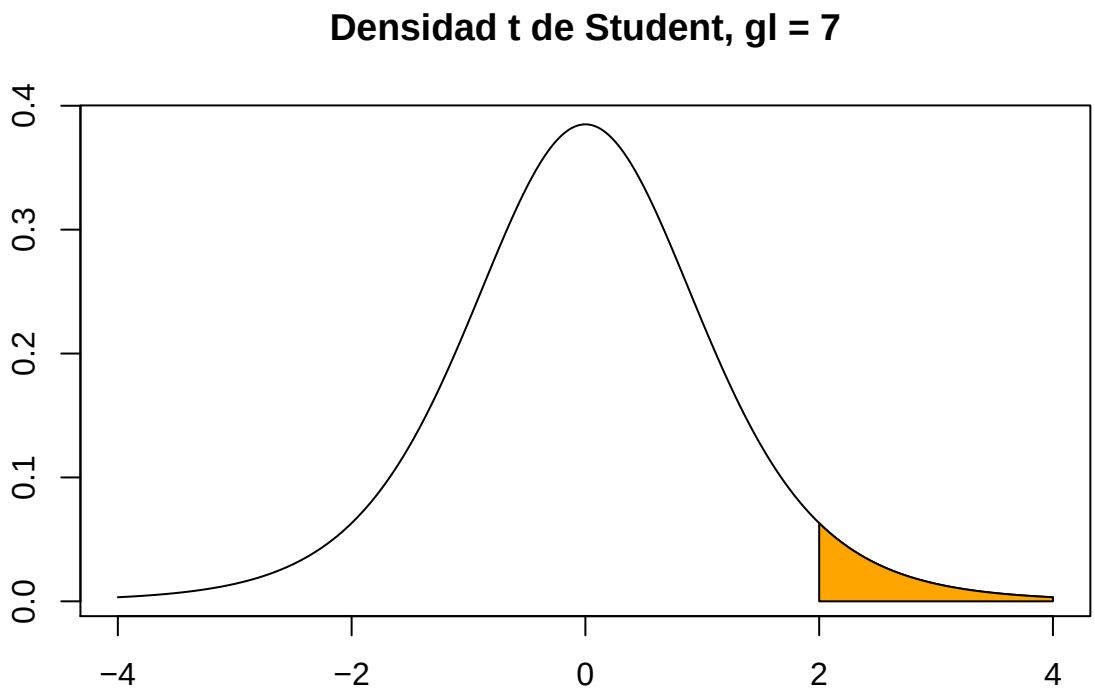
Para encontrar P(T >= 2), ejecutamos

```
pt(q = 2, df = 7, lower.tail = FALSE)
```

```
[1] 0.04280966
```

Observemos la región que corresponde a esta probabilidad en la siguiente gráfica

```
plot(x, y, type = "l",
     main = "Densidad t de Student, gl = 7",
     xlab="", ylab="")
polygon(c(2, x[x>=2], max(x)),
        c(0, y[x>=2], 0),
        col="orange")
```



CUANTILES

Para encontrar el número d tal que P(T <= d) = 0.025, es decir, el cuantil de orden 0.025, corremos la siguiente instrucción

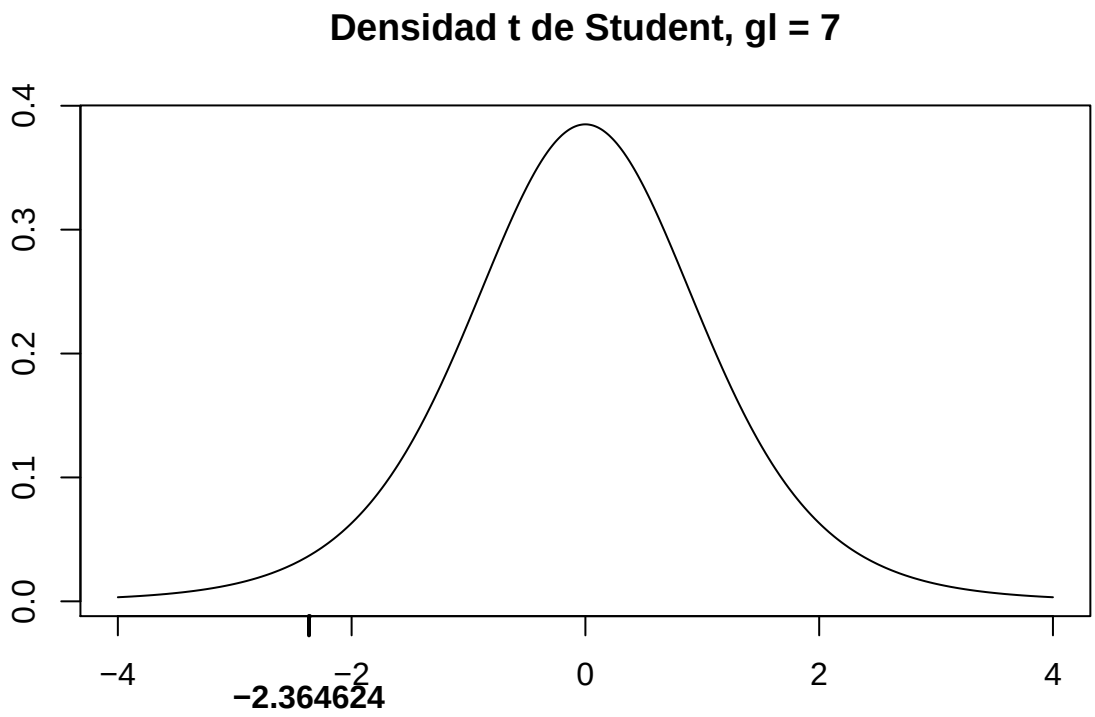
```
(d <- qt(p = 0.025, df = 7))
```

```
[1] -2.364624
```

```
#comprobando
pt(q = d, df = 7)
```

```
[1] 0.025
```

```
# Mostramos el cuantil encontrado en el eje de medición (eje horizontal)
plot(x, y,
     type = "l",
     main = "Densidad t de Student, gl = 7",
     xlab="",
     ylab="")
axis(side = 1,
     at = d,
     font = 2,
     padj = 1,
     lwd = 2)
```



MUESTRAS ALEATORIAS

Para generar una muestra aleatoria de tamaño n = 1000 de la v.a. T corremos la siguiente instrucción

```
set.seed(777) # Para poder reproducir la muestra en el futuro
muestra <- rt(n = 1000, df = 7)
length(muestra); mdf <- as.data.frame(muestra)
```

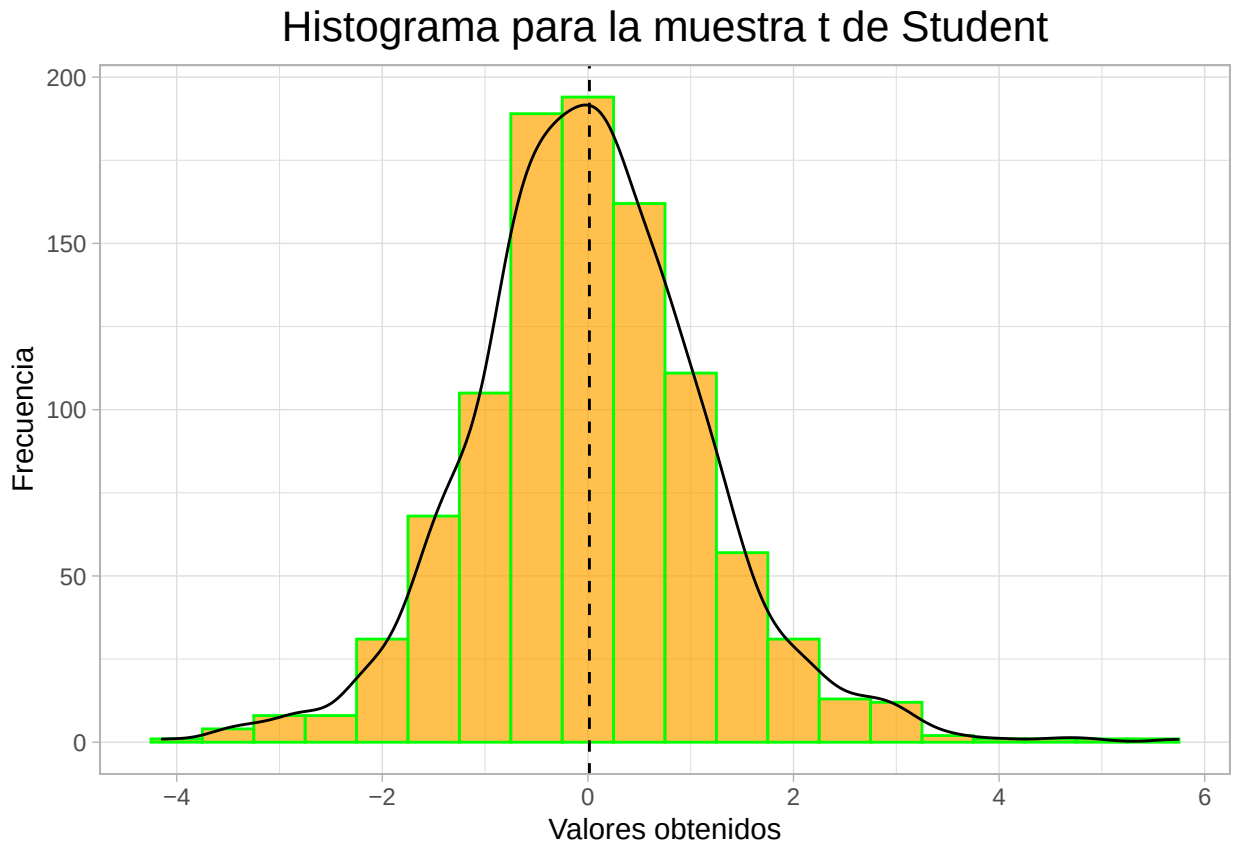
```
[1] 1000
```

```
tail(mdf)
```

```
          muestra
995   0.6257672958
996  -0.2794513280
997  -0.3037697310
998   0.4514007565
999  -0.0003503659
1000 -0.1808370397
```

Observamos que el histograma de la muestra generada tiene forma de campana similar a la densidad t de Student

```r
ggplot(mdf, aes(muestra)) +
  geom_histogram(colour = 'green',
                 fill = 'orange',
                 alpha = 0.7, # Intensidad del color fill
                 binwidth = 0.5) +
  geom_density(aes(y = 0.5*..count..))+
  geom_vline(xintercept = mean(mdf$muestra), linetype="dashed", color = "black") +
  ggtitle('Histograma para la muestra t de Student') +
  labs(x = 'Valores obtenidos', y = 'Frecuencia')+
  theme_light() +
  theme(plot.title = element_text(hjust = 0.5, size = 16))
```



ATENUAR COLORES

Para atenuar colores es necesario utilzar $alpha = 0.8$ o la cantidad porcentual para atenuarse.

RETO 1. DISTRIBUCIONES BINOMIAL, NORMAL Y T DE STUDENT

OBJETIVO

- Calcular probabilidades y cuantiles relacionadas con algunas distribuciones de probabilidad útiles y comunes
- Generar muestras aleatorias que provengan de las distribuciones estudiadas

REQUISITOS

- Haber trabajado con el Prework y el Work

DESARROLLO

DISTRIBUCIÓN BINOMIAL

Consideremos un experimento binomial con $n = 35$ pruebas idénticas e independientes, en donde la probabilidad de éxito en cada prueba es $p = 0.51$. Encuentre lo siguiente:

1. La probabilidad de observar exactamente 10 éxitos
2. La probabilidad de observar 10 o más exitos
3. El cuantil de orden 0.5
4. Genere una muestra aleatoria de tamaño 1000 de esta distribución, construya una tabla de frecuencias relativas con los resultados y realice el gráfico de barras de los resultados que muestre las frecuencias relativas.

DISTRIBUCIÓN NORMAL

Considere una variable aleatoria normal con media 110 y desviación estándar 7. Realice lo siguiente:

1. Grafique la función de densidad de probabilidad
2. Encuentre la probabilidad de que la v.a. sea mayor o igual a 140
3. Encuentre el cuantil de orden 0.95
4. Genere una muestra aleatoria de tamaño 1000 y realice el histograma de frecuencias relativas para esta muestra

```
#RETO 1 SESSION 4


##distribucion binomial
# Consideremos un experimento binomial con $n = 35$ pruebas idénticas e
# independientes, en donde la probabilidad de éxito en cada prueba es
# $p = 0.51$. Encuentre lo siguiente:
#
# 1. La probabilidad de observar exactamente 10 éxitos
# 2. La probabilidad de observar 10 o más exitos
# 3. El cuantil de orden 0.5
# 4. Genere una muestra aleatoria de tamaño 1000 de esta
# distribución, construya una tabla de frecuencias relativas con los
```

```r
# resultados y realice el gráfico de barras de los resultados que
# muestre las frecuencias relativas.

library(ggplot2)
#PUNTO 1
dbinom(x=10,size = 35,prob = 0.51)
```

```
[1] 0.003930318
```

```r
#PUNTO 2
pbinom(q=9,size = 35,prob = 0.51,
       lower.tail = FALSE)
```

```
[1] 0.9979185
```

```r
#PUNTO 3
qbinom(p = 0.5,size = 35,prob = 0.51)
```

```
[1] 18
```

```r
# PUNTO 4
set.seed(123)
ale1000<- rbinom(n = 1000,size = 35,prob = 0.51)
class(ale1000)
```

```
[1] "integer"
```

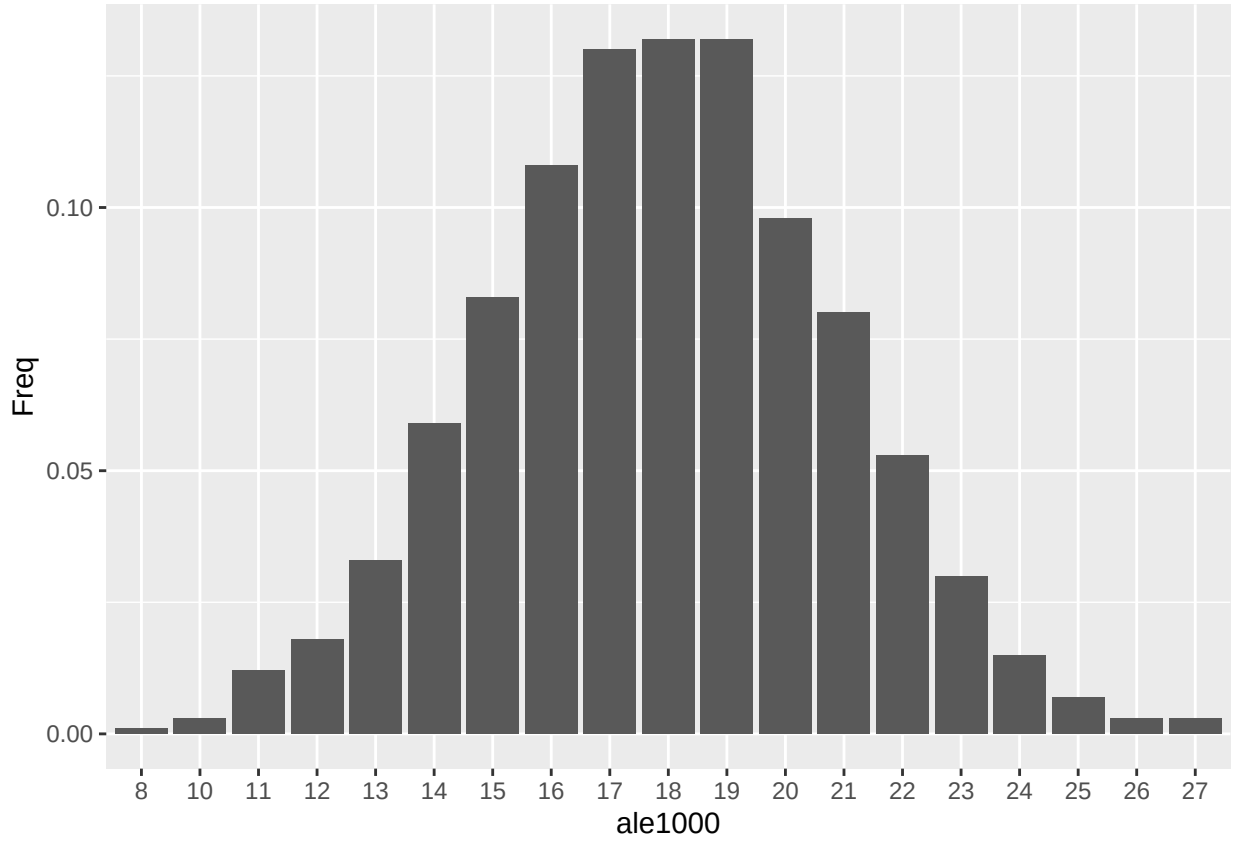
```r
ale1000dataaframe<-as.data.frame(table(ale1000)/length(ale1000))
head(ale1000dataaframe)
```

```
  ale1000  Freq
1       8 0.001
2      10 0.003
3      11 0.012
4      12 0.018
5      13 0.033
6      14 0.059
```

```r
tail(ale1000dataaframe)
```

```
   ale1000  Freq
14      22 0.053
15      23 0.030
16      24 0.015
17      25 0.007
18      26 0.003
19      27 0.003
```

```
ggplot(ale1000dataaframe,
       aes(x = ale1000,y = Freq))+geom_bar(stat = "identity")
```




```
##DISTRIBUCION NORMAL
# Considere una variable aleatoria normal con media 110
#y desviación estándar 7. Realice lo siguiente:
# 1. Grafique la función de densidad de probabilidad
# 2. Encuentre la probabilidad de que la v.a. sea mayor o igual a 140
# 3. Encuentre el cuantil de orden 0.95
# 4. Genere una muestra aleatoria de tamaño 1000 y realice el histograma
# de frecuencias relativas para esta muestra

#PUNTO 1

#GENERANDO LOS VALORES ALEATORIOS
vala1<- seq(10,100,by = 0.1)
View(vala1)
vala2<- dnorm(x = vala1,mean = 110,sd = 7)
class(vala1)
```

```
[1] "numeric"
```

```
class(vala2)
```

```
[1] "numeric"
```

```r
df<-data.frame(vala1,vala2)
class(df)
```

```
[1] "data.frame"
```

```r
View(df)
grafico<-ggplot(df,aes(vala1,vala2))+geom_line()


#PUNTO 2 PROBABILIDAD DE QUE VARIABLE ALEATORIA SEA MAYOR O IGUAL A 140
#LOWR TAIL SI ES VERDADERO LA PROBABILIDAD SERÁ P(X<=x) SI ES FALSO ES P(X>x)
pnorm(q = 140,mean = 110,sd = 7,lower.tail = FALSE)
```

```
[1] 9.107649e-06
```

```r
#PUNTO 3. CUARTIL DE ORDEN 0.95
qnorm(p = 0.95,mean = 110,sd = 7)
```

```
[1] 121.514
```

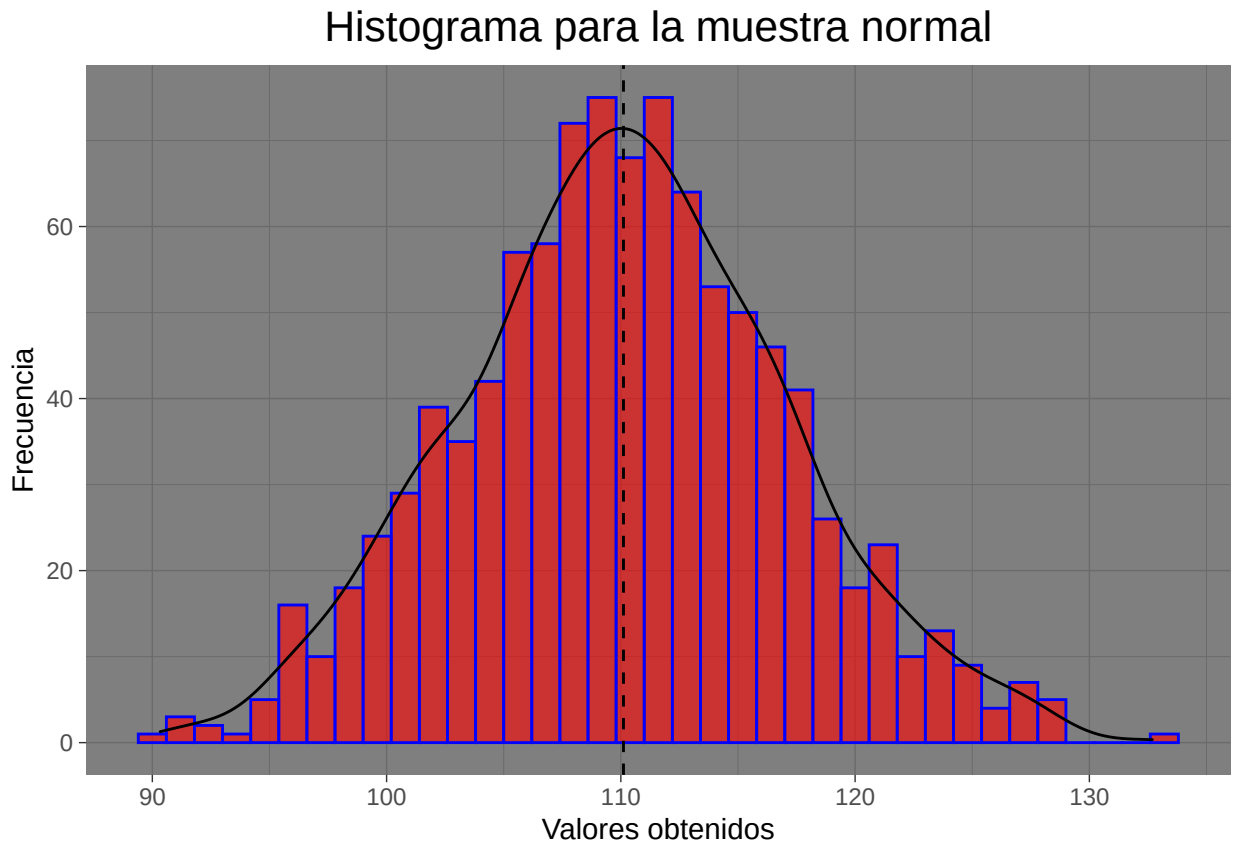
```r
#PUNTO 4.MUESTRA ALEATORIA DE TAMAÑO 1000 Y GENERE HISTOGRAMA DE FRECUENCIAS
set.seed(123)
datara<-rnorm(n = 1000,mean = 110,sd = 7)
dataradf<-as.data.frame(x = datara)
class(datara)
```

```
[1] "numeric"
```

```r
class(dataradf)
```

```
[1] "data.frame"
```

```r
ggplot(data = dataradf,aes(datara))+
  geom_histogram(colour="blue",
                 fill="red",
                 alpha=0.6,
                 binwidth = 1.2)+
geom_density(aes(y=1.2*..count..))+
  geom_vline(xintercept = mean(datara),
             linetype="dashed", color = "black")+
  ggtitle(label = 'Histograma para la muestra normal')+
  labs(x='Valores obtenidos',y='Frecuencia')+
  theme_dark()+
  theme(plot.title = element_text(hjust = 0.5, size = 16))
```



TEOREMA CENTRAL DEL LÍMITE

OBJETIVO

- Comprender lo que afirma el teorema central del límite obteniendo muestras aleatorias de diferentes tamaños en R y observando la manera en la que se distribuyen las medias de las muestras generadas

REQUISITOS

- Tener R y RStudio instalados
- Haber estudiado el Prework

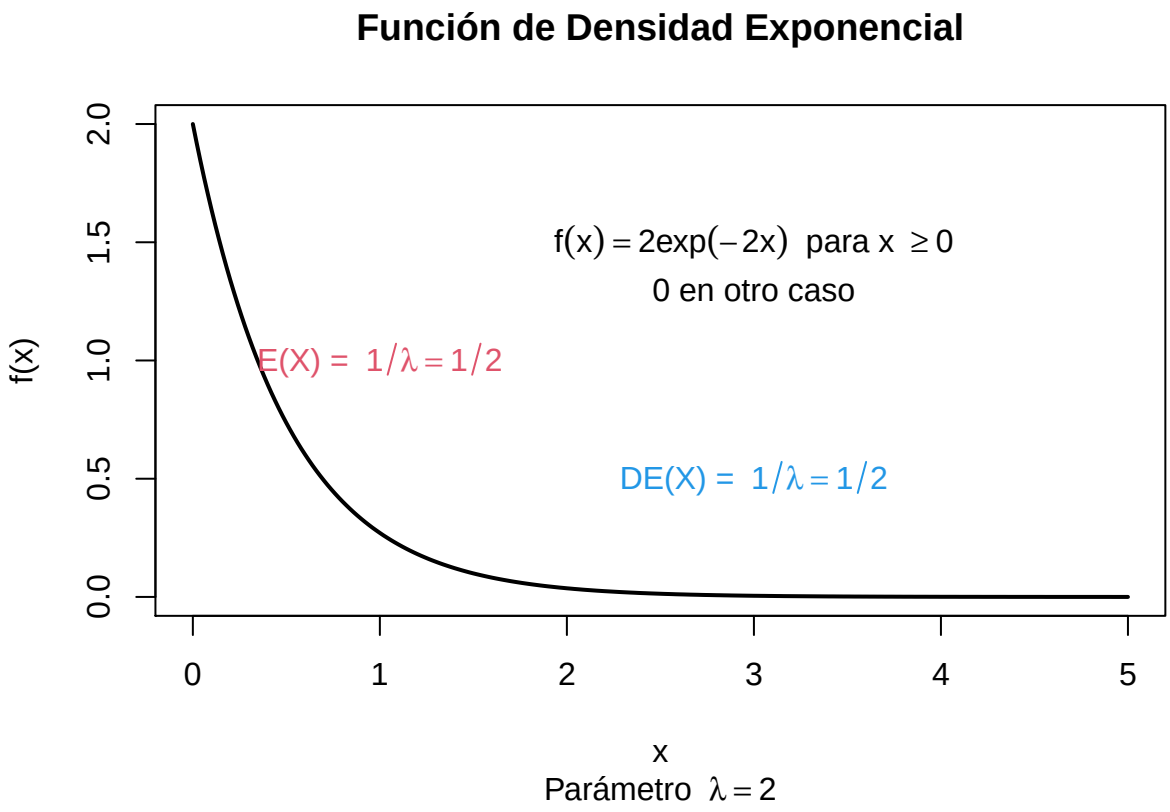
##DESARROLLO

El teorema central del límite (TCL) es una teoría estadística que establece que, dada una muestra suficientemente grande de la población, la distribución de las medias muestrales seguirá una distribución normal.

Además, el TCL afirma que a medida que el tamaño de la muestra se incrementa, la media muestral se acercará a la media de la población. Por tanto, mediante el TCL podemos definir la distribución de la media muestral de una determinada población con una varianza conocida. De manera que la distribución seguirá una distribución normal si el tamaño de la muestra es lo suficientemente grande.

Cargamos el paquete ggplot2 para hacer algunas gráficas

```r
#TEOREMA CENTRAL DEL LIMITE


library(ggplot2)
# Consideremos una variable aleatoria (v.a.) X
# con distribución exponencial y parámetro lambda = 2
x <- seq(0, 5, 0.02)
plot(x, dexp(x, rate = 2), type = "l", lwd = 2, ylab = "")
title(main = "Función de Densidad Exponencial", ylab = "f(x)",
      sub = expression("Parámetro " ~ lambda == 2))
text(x = 3, y = 1.5, labels = expression(f(x)==2*exp(-2*x) ~ " para x "  >= 0))
text(x = 3, y = 1.3, labels = paste("0 en otro caso"))
text(x = 1, y = 1, labels = expression("E(X) = " ~ 1/lambda == 1/2), col = 2)
text(x = 3, y = 0.5, labels = expression("DE(X) = " ~ 1/lambda == 1/2), col = 4)
```




```r
print("Ahora obtenemos una muestra aleatoria de tamaño n = 4 de la distribución exponencial considerada")
```

```
[1] "Ahora obtenemos una muestra aleatoria de tamaño n = 4 de la distribución exponencial considerada"
```

```r
set.seed(10) # Para reproducir posteriormente la muestra
(m1.4 <- rexp(n = 4, rate = 2))
```

```
[1] 0.007478203 0.460110602 0.376079469 0.787520925
```

```r
print("Obtenemos la media de la muestra generada")
```

```
[1] "Obtenemos la media de la muestra generada"
```

```r
mean(m1.4)
```

```
[1] 0.4077973
```

```r
print("Ahora obtenemos 5 muestras de tamaño 3")
```

```
[1] "Ahora obtenemos 5 muestras de tamaño 3"
```

```r
set.seed(64) # Para reproducir las muestras en el futuro
(m5.3 <- sapply(X = rep(3, 5), FUN = rexp, 2))
```

```
          [,1]        [,2]       [,3]      [,4]       [,5]
[1,] 1.5874413 0.031138339 0.08650897 0.3297026 0.82773066
[2,] 0.3272063 0.120168796 0.59122902 0.1333955 0.30709894
[3,] 1.5801568 0.006056407 1.08174334 0.5965121 0.02107677
```

```r
print("Obtenemos las medias de las 5 muestras")
```

```
[1] "Obtenemos las medias de las 5 muestras"
```

```r
(media5.3 <- apply(m5.3, 2, mean))
```

```
[1] 1.16493482 0.05245451 0.58649378 0.35320341 0.38530212
```

```r
print("Ahora obtenemos 1000 muestras de tamaño 7 y las 1000 medias correspondientes a las muestras")
```

```
[1] "Ahora obtenemos 1000 muestras de tamaño 7 y las 1000 medias correspondientes a las muestras"
```

```r
set.seed(465) # Para reproducir las muestras en el futuro
m1000.7 <- sapply(X = rep(7, 1000), FUN = rexp, 2)
media1000.7 <- apply(m1000.7, 2, mean)
mdf <- as.data.frame(media1000.7)
tail(mdf)
```

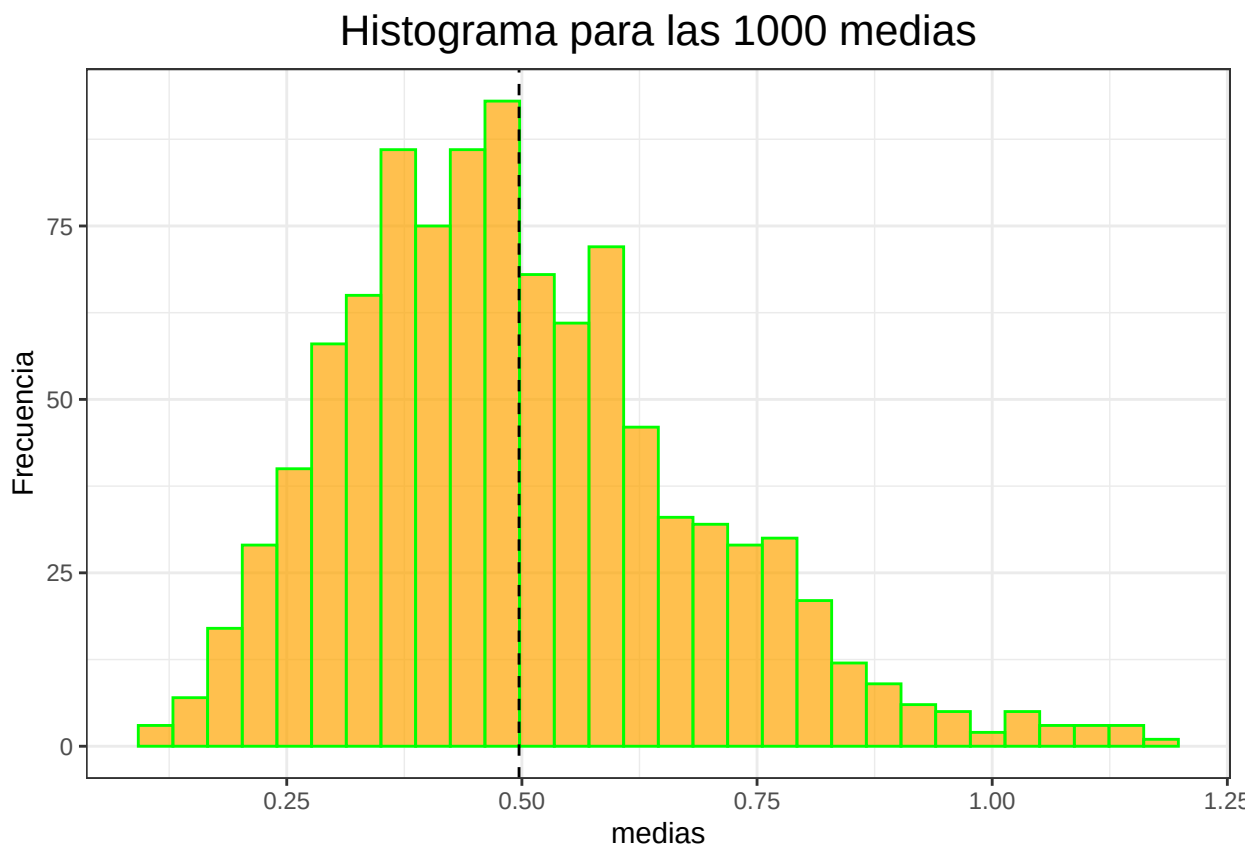
```
     media1000.7
995    0.5444073
996    0.3751581
997    0.5589332
998    0.6565002
999    0.2337858
1000   0.3259742
```

```
print("Observamos que el histograma de las medias tiene forma de campana")
```

```
[1] "Observamos que el histograma de las medias tiene forma de campana"
```

```
ggplot(mdf, aes(media1000.7)) +
  geom_histogram(colour = 'green',
                 fill = 'orange',
                 alpha = 0.7) + # Intensidad del color fill
  geom_vline(xintercept = mean(media1000.7), linetype="dashed", color = "black") +
  ggtitle('Histograma para las 1000 medias') +
  labs(x = 'medias', y = 'Frecuencia')+
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5, size = 16))
```

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




```
mean(media1000.7); 1/2 # Media de las 1000 medias y media de
```

```
[1] 0.4969754
```

```
[1] 0.5
```

```r
# la población de la cual vienen las 1000 muestras
sd(media1000.7); (1/2)/sqrt(7) # DE de las 1000 medias y DE
```

```
[1] 0.1854891
```

```
[1] 0.1889822
```

```r
# de la población de la cual vienen las 1000 muestras dividida
# por la raíz del tamaño de la muestra

print("Ahora obtenemos 1000 muestras de tamaño 33 y las 1000 medias correspondientes a las muestras")
```

```
[1] "Ahora obtenemos 1000 muestras de tamaño 33 y las 1000 medias correspondientes a las muestras"
```

```r
set.seed(4465) # Para reproducir las muestras en el futuro
m1000.33 <- sapply(X = rep(33, 1000), FUN = rexp, 2)
media1000.33 <- apply(m1000.33, 2, mean)
mdf <- as.data.frame(media1000.33)
tail(mdf)
```

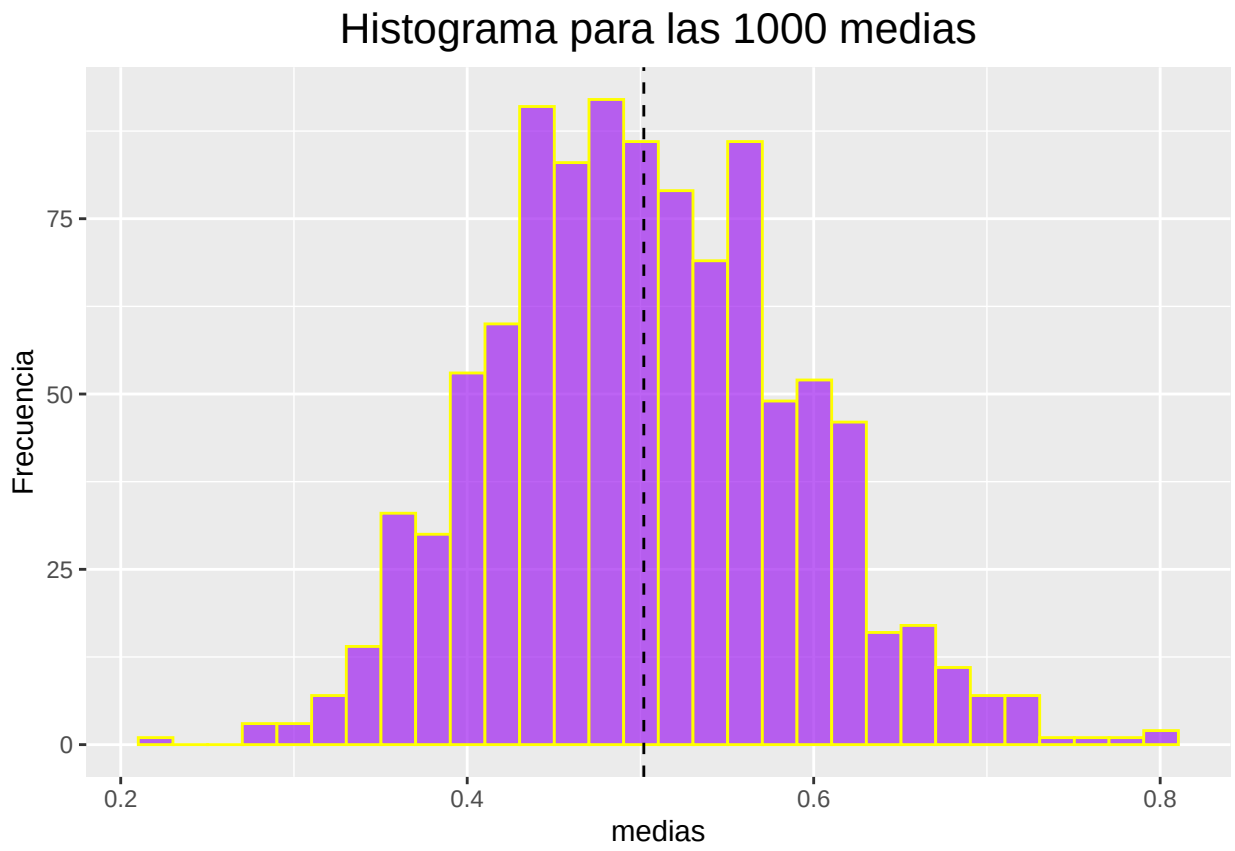
```
     media1000.33
995     0.3818621
996     0.3609060
997     0.5153507
998     0.5261520
999     0.5053655
1000    0.4573147
```

```r
print("Observamos que el histograma de las medias es más parecida todavía a una campana")
```

```
[1] "Observamos que el histograma de las medias es más parecida todavía a una campana"
```

```r
ggplot(mdf, aes(media1000.33)) +
  geom_histogram(colour = 'yellow',
                 fill = 'purple',
                 alpha = 0.7) + # Intensidad del color fill
  geom_vline(xintercept = mean(media1000.33), linetype="dashed", color = "black") +
  ggtitle('Histograma para las 1000 medias') +
  labs(x = 'medias', y = 'Frecuencia')+
  theme_get() +
  theme(plot.title = element_text(hjust = 0.5, size = 16))
```

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




```
mean(media1000.33); 1/2 # Media de las 1000 medias y media de la
```

```
[1] 0.501952
```

```
[1] 0.5
```

```
# población de la cual vienen las 1000 muestras
sd(media1000.33); (1/2)/sqrt(33) # DE de las 1000 medias
```

```
[1] 0.08624136
```

```
[1] 0.08703883
```

```
# y DE de la población de la cual vienen las 1000 muestras dividida
# por la raíz del tamaño de la muestra

print("Ahora obtenemos 1000 muestras de tamaño 400 y las 1000 medias correspondientes a las muestras")
```

```
[1] "Ahora obtenemos 1000 muestras de tamaño 400 y las 1000 medias correspondientes a las muestras"
```

```
set.seed(543465) # Para reproducir las muestras en el futuro
m1000.400 <- sapply(X = rep(400, 1000), FUN = rexp, 2)
media1000.400 <- apply(m1000.400, 2, mean)
mdf <- as.data.frame(media1000.400)
tail(mdf)
```

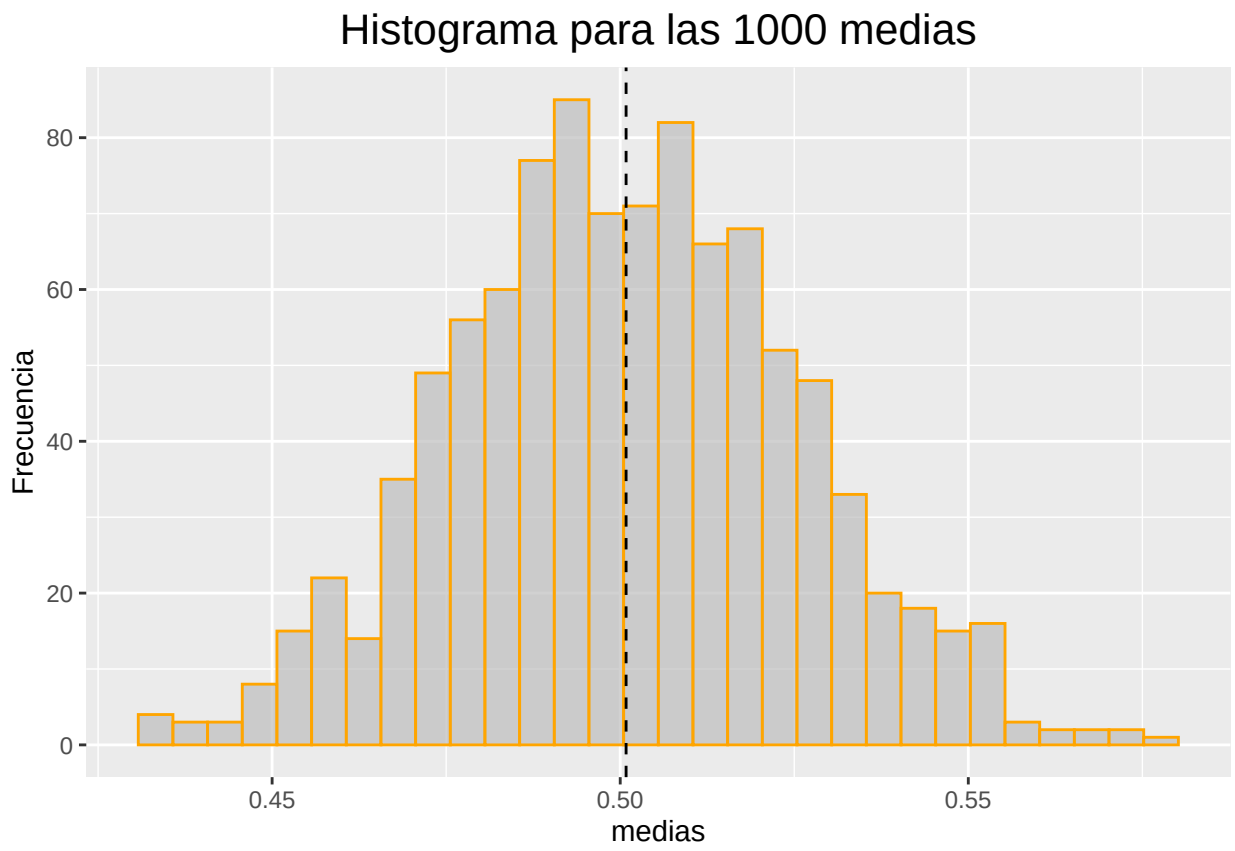
```
     media1000.400
995      0.4656883
996      0.4779040
997      0.4803765
998      0.4944309
999      0.4988992
1000     0.5106146
```

```
print("Observamos que el histograma de las medias es más parecida todavía a una campana")
```

```
[1] "Observamos que el histograma de las medias es más parecida todavía a una campana"
```

```
ggplot(mdf, aes(media1000.400)) +
  geom_histogram(colour = 'orange',
                 fill = 'gray',
                 alpha = 0.7) + # Intensidad del color fill
  geom_vline(xintercept = mean(media1000.400), linetype="dashed", color = "black") +
  ggtitle('Histograma para las 1000 medias') +
  labs(x = 'medias', y = 'Frecuencia')+
  theme_gray() +
  theme(plot.title = element_text(hjust = 0.5, size = 16))
```

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



```r
mean(media1000.400); 1/2 # Media de las 1000 medias y media
```

```
[1] 0.5008446
```

```
[1] 0.5
```

```r
# de la población de la cual vienen las 1000 muestras
sd(media1000.400); (1/2)/sqrt(400) # DE de las 1000 medias
```

```
[1] 0.02453529
```

```
[1] 0.025
```

```r
# y DE de la población de la cual vienen las 1000 muestras
# dividida por la raíz del tamaño de la muestra
```

EJEMPLO 3. ALGUNOS ESTIMADORES PUNTUALES INSESGADOS COMUNES

OBJETIVO

- Entender la idea de estimador insesgado de un parámetro

REQUISITOS

- Tener R y RStudio instalados
- Hbaer leido el prework

DESARROLLO

Un estimador insesgado es aquel cuya esperanza matemática coincide con el valor del parámetro que sea desea estimar. En caso de no coincidir se dice que el estimador tiene sesgo. La razón de buscar un estimador insesgado es que el parámetro que deseamos estimar esté bien estimado. Es decir, si queremos estimar la media de goles por partido de determinado jugador de fútbol, hemos de utilizar una fórmula que nos proporcione un valor lo más aproximado posible al valor real.

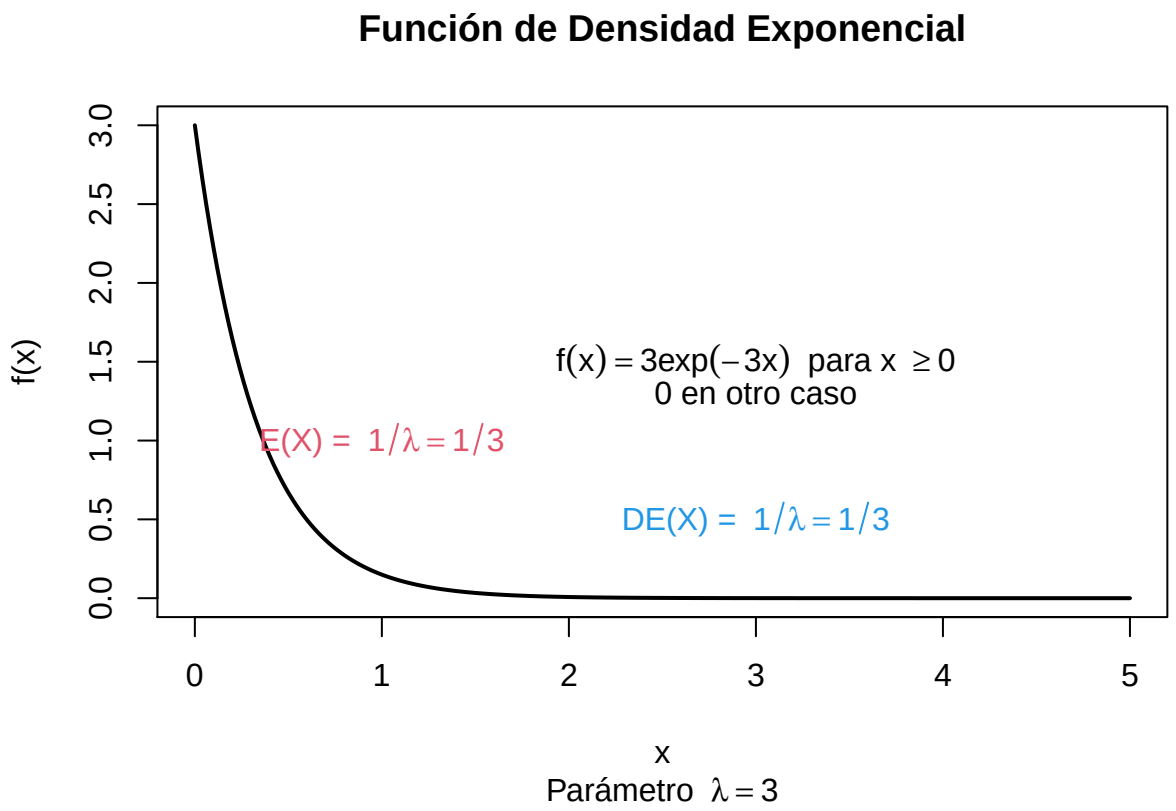
```r
#EJEMPLO 3

print("vargamos el paquete ggplot2 para hacer algunas gráficas")
```

```
[1] "vargamos el paquete ggplot2 para hacer algunas gráficas"
```

```r
library(ggplot2)

print("Consideremos una variable aleatoria X con distribución exponencial y parametro lambda=3")
```

```
[1] "Consideremos una variable aleatoria X con distribución exponencial y parametro lambda=3"
```

```r
x <- seq(0, 5, 0.02)
plot(x, dexp(x, rate = 3), type = "l", lwd = 2, ylab = "")
title(main = "Función de Densidad Exponencial", ylab = "f(x)",
      sub = expression("Parámetro " ~ lambda == 3))
text(x = 3, y = 1.5, labels = expression(f(x)==3*exp(-3*x) ~ " para x "  >= 0))
text(x = 3, y = 1.3, labels = paste("0 en otro caso"))
text(x = 1, y = 1, labels = expression("E(X) = " ~ 1/lambda == 1/3), col = 2)
text(x = 3, y = 0.5, labels = expression("DE(X) = " ~ 1/lambda == 1/3), col = 4)
```




```r
print("Obtenemos 1200 muestras aleatorias de tamaño 350 y las 1200 medias correspondientes a las muestra
```

```
[1] "Obtenemos 1200 muestras aleatorias de tamaño 350 y las 1200 medias correspondientes a las muestras
```

```r
set.seed(65) # Para reproducir las muestras en el futuro
m1200.350 <- sapply(X = rep(350, 1200), FUN = rexp, rate = 3)
media1200.350 <- apply(m1200.350, 2, mean)
mdf <- as.data.frame(media1200.350)
tail(mdf)
```

```
     media1200.350
1195     0.3757215
1196     0.3415659
1197     0.3043333
1198     0.3233880
```

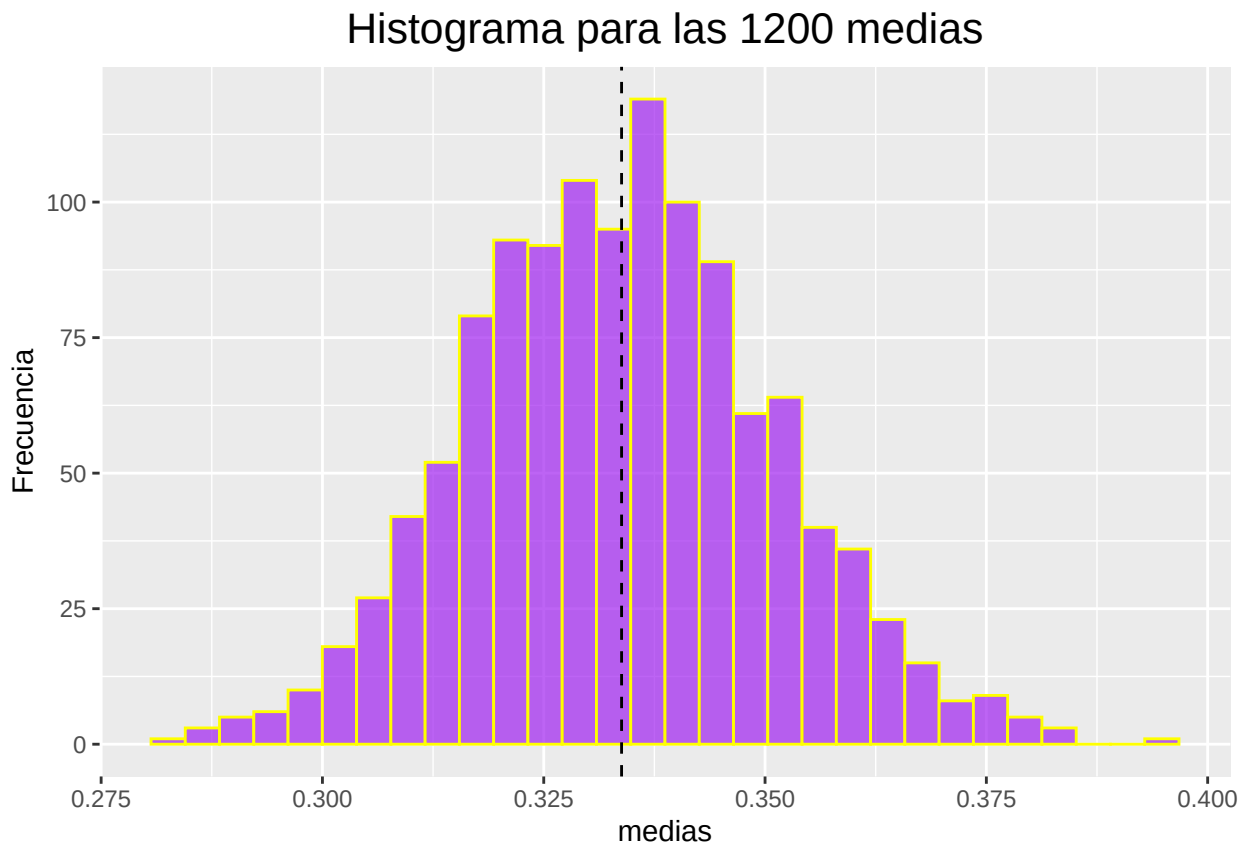
```
1199      0.3240008
1200      0.3344396
```

```
print("Observamos que el histograma de las medias tiene forma de campana")
```

```
[1] "Observamos que el histograma de las medias tiene forma de campana"
```

```
ggplot(mdf, aes(media1200.350)) +
  geom_histogram(colour = 'yellow',
                 fill = 'purple',
                 alpha = 0.7) + # Intensidad del color fill
  geom_vline(xintercept = mean(media1200.350),
             linetype="dashed",
             color = "black") +
  ggtitle('Histograma para las 1200 medias') +
  labs(x = 'medias',
       y = 'Frecuencia')+
  theme_get() +
  theme(plot.title = element_text(hjust = 0.5, size = 16))
```

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



```r
mean(media1200.350); 1/3 # Media de las 1200 medias y media de
```

```
[1] 0.3337942
```

```
[1] 0.3333333
```

```r
# la población de la cual provienen las 1200 muestras
sd(media1200.350); (1/3)/sqrt(350) # DE de las 1200 medias y DE
```

```
[1] 0.01714254
```

```
[1] 0.01781742
```

```r
# de la población de la cual provienen las 1200 muestras dividida
# por la raíz del tamaño de las muestras




print("ENSAYO BERNOULLI Con las siguientes instrucciones obtenemos un solo valor, en donde el 0 (fracas
```

```
[1] "ENSAYO BERNOULLI Con las siguientes instrucciones obtenemos un solo valor, en donde el 0 (fracaso)
```

```r
set.seed(345)
sample(x = c(0, 1), size = 1, prob = c(0.45, 0.55))
```

```
[1] 1
```

```r
rbinom(n = 1, size = 1, prob = 0.55)
```

```
[1] 1
```

```r
print("Obtenemos 1000 muestras de tamaño 31 de una v.a. Bernoulli con p = 0.55")
```

```
[1] "Obtenemos 1000 muestras de tamaño 31 de una v.a. Bernoulli con p = 0.55"
```

```r
set.seed(5434) # Para reproducir las muestras en el futuro
m1000.31 <- sapply(X = rep(31, 1000), FUN = function(n) sample(x = c(0, 1), size = n, replace = TRUE, pr
media1000.31 <- apply(m1000.31, 2, mean)
mdf <- as.data.frame(media1000.31)
tail(mdf)
```

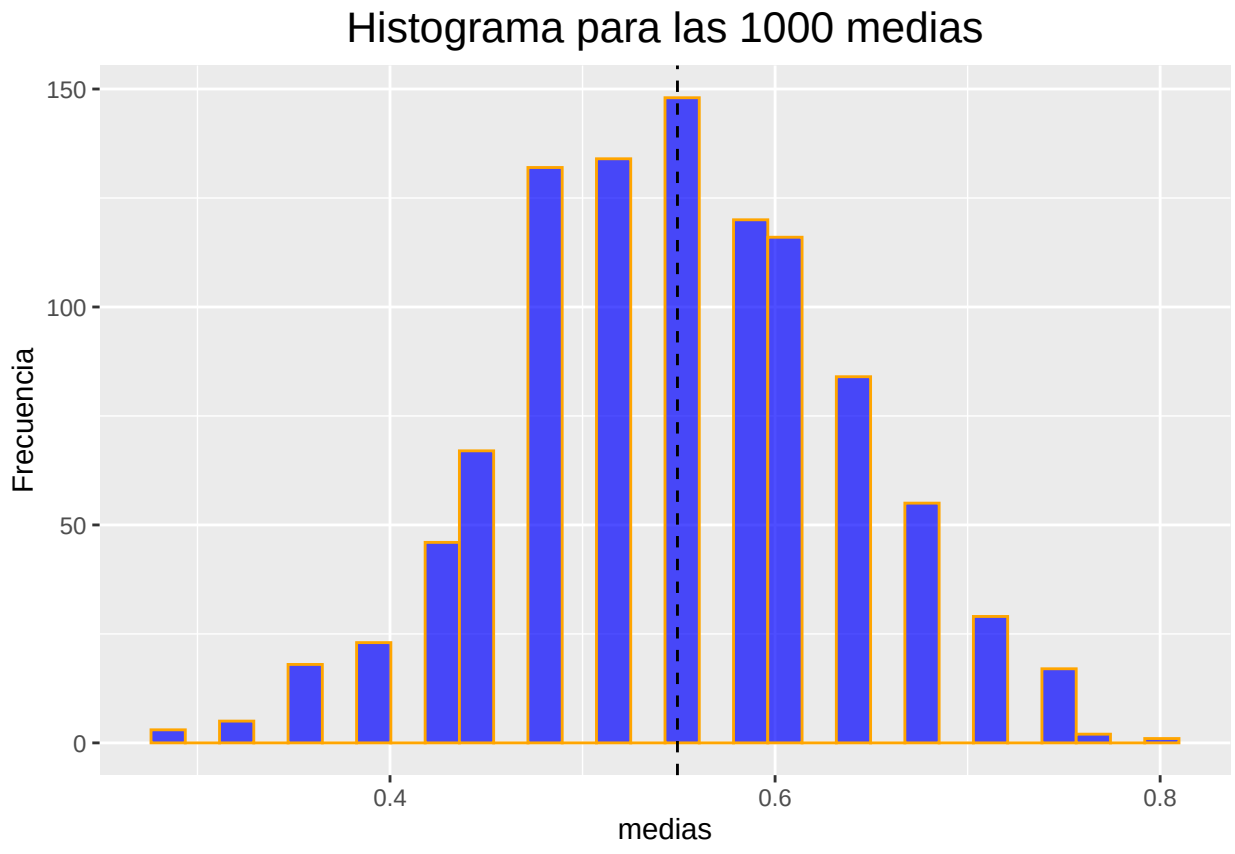
```
     media1000.31
995     0.6451613
996     0.6129032
997     0.6774194
998     0.5161290
999     0.5806452
1000    0.5161290
```

```r
print("Observamos que el histograma de las medias es parecida a una campana")
```

```
[1] "Observamos que el histograma de las medias es parecida a una campana"
```

```r
ggplot(mdf, aes(media1000.31)) +
  geom_histogram(colour = 'orange',
                 fill = 'blue',
                 alpha = 0.7) + # Intensidad del color fill
  geom_vline(xintercept = mean(media1000.31),
             linetype="dashed",
             color = "black") +
  ggtitle('Histograma para las 1000 medias') +
  labs(x = 'medias',
       y = 'Frecuencia')+
  theme_grey() +
  theme(plot.title = element_text(hjust = 0.5, size = 16))
```

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




```r
mean(media1000.31); 0.55 # Media de las 1000 medias y media de la población de la cual provienen las 10
```

```
[1] 0.5492903
```

```
[1] 0.55
```

```
sd(media1000.31); sqrt(0.55*0.45)/sqrt(31) # DE de las 1000 medias y DE
```

```
[1] 0.08767195
```

```
[1] 0.08935251
```

```
# de la población de la  cual provienen las 1000 muestras dividida
# por la raíz del tamaño de la muestra


print("Obtenemos 1150 muestras aleatorias de tamaño n1 = 54 de una distribución exponencial con parámet
```

```
[1] "Obtenemos 1150 muestras aleatorias de tamaño n1 = 54 de una distribución exponencial con parámetro
```

```
set.seed(65) # Para reproducir las muestras en el futuro
m1150.54 <- sapply(X = rep(54, 1150), FUN = rexp, rate = 3.2)
media1150.54 <- apply(m1150.54, 2, mean)
m1150.41 <- sapply(X = rep(41, 1150), FUN = rexp, rate = 1.5)
media1150.41 <- apply(m1150.41, 2, mean)
dif.medias <- media1150.54 - media1150.41 # Diferencia de medias
dmdf <- as.data.frame(dif.medias)
tail(dmdf)
```

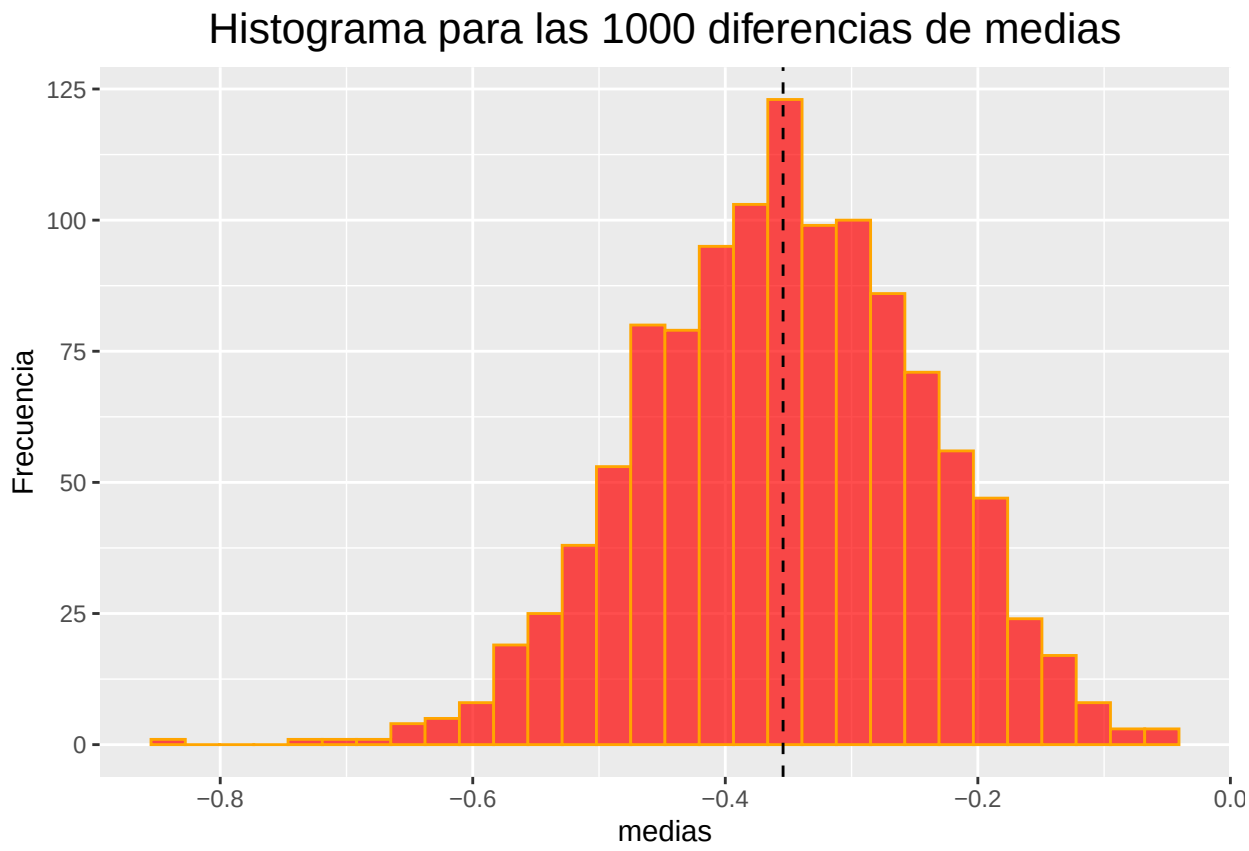
```
     dif.medias
1145 -0.3322635
1146 -0.4433986
1147 -0.2908058
1148 -0.3414530
1149 -0.1469445
1150 -0.4128351
```

```
print("Observamos que el histograma de las diferencias de medias es parecida a una campana")
```

```
[1] "Observamos que el histograma de las diferencias de medias es parecida a una campana"
```

```
ggplot(dmdf, aes(dif.medias)) +
  geom_histogram(colour = 'orange',
                 fill = 'red',
                 alpha = 0.7) + # Intensidad del color fill
  geom_vline(xintercept = mean(dif.medias), linetype="dashed", color = "black") +
  ggtitle('Histograma para las 1000 diferencias de medias') +
  labs(x = 'medias', y = 'Frecuencia')+
  theme_grey() +
  theme(plot.title = element_text(hjust = 0.5, size = 16))
```

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




```
mean(dif.medias); 1/3.2-1/1.5 # Media de las 1150 diferencias de medias y
```

```
[1] -0.3542439
```

```
[1] -0.3541667
```

```
# diferencia de medias de las poblaciones de las cuales provienen
# las 11500 muestras
sd(dif.medias); sqrt((1/3.2^2)/54 + (1/1.5^2)/41) # DE de las 1150
```

```
[1] 0.110714
```

```
[1] 0.1124658
```

```
# diferencias de medias y DE dada en literatura

print("Obtenemos 1100 muestras de tamaño n1 = 48 de una v.a. Bernoulli con p1 = 0.65 y otras 1100 muest
```

```
[1] "Obtenemos 1100 muestras de tamaño n1 = 48 de una v.a. Bernoulli con p1 = 0.65 y otras 1100 muestras
```

```
set.seed(7434) # Para reproducir las muestras en el futuro
m1100.48 <- sapply(X = rep(48, 1100), FUN = function(n) sample(x = c(0, 1), size = n, replace = TRUE, p
m1100.35 <- sapply(X = rep(35, 1100), FUN = function(n) sample(x = c(0, 1), size = n, replace = TRUE, p
```

```r
media1100.48 <- apply(m1100.48, 2, mean)
media1100.35 <- apply(m1100.35, 2, mean)
dif.medias <- media1100.48 - media1100.35
dmdf <- as.data.frame(dif.medias)
tail(dmdf)
```

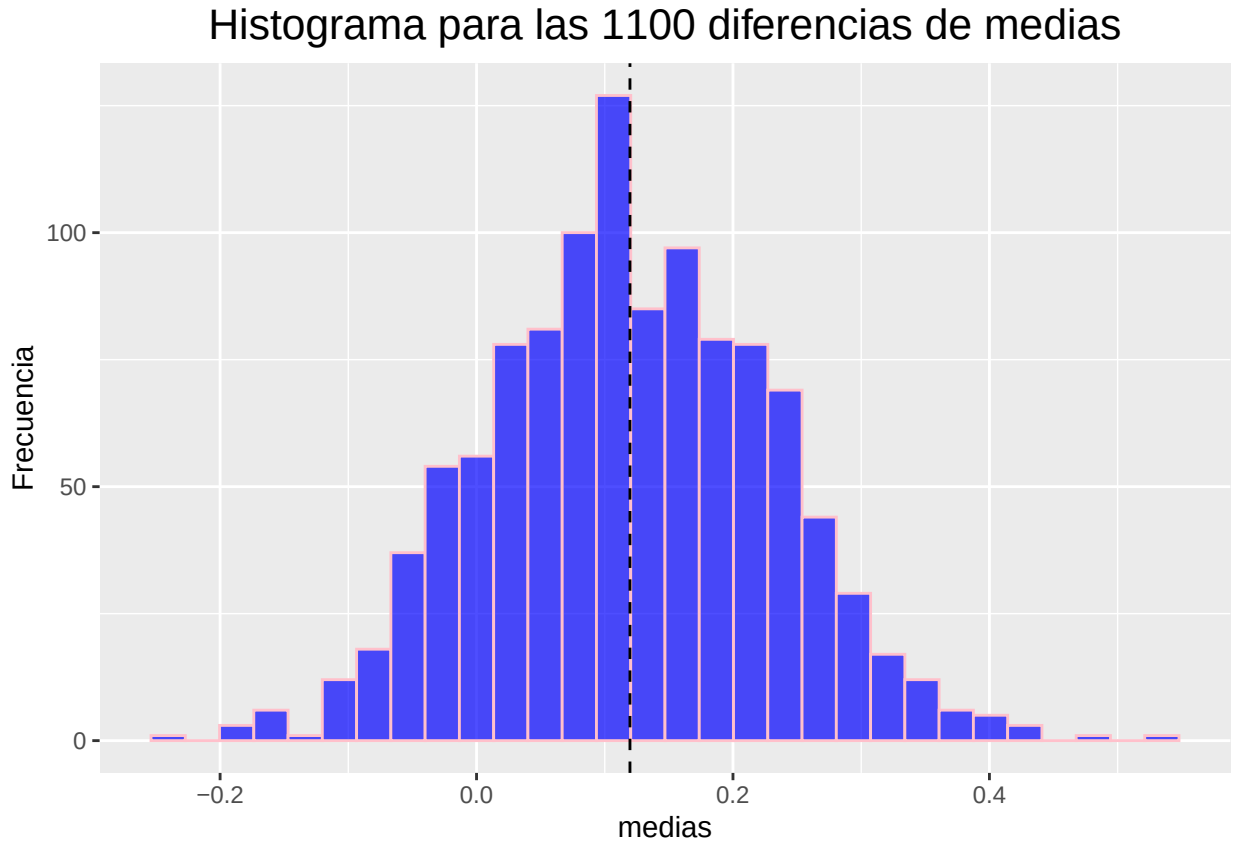
```
     dif.medias
1095 0.05357143
1096 0.07202381
1097 0.37321429
1098 0.20952381
1099 0.25119048
1100 0.09761905
```

```r
print("Observamos que el histograma de las diferencias de medias es parecida a una campana")
```

```
[1] "Observamos que el histograma de las diferencias de medias es parecida a una campana"
```

```r
ggplot(dmdf, aes(dif.medias)) +
  geom_histogram(colour = 'pink',
                 fill = 'blue',
                 alpha = 0.7) + # Intensidad del color fill
  geom_vline(xintercept = mean(dif.medias),
             linetype="dashed",
             color = "black") +
  ggtitle('Histograma para las 1100 diferencias de medias') +
  labs(x = 'medias',
       y = 'Frecuencia')+
  theme_grey() +
  theme(plot.title = element_text(hjust = 0.5, size = 16))
```

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




```
mean(dif.medias); 0.65 - 0.53 # Media de las 1100
```

```
[1] 0.1196558
```

```
[1] 0.12
```

```
#diferencias de medias y diferencia de medias de las poblaciones
# de las cuales provienen las muestras
sd(dif.medias); sqrt((0.65*0.35)/48 + (0.53*0.47)/35) # DE de las 1100
```

```
[1] 0.110132
```

```
[1] 0.1088886
```

```
# diferencias de medias y DE dada en literatura
```