Temario de la sesión

1. Probabilidad en R
    1.1. Funciones de probabilidad en R: convención `d`, `p`, `q`, `r`
    1.2. Distribuciones incluidas en R: discretas y continuas
    1.3. Ejemplos avanzados de problemas de probabilidad
    1.4. Probabilidad condicional: esperanza y varianza condicional
    1.5. Teorema Central del Límite (demostración con simulación en R)
    1.6. Ley de los Grandes Números (demostración con simulación en R)
2. Visualización de distribuciones y probabilidades con `ggplot2`
    2.1. Histogramas y densidades simuladas
    2.2. Superposición de densidades teóricas y empíricas
    2.3. Representación de distribuciones discretas
3. Estadística descriptiva en R
    3.1. Medidas de tendencia central, dispersión y posición
    3.2. Forma de la distribución
    3.3. Tablas de frecuencias y proporciones
4. Sumas aleatorias
    4.1. Distribución de sumas de variables independientes
    4.2. Ejemplos de aplicación: Poisson y Normal

1. Probabilidad con R

1.1. Funciones de probabilidad en R: convención d, p, q, r

En R, todas las distribuciones de probabilidad siguen la misma convención de nombres de funciones:

- `d*` : densidad o masa de probabilidad
- `p*` : función de distribución acumulada
- `q*` : cuantiles (función inversa de la acumulada)
- `r*` : generación aleatoria

Donde el sufijo * cambia según la distribución: norm (Normal), binom (Binomial), pois (Poisson), etc.

Ejemplo con la distribución normal estándar

```
# Densidad en x = 0
dnorm(0, mean = 0, sd = 1)
```

```
## [1] 0.3989423
```

```
# Probabilidad acumulada P(Z < 1.65)
pnorm(1.65, mean = 0, sd = 1)
```

```
## [1] 0.9505285
```

```
# Cuantil 95%
qnorm(0.95, mean = 0, sd = 1)
```

```
## [1] 1.644854
```

```
# 5 valores simulados de N(0,1)
rnorm(5, mean = 0, sd = 1)
```

```
## [1] -0.5155166 -1.5091724 -0.2514924  0.2909684  0.7776932
```

1.2. Distribuciones incluidas en R: discretas y continuas

R cuenta con una amplia colección de distribuciones de probabilidad implementadas. A continuación, se presenta un resumen de las más utilizadas.

Distribuciones Discretas

Distribución	Base	Parámetros	Ejemplo en R
Binomial	binom	size, prob	rbinom(10, size=10, prob=0.3)
Poisson	pois	lambda	rpois(10, lambda=4)
Geométrica	geom	prob	rgeom(10, prob=0.4)
Hipergeométrica	hyper	m, n, k	rhyper(10, 5, 7, 3)
Binomial negativa	nbinom	size, prob	rnbinom(10, size=5, prob=0.2)
Multinomial*	multinom	size, prob	rmultinom(5, size=10, prob=c(0.2,0.3,0.5))

- La distribución multinomial se maneja con rmultinom() únicamente para simulación.

Distribuciones continuas

Distribución	Base	Parámetros	Ejemplo en R
Normal	norm	mean, sd	rnorm(10, mean=0, sd=1)
Uniforme	unif	min, max	runif(10, min=0, max=1)
Exponencial	exp	rate	rexp(10, rate=0.5)
Gamma	gamma	shape, rate	rgamma(10, shape=2, rate=1)
Chi-cuadrado	chisq	df	rchisq(10, df=5)
t de Student	t	df	rt(10, df=10)
F de Fisher	f	df1, df2	rf(10, df1=5, df2=10)
Beta	beta	shape1, shape2	rbeta(10, 2, 5)
Weibull	weibull	shape, scale	rweibull(10, 2, 1)
Log-normal	lnorm	meanlog, sdlog	rlnorm(10, 0, 1)

Ejemplo

```r
set.seed(123) # Establecer una semilla para reproducir los resultados

# Simulamos tres distribuciones distintas mil veces
X_binom <- rbinom(10000, size=10, prob=0.3) # X ~ Bin(10,0.3)
X_pois  <- rpois(10000, lambda=4) # X ~ Po(4)
X_norm  <- rnorm(10000, mean=0, sd=1) # X ~ N (0,1)

# Medias empíricas vs teóricas
c(mean(X_binom), 10*0.3)
```

```
## [1] 2.9899 3.0000
```

```r
c(mean(X_pois), 4)
```

```
## [1] 3.9754 4.0000
```

```r
c(mean(X_norm), 0)
```

```
## [1] -0.009106453  0.000000000
```

1.3. Ejemplos avanzados de problemas de probabilidad

Aquí no solo se usan fórmulas cerradas, sino que se combinan las funciones de R (d, p, q, r) con simulaciones para verificar resultados.

Ejemplo 1: Tengo un seguro de automóviles y por cada 10 asegurados, 2 hacen una reclamación. Si tengo 100 asegurados, ¿cuál es la probabilidad de obtener al menos 30 reclamaciones?

Defino la v.a. X como el número de reclamaciones. Entonces $X \sim \text{Bin}(100, 0.2)$, y buscamos $\mathbb{P}(X \geq 30)$.

```r
# Probabilidad exacta P(X >= 30) para X ~ Bin(100, 0.2)
p_exacta <- pbinom(29, size = 100, prob = 0.2, lower.tail = FALSE)
p_exacta
```

```
## [1] 0.01124898
```

Ejemplo 2: Tengo un seguro de automóviles y por cada 10 asegurados, 2 hacen una reclamación. Si tengo 100 asegurados, ¿cuál es la probabilidad de obtener menos de 30 reclamaciones?

Notar que es muy parecido al problema anterior, la diferencia esta en la probilidad a calcular:

```
# Probabilidad exacta P(X < 30) para X ~ Bin(100, 0.2)
p_exacta <- pbinom(29, size = 100, prob = 0.2, lower.tail = TRUE)
p_exacta
```

```
## [1] 0.988751
```

Esta es una forma de hacerlo pero también se puede hacer de otras maneras:

```
## Con un for
n <- 100
p <- 0.2
k_max <- 29
acum <- 0
for (k in 0:k_max) {
  acum <- acum + dbinom(k, size = n, prob = p)
}
acum
```

```
## [1] 0.988751
```

```
### Vectorizando
sum(dbinom(0:29, size = 100, prob = 0.2))
```

```
## [1] 0.988751
```

Ejemplo 3: Sea $X \sim Po(4)$. Calcular $\mathbb{E}[X|X \geq 3]$ y $\text{Var}(X|X \geq 3)$.

```
set.seed(123)
X <- rpois(1e6, lambda = 4)

E_cond <- mean(X[X >= 3])
Var_cond <- var(X[X >= 3])
c(E_cond, Var_cond)
```

```
## [1] 4.768926 2.639590
```

1.4. Probabilidad condicional: esperanza y varianza condicional

Recordando las formulas de la esperanza y varianza condicional:

Sea (X, Y) un par de v.a. (discretas o continuas). Recordatorio clave:

- $\mathbb{E}[\,X\,] \;=\; \mathbb{E}\big[\,\mathbb{E}[\,X \mid Y\,]\big]$
- $\text{Var}(X) \;=\; \mathbb{E}\big[\text{Var}(X \mid Y)\big] \;+\; \text{Var}\big(\mathbb{E}[X \mid Y]\big)$

Ejemplo

Para $X \sim \text{Poisson}(\lambda)$, al condicionar en $X \geq a$ se obtiene una distribución truncada a la izquierda. Podemos aproximar $\mathbb{E}[X \mid X \geq a]$ y $\text{Var}(X \mid X \geq a)$ por simulación:

```r
set.seed(123)
lambda <- 4
a <- 3
n <- 1e6
X <- rpois(n, lambda = lambda)
Xc <- X[X >= a]
c(
  E_cond  = mean(Xc),
  Var_cond = var(Xc)
)
```

```
##   E_cond Var_cond
## 4.768926 2.639590
```

Chequeo con ley total de varianza (descomposición respecto al evento $A = \{X \geq a\}$: $\text{Var}(X) = \mathbb{E}[\text{Var}(X \mid A)] + \text{Var}(\mathbb{E}[X \mid A])$, donde $X \mid A$ y $X \mid A^c$ son mezclas degeneradas/condicionadas por el evento. Verificación empírica:

```r
set.seed(123)
n <- 1e6
X <- rpois(n, lambda)
A <- (X >= a)
E_XA   <- mean(X[A])
E_XAc  <- mean(X[!A])
Var_XA <- var(X[A])
Var_XAc<- var(X[!A])
pA     <- mean(A)

Var_total_decomp <- pA*Var_XA + (1-pA)*Var_XAc + pA*(E_XA - mean(X))^2 + (1-pA)*(E_XAc - mean(X))^2
c(Var_empirica = var(X), Var_descomp = Var_total_decomp)
```

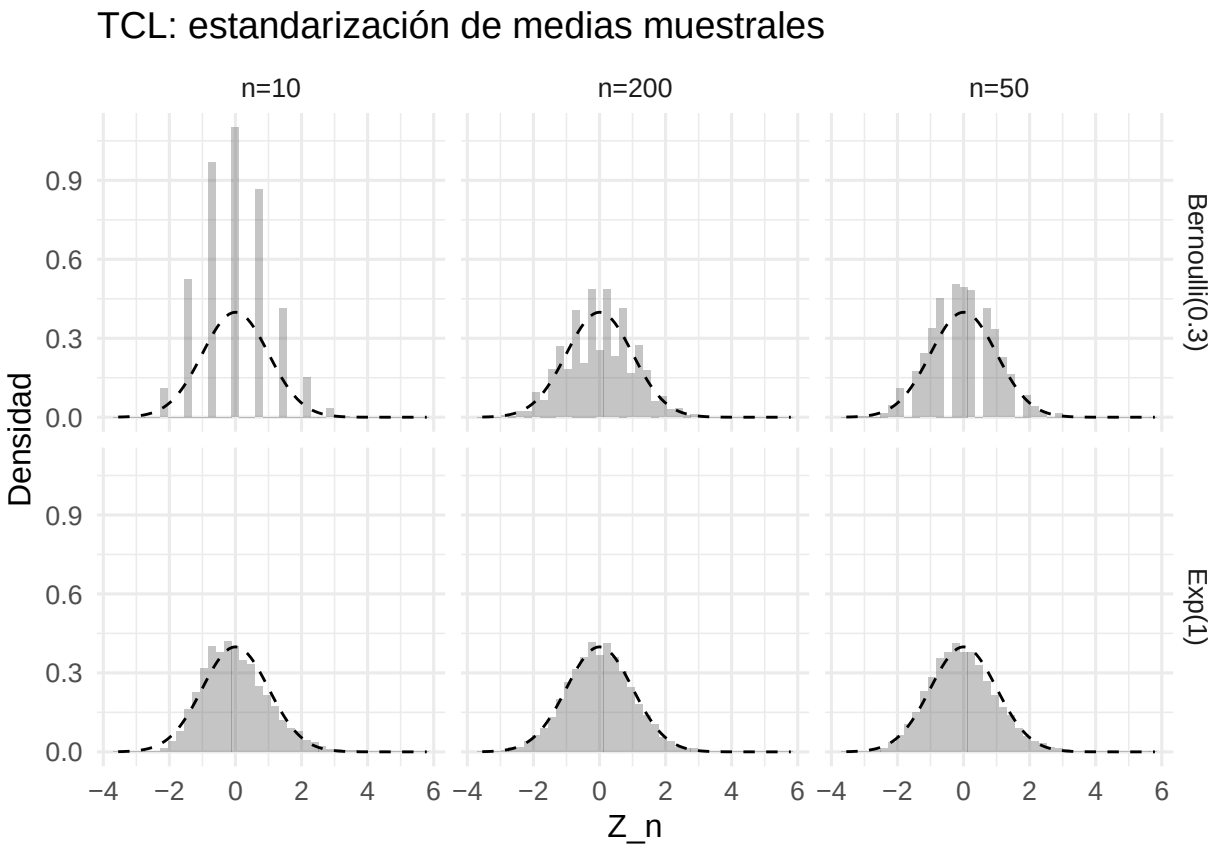
```
## Var_empirica  Var_descomp
##     4.002281     4.002280
```

1.5. Teorema Central del Límite (demostración con simulación en R)

Si $X_1, \ldots, X_n$ son i.i.d. con $\mathbb{E}[X_i] = \mu$ y $\text{Var}(X_i) = \sigma^2 \in (0, \infty)$, entonces $Z_n = \frac{\sqrt{n}(\bar{X}_n - \mu)}{\sigma} \xrightarrow{d} \text{N}(0,1)$.

```r
set.seed(123)
library(ggplot2)

simulate_Z <- function(rgen, mu, sigma, n, B = 5000) {
  xbar <- replicate(B, mean(rgen(n)))
  (sqrt(n) * (xbar - mu)) / sigma
}

# Exponencial(1): mu=1, sigma=1
Z_exp_10  <- simulate_Z(function(n) rexp(n, 1), mu = 1, sigma = 1, n = 10)
Z_exp_50  <- simulate_Z(function(n) rexp(n, 1), mu = 1, sigma = 1, n = 50)
Z_exp_200 <- simulate_Z(function(n) rexp(n, 1), mu = 1, sigma = 1, n = 200)

# Bernoulli(0.3): mu=0.3, sigma=sqrt(p(1-p))
p <- 0.3; sig <- sqrt(p*(1-p))
Z_ber_10  <- simulate_Z(function(n) rbinom(n, 1, p), mu = p, sigma = sig, n = 10)
Z_ber_50  <- simulate_Z(function(n) rbinom(n, 1, p), mu = p, sigma = sig, n = 50)
```

```r
Z_ber_200 <- simulate_Z(function(n) rbinom(n, 1, p), mu = p, sigma = sig, n = 200)

df <- rbind(
  data.frame(Z = Z_exp_10,  dist = "Exp(1)",     n = "n=10"),
  data.frame(Z = Z_exp_50,  dist = "Exp(1)",     n = "n=50"),
  data.frame(Z = Z_exp_200, dist = "Exp(1)",     n = "n=200"),
  data.frame(Z = Z_ber_10,  dist = "Bernoulli(0.3)", n = "n=10"),
  data.frame(Z = Z_ber_50,  dist = "Bernoulli(0.3)", n = "n=50"),
  data.frame(Z = Z_ber_200, dist = "Bernoulli(0.3)", n = "n=200")
)

ggplot(df, aes(Z)) +
  geom_histogram(aes(y = after_stat(density)), bins = 40, alpha = 0.35) +
  stat_function(fun = dnorm, args = list(mean = 0, sd = 1), linetype = "dashed") +
  facet_grid(dist ~ n) +
  labs(title = "TCL: estandarización de medias muestrales",
       x = "Z_n", y = "Densidad") +
  theme_minimal(base_size = 12)
```



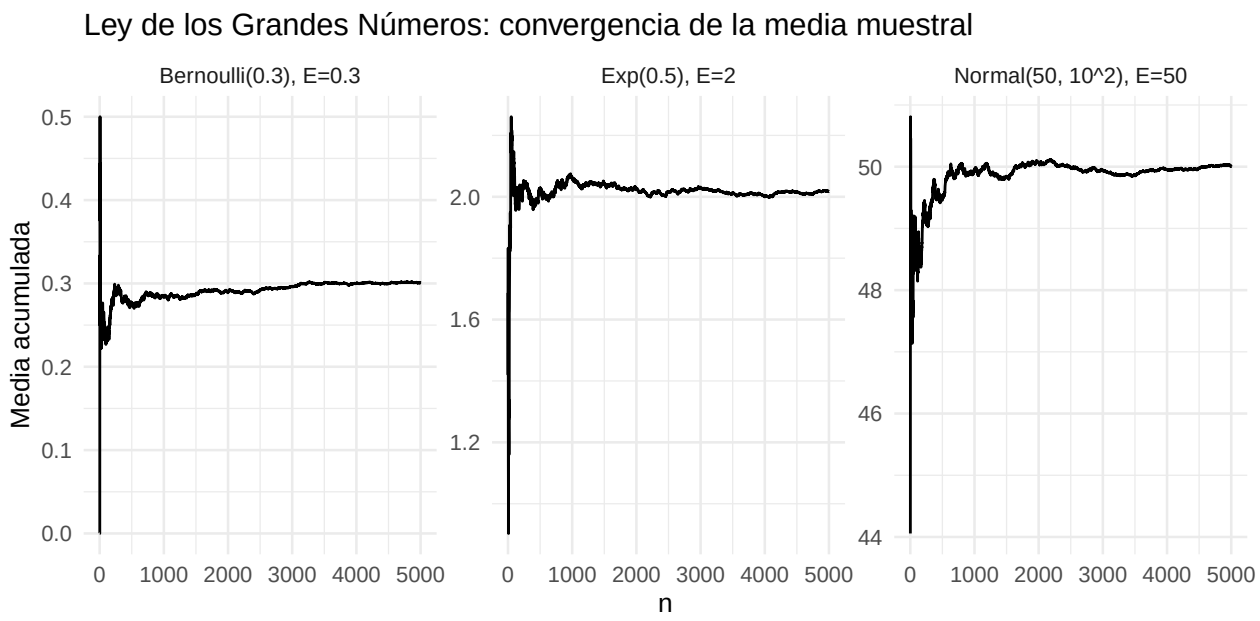
A medida que n crece, las distribuciones de Z_n se acercan a $\mathrm{N}(0,1)$, incluso partiendo de distribuciones sesgadas o discretas.

1.6. Ley de los Grandes Números (demostración con simulación en R)

Sea $X_1, X_2, \ldots$ una sucesión i.i.d. con $\mu = \mathbb{E}[X_1]$ y $\sigma^2 = \mathrm{Var}(X_1) < \infty$. La LGN débil establece que la media muestral $\bar{X}_n = \frac{1}{n}\sum_{i=1}^{n} X_i$ converge en probabilidad a μ: $\bar{X}_n \xrightarrow{p} \mu$.

```r
set.seed(123)
library(ggplot2)
```

```r
# Helper: trayectoria de medias acumuladas
medias_acum <- function(x) cumsum(x) / seq_along(x)

n <- 5000

# 1) Exponencial(rate=0.5): E[X]=2
x_exp <- rexp(n, rate = 0.5)
m_exp <- medias_acum(x_exp)

# 2) Bernoulli(p=0.3): E[X]=0.3
x_ber <- rbinom(n, 1, 0.3)
m_ber <- medias_acum(x_ber)

# 3) Normal(50, 10^2): E[X]=50
x_norm <- rnorm(n, mean = 50, sd = 10)
m_norm <- medias_acum(x_norm)

df <- rbind(
  data.frame(n = 1:n, media = m_exp, dist = "Exp(0.5), E=2"),
  data.frame(n = 1:n, media = m_ber, dist = "Bernoulli(0.3), E=0.3"),
  data.frame(n = 1:n, media = m_norm, dist = "Normal(50, 10^2), E=50")
)

ggplot(df, aes(n, media)) +
  geom_line(linewidth = 0.6) +
  facet_wrap(~ dist, scales = "free_y") +
  labs(title = "Ley de los Grandes Números: convergencia de la media muestral",
       x = "n", y = "Media acumulada") +
  theme_minimal(base_size = 12)
```



2. Visualización de distribuciones y probabilidades con ggplot2

La probabilidad y la estadística no solo se comprenden desde la teoría; su visualización permite contrastar la intuición con los resultados matemáticos. En R, el paquete ggplot2 es la herramienta más utilizada para construir gráficos claros, reproducibles y flexibles.

ggplot2 fue creado por Hadley Wickham en 2005 y está inspirado en The Grammar of Graphics de Leland Wilkinson. Esta "gramática de los gráficos" parte de la idea de que toda visualización puede descomponerse en elementos básicos:

- datos,
- mapeos estéticos (ejes, color, forma), y
- geometrías (barras, puntos, líneas, etc.).

De esta manera, ggplot2 ofrece un lenguaje coherente y modular que facilita construir desde gráficos simples hasta visualizaciones complejas, manteniendo consistencia y evitando código repetitivo.

En el contexto de este curso, utilizaremos ggplot2 para:

- Visualizar histogramas y densidades a partir de simulaciones.
- Comparar distribuciones empíricas (simuladas) con sus contrapartes teóricas.
- Representar distribuciones discretas como la Binomial y Poisson.

El objetivo es que el alumno pueda validar conceptos probabilísticos y verificar teoremas fundamentales mediante experimentos computacionales apoyados en gráficos.

2.1. Histogramas y densidades simuladas

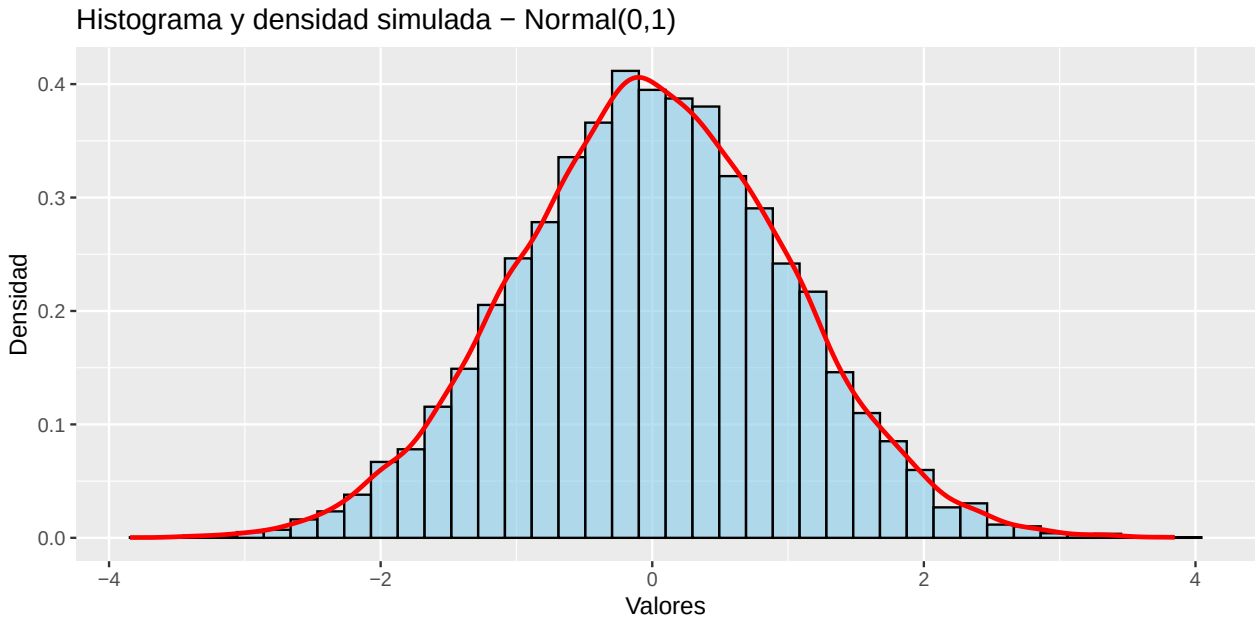
Un histograma es la primera herramienta básica para visualizar cómo se comporta una variable aleatoria. Al simular muchas veces una distribución conocida, podemos contrastar cómo la muestra se acerca a la forma teórica.

Distribución normal simulada

```
library(ggplot2)

# Simulación de 10,000 observaciones de una Normal(0,1)
set.seed(123)
muestra_norm <- data.frame(x = rnorm(10000, mean = 0, sd = 1))

# Gráfico: histograma + densidad suavizada
ggplot(muestra_norm, aes(x = x)) +
  geom_histogram(aes(y = ..density..), bins = 40, fill = "skyblue", color = "black", alpha = 0.6) +
  geom_density(color = "red", size = 1) +
  labs(title = "Histograma y densidad simulada - Normal(0,1)",
       x = "Valores", y = "Densidad")
```

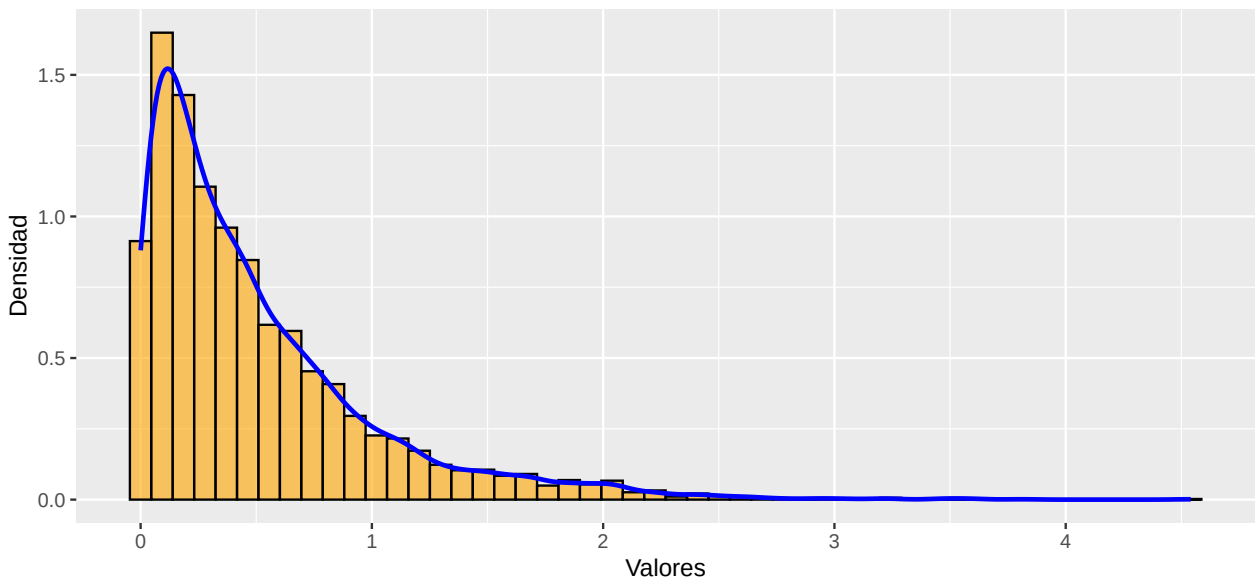
```
## Warning: Using 'size' aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use 'linewidth' instead.
## This warning is displayed once every 8 hours.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was
## generated.
```

```
## 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.
```



Distribución exponencial

```r
# Simulación de 5000 observaciones de una Exponencial(lambda = 2)
set.seed(123)
muestra_exp <- data.frame(x = rexp(5000, rate = 2))

ggplot(muestra_exp, aes(x = x)) +
  geom_histogram(aes(y = ..density..), bins = 50, fill = "orange", color = "black", alpha = 0.6) +
  geom_density(color = "blue", size = 1) +
  labs(title = "Histograma y densidad simulada - Exponencial(2)",
       x = "Valores", y = "Densidad")
```

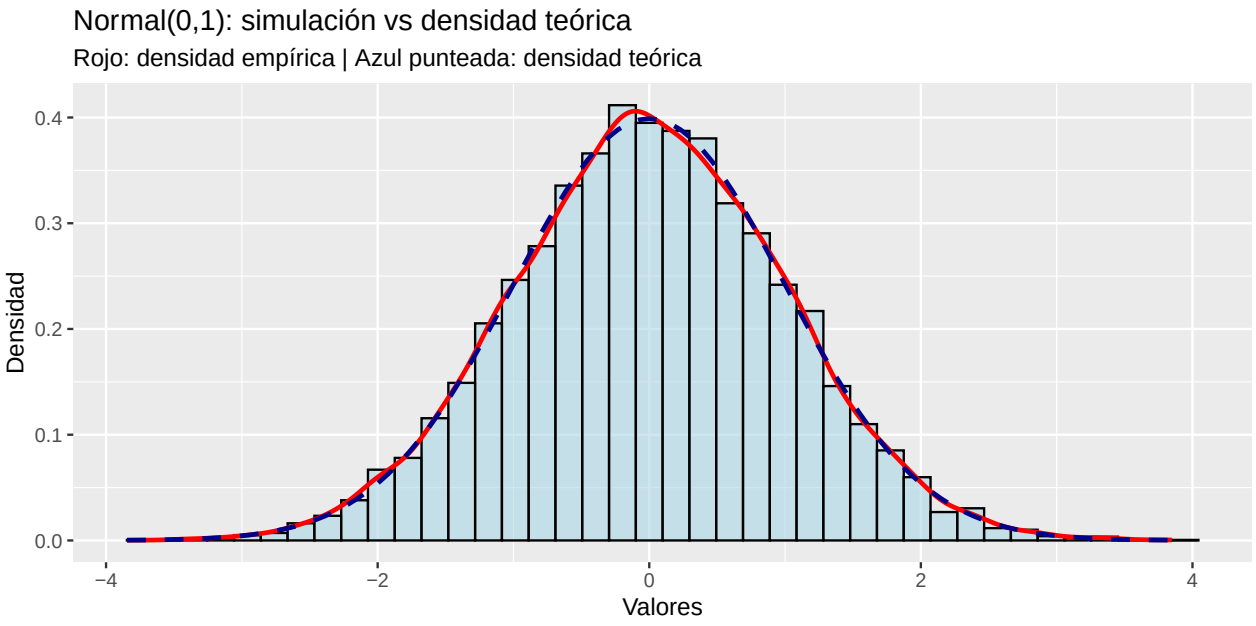


2.2. Superposición de densidades teóricas y empíricas

Un paso más allá de graficar histogramas es comparar directamente la muestra simulada con la función de densidad teórica. Esto permite confirmar visualmente que la simulación se ajusta a la distribución que modelamos.

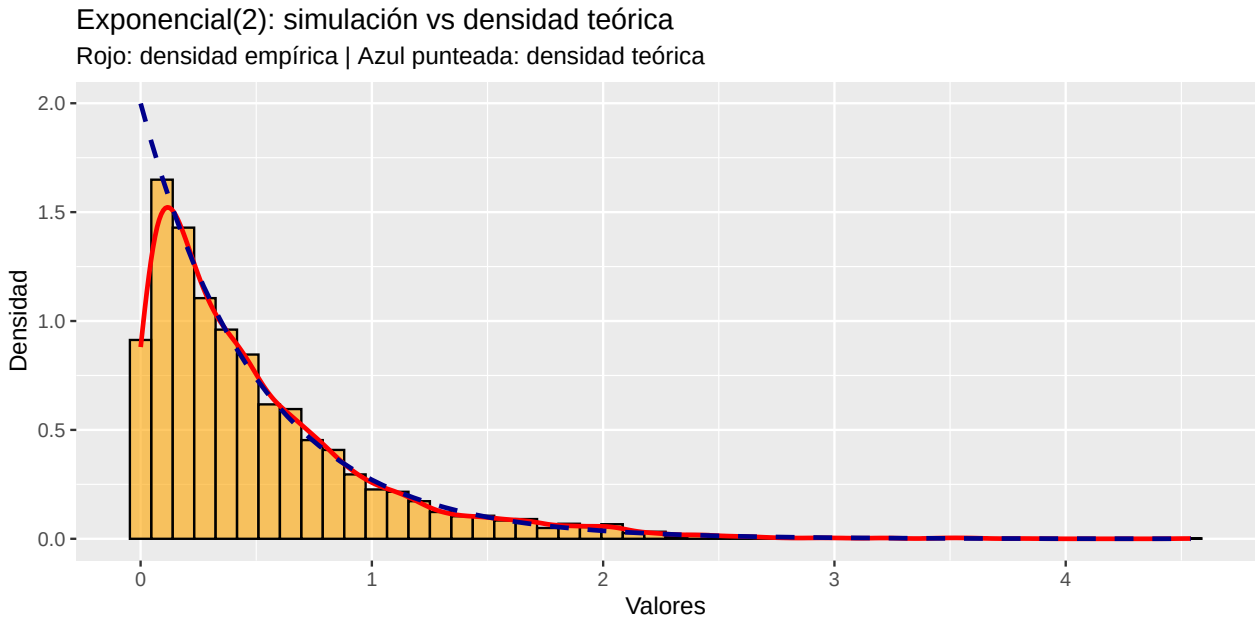
Ejemplo 1: Normal(0,1)

```
set.seed(123)
muestra_norm <- data.frame(x = rnorm(10000, mean = 0, sd = 1))

ggplot(muestra_norm, aes(x = x)) +
  geom_histogram(aes(y = ..density..), bins = 40, fill = "lightblue", color = "black", alpha = 0.6) +
  geom_density(color = "red", size = 1) +
  stat_function(fun = dnorm, args = list(mean = 0, sd = 1), color = "darkblue", size = 1, linetype = "da
  labs(title = "Normal(0,1): simulación vs densidad teórica",
       subtitle = "Rojo: densidad empírica | Azul punteada: densidad teórica",
       x = "Valores", y = "Densidad")
```



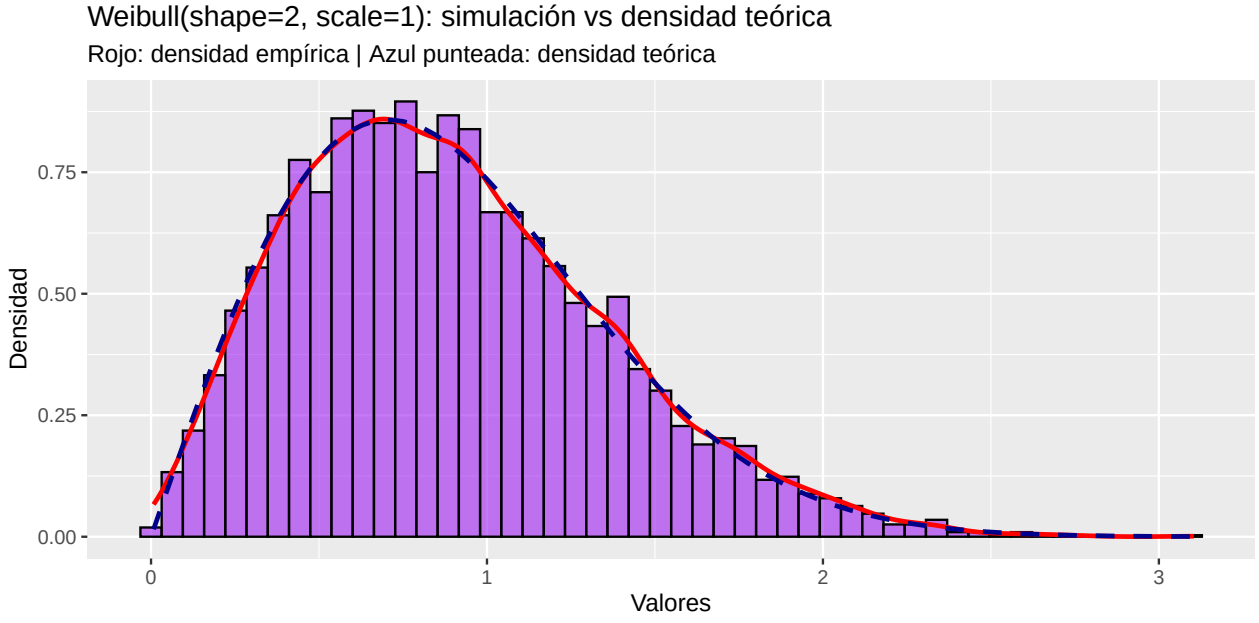
Ejemplo 2: Exponencial(2)

```
set.seed(123)
muestra_exp <- data.frame(x = rexp(5000, rate = 2))

ggplot(muestra_exp, aes(x = x)) +
  geom_histogram(aes(y = ..density..), bins = 50, fill = "orange", color = "black", alpha = 0.6) +
  geom_density(color = "red", size = 1) +
  stat_function(fun = dexp, args = list(rate = 2), color = "darkblue", size = 1, linetype = "dashed") +
  labs(title = "Exponencial(2): simulación vs densidad teórica",
       subtitle = "Rojo: densidad empírica | Azul punteada: densidad teórica",
       x = "Valores", y = "Densidad")
```



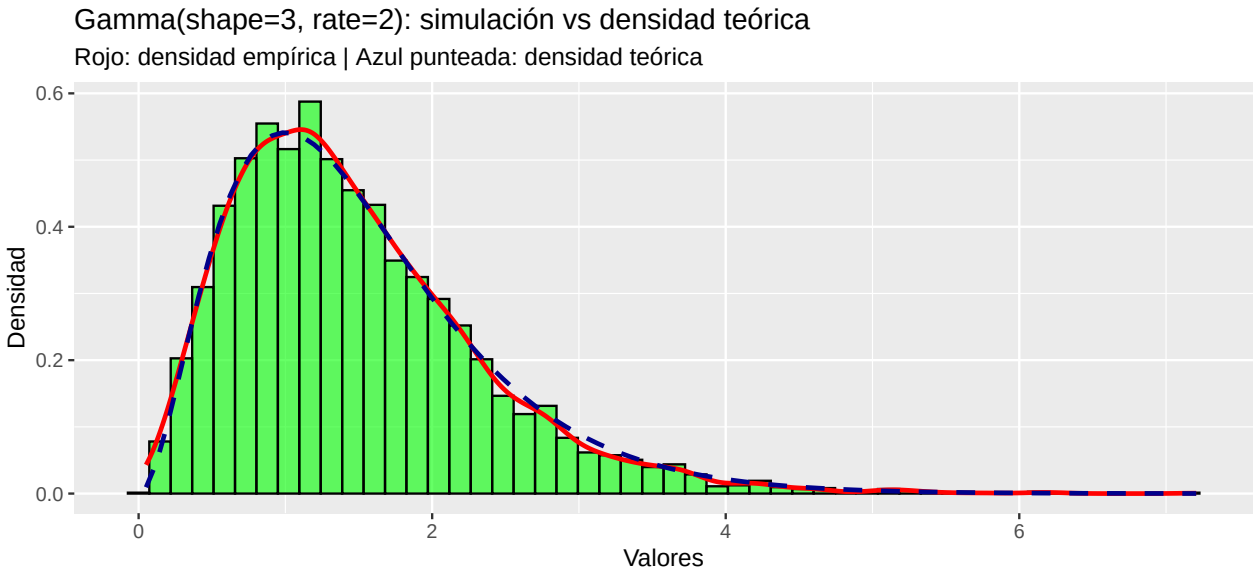
Ejemplo 3: Weibull(shape=2, scale=1)

```
set.seed(123)
muestra_weib <- data.frame(x = rweibull(5000, shape = 2, scale = 1))

ggplot(muestra_weib, aes(x = x)) +
  geom_histogram(aes(y = ..density..), bins = 50, fill = "purple", color = "black", alpha = 0.6) +
  geom_density(color = "red", size = 1) +
  stat_function(fun = dweibull, args = list(shape = 2, scale = 1), color = "darkblue", size = 1, linety
  labs(title = "Weibull(shape=2, scale=1): simulación vs densidad teórica",
       subtitle = "Rojo: densidad empírica | Azul punteada: densidad teórica",
       x = "Valores", y = "Densidad")
```



Ejemplo 4: Gamma(shape=3, rate=2)

```
set.seed(123)
muestra_gamma <- data.frame(x = rgamma(5000, shape = 3, rate = 2))

ggplot(muestra_gamma, aes(x = x)) +
  geom_histogram(aes(y = ..density..), bins = 50, fill = "green", color = "black", alpha = 0.6) +
  geom_density(color = "red", size = 1) +
  stat_function(fun = dgamma, args = list(shape = 3, rate = 2), color = "darkblue", size = 1, linetype =
  labs(title = "Gamma(shape=3, rate=2): simulación vs densidad teórica",
       subtitle = "Rojo: densidad empírica | Azul punteada: densidad teórica",
       x = "Valores", y = "Densidad")
```



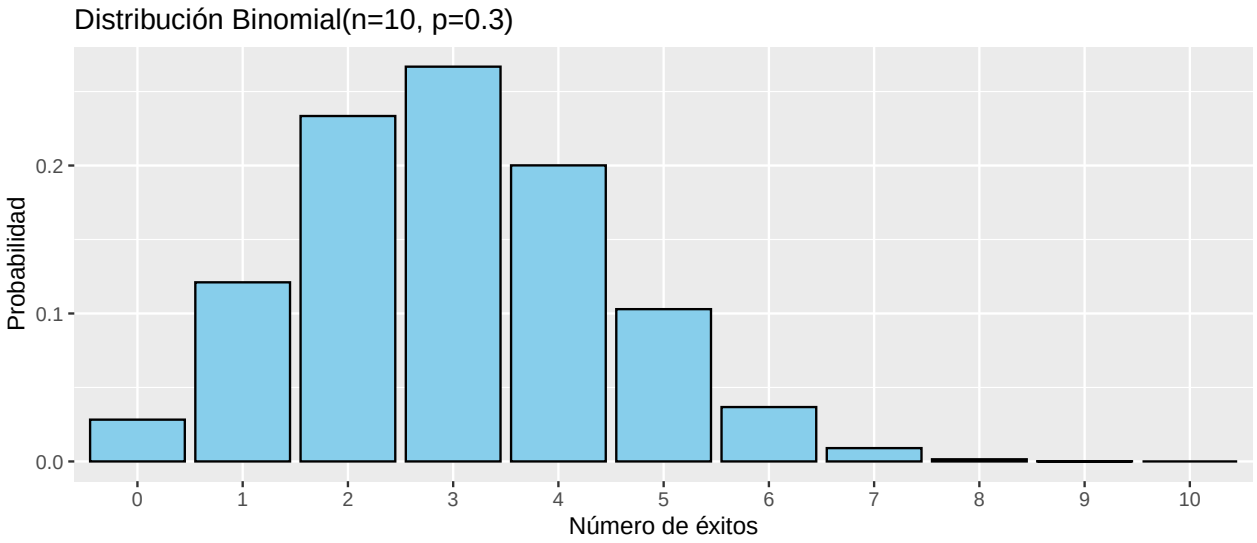
2.3. Representación de distribuciones discretas

A diferencia de las distribuciones continuas, las discretas toman valores enteros y sus probabilidades se representan mejor mediante gráficos de barras o puntos que muestren la probabilidad exacta de cada posible valor.

Ejemplo 1: Distribución Binomial

Supongamos que $X \sim \text{Binomial}(n = 10, p = 0.3)$.

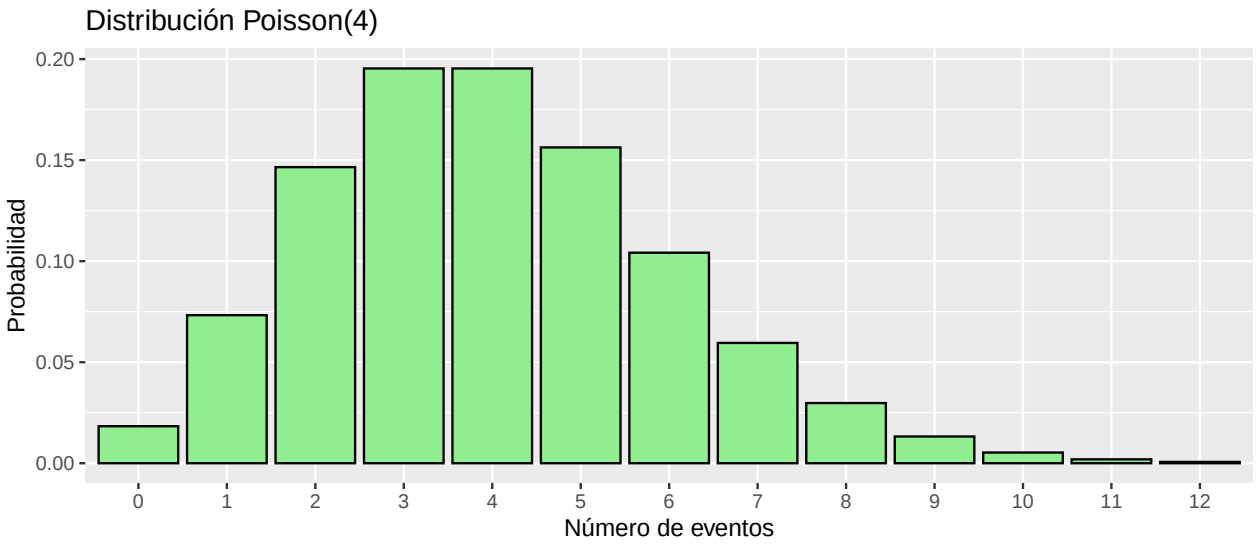
```
library(ggplot2)

n <- 10; p <- 0.3
x_vals <- 0:n
df_binom <- data.frame(
  x = x_vals,
  P = dbinom(x_vals, size = n, prob = p)
)

ggplot(df_binom, aes(x = factor(x), y = P)) +
  geom_col(fill = "skyblue", color = "black") +
  labs(title = "Distribución Binomial(n=10, p=0.3)",
       x = "Número de éxitos", y = "Probabilidad")
```



Ejemplo 2: Distribución Poisson

Ahora supongamos que $Y \sim \text{Poisson}(\lambda = 4)$.

```r
lambda <- 4
x_vals <- 0:12
df_pois <- data.frame(
  x = x_vals,
  P = dpois(x_vals, lambda = lambda)
)

ggplot(df_pois, aes(x = factor(x), y = P)) +
  geom_col(fill = "lightgreen", color = "black") +
  labs(title = "Distribución Poisson(4)",
       x = "Número de eventos", y = "Probabilidad")
```



3. Estadística descriptiva en R

La estadística descriptiva busca resumir la información contenida en un conjunto de datos de forma clara y comprensible. Es la base para cualquier análisis posterior, ya que nos permite:

- Conocer la tendencia central (media, mediana, moda).
- Evaluar la dispersión (varianza, desviación estándar, rango intercuartílico).
- Identificar la forma de la distribución (asimetría, curtosis).
- Detectar valores atípicos y patrones generales.

En R existen múltiples formas de obtener estos resúmenes, desde funciones básicas hasta paquetes especializados que generan reportes completos. En esta sección exploraremos desde el uso de funciones elementales (mean, median, var, etc.) hasta herramientas modernas como skimr y janitor para diagnóstico rápido.

3.1. Medidas de tendencia central, dispersión y posición

Las medidas descriptivas resumen información clave de una variable aleatoria o conjunto de datos. En R, la mayoría de estas medidas pueden calcularse con funciones base, sin necesidad de paquetes adicionales.

Tendencia central

- Media aritmética: $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$

```
set.seed(123)
x <- rnorm(100, mean = 50, sd = 10)  # muestra simulada

mean(x)   # media
```

```
## [1] 50.90406
```

- Mediana: el valor central al ordenar los datos.

```
median(x)   # mediana
```

```
## [1] 50.61756
```

- Moda: no existe función directa en R base, pero se puede calcular así:

```
# función propia para la moda
moda <- function(v) {
  uniqv <- unique(v)
  uniqv[which.max(tabulate(match(v, uniqv)))]
}
moda(x)
```

```
## [1] 44.39524
```

Disperción

- Varianza: $s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$

```r
var(x)   # varianza muestral
```

```
## [1] 83.32328
```

- Desviación estándar: $s = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2}$

```r
sd(x)    # desviación estándar
```

```
## [1] 9.128159
```

- Rango (mínimo y máximo):

```r
range(x)   # valores mínimo y máximo
```

```
## [1] 26.90831 71.87333
```

```r
diff(range(x))   # rango
```

```
## [1] 44.96502
```

- Rango Intercuartilico

```r
IQR(x)
```

```
## [1] 11.85673
```

Posición

- Cuantiles (incluye cuartiles, percentiles, etc.):

```r
quantile(x)                # cuartiles
```

```
##       0%      25%      50%      75%     100%
## 26.90831 45.06146 50.61756 56.91819 71.87333
```

```r
quantile(x, probs = 0.9)    # percentil 90
```

```
##      90%
## 62.64499
```

- Resumen de cinco números + media:

```r
summary(x)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   26.91   45.06   50.62   50.90   56.92   71.87
```

3.2 Forma de la distribución

La "forma" describe asimetría (sesgo hacia una cola) y apuntamiento/colas (curtosis). En R base no existen funciones directas, pero podemos implementarlas con momentos muestrales y cuantiles.

Asimetría (skewness)

Sea $x_1, \ldots, x_n$, media $\bar{x}$ y varianza muestral s^2.

- Skewness muestral (no corregida):

$$g_1 = \frac{\frac{1}{n}\sum (x_i - \bar{x})^3}{\left(\frac{1}{n}\sum (x_i - \bar{x})^2\right)^{3/2}}$$

- Skewness ajustada de Fisher–Pearson (corrige sesgo finito, n>2):

$$G_1 = \frac{\sqrt{n(n-1)}}{n-2}\, g_1$$

```
skewness <- function(x, na.rm = TRUE, fisher = TRUE) {
  if (na.rm) x <- x[!is.na(x)]
  n <- length(x)
  if (n < 3) stop("Se requieren al menos 3 observaciones.")
  m  <- mean(x)
  m2 <- mean( (x - m)^2 )
  m3 <- mean( (x - m)^3 )
  g1 <- m3 / (m2^(3/2))
  if (!fisher) return(g1)
  if (n <= 2) stop("Fisher-Pearson requiere n > 2.")
  G1 <- sqrt(n * (n - 1)) / (n - 2) * g1
  G1
}
```

Curtosis (kurtosis)

- Exceso de curtosis (0 en normal):

$$g_2 = \frac{\frac{1}{n}\sum (x_i - \bar{x})^4}{\left(\frac{1}{n}\sum (x_i - \bar{x})^2\right)^2} - 3$$

- Curtosis "total" $= g_2 + 3$ (normal $= 3$).

```
kurtosis <- function(x, na.rm = TRUE, excess = TRUE) {
  if (na.rm) x <- x[!is.na(x)]
  n <- length(x)
  if (n < 4) stop("Se requieren al menos 4 observaciones.")
  m  <- mean(x)
  m2 <- mean( (x - m)^2 )
  m4 <- mean( (x - m)^4 )
  g2 <- m4 / (m2^2) - 3
  if (excess) return(g2) else return(g2 + 3)
}
```

3.3 Tablas de frecuencia y proporciones

Las tablas de frecuencias permiten resumir cuántas observaciones caen en cada categoría/clase. En R base las herramientas clave son: table(), prop.table(), addmargins(), cut(), cumsum(), round().

Categoricas (Univariado):

```r
set.seed(123)
sexo <- sample(c("Mujer", "Hombre"), size = 200, replace = TRUE, prob = c(0.55, 0.45))

# Frecuencia absoluta
tab_sexo <- table(sexo)
tab_sexo
```

```
## sexo
## Hombre  Mujer
##     89    111
```

```r
# Frecuencia relativa (proporciones)
prop_sexo <- prop.table(tab_sexo)
round(prop_sexo, 3)
```

```
## sexo
## Hombre  Mujer
##  0.445  0.555
```

```r
# Frecuencia (y proporción) con totales de margen
addmargins(tab_sexo)                # totales
```

```
## sexo
## Hombre  Mujer    Sum
##     89    111    200
```

```r
round(addmargins(prop_sexo), 3)      # proporciones + total (=1)
```

```
## sexo
## Hombre  Mujer    Sum
##  0.445  0.555  1.000
```

Tip: si el vector es character, puedes convertir a factor y controlar el orden de los niveles:

sexo <- factor(sexo, levels = c("Mujer","Hombre")).

Numéricas (discretización en clases)

Para variables continuas, primero cortamos en intervalos con cut().

```r
set.seed(123)
ingreso <- rlnorm(300, meanlog = 10, sdlog = 0.5)  # positiva, sesgada

# Definir cortes (cuantiles o secuencia fija)
br <- quantile(ingreso, probs = seq(0, 1, by = 0.2), names = FALSE)  # quintiles
clase <- cut(ingreso, breaks = br, include.lowest = TRUE, right = TRUE)

# Frecuencias por clase
tab_ing <- table(clase)
tab_ing
```

```
## clase
## [6.94e+03,1.54e+04] (1.54e+04,1.92e+04]  (1.92e+04,2.4e+04]  (2.4e+04,3.33e+04]
##                  60                  60                  60                  60
## (3.33e+04,1.11e+05]
##                  60
```

```r
# Proporciones y porcentajes
prop_ing <- prop.table(tab_ing)
cbind(Frecuencia = as.integer(tab_ing),
      Proporcion = round(prop_ing, 3),
      Porcentaje = sprintf("%.1f%%", 100 * prop_ing))
```

```
##                     Frecuencia Proporcion Porcentaje
## [6.94e+03,1.54e+04] "60"       "0.2"      "20.0%"
## (1.54e+04,1.92e+04] "60"       "0.2"      "20.0%"
## (1.92e+04,2.4e+04]  "60"       "0.2"      "20.0%"
## (2.4e+04,3.33e+04]  "60"       "0.2"      "20.0%"
## (3.33e+04,1.11e+05] "60"       "0.2"      "20.0%"
```

Frecuencias acumuladas (numéricas)

```r
freq_abs  <- as.integer(tab_ing)
freq_rel  <- prop_ing
freq_acum <- cumsum(freq_abs)
prop_acum <- cumsum(freq_rel)

cbind(Clase = levels(clase),
      Frec = freq_abs,
      FrecAcum = freq_acum,
      Prop = round(freq_rel, 3),
      PropAcum = round(prop_acum, 3))
```

```
##                     Clase                 Frec FrecAcum Prop  PropAcum
## [6.94e+03,1.54e+04] "[6.94e+03,1.54e+04]" "60" "60"     "0.2" "0.2"
## (1.54e+04,1.92e+04] "(1.54e+04,1.92e+04]" "60" "120"    "0.2" "0.4"
## (1.92e+04,2.4e+04]  "(1.92e+04,2.4e+04]"  "60" "180"    "0.2" "0.6"
## (2.4e+04,3.33e+04]  "(2.4e+04,3.33e+04]"  "60" "240"    "0.2" "0.8"
## (3.33e+04,1.11e+05] "(3.33e+04,1.11e+05]" "60" "300"    "0.2" "1"
```

Tablas conjuntas (bivariadas)

```r
set.seed(123)
educ <- sample(c("Básica","Media","Superior"), 200, replace = TRUE, prob = c(0.3, 0.4, 0.3))
empleo <- sample(c("Formal","Informal"), 200, replace = TRUE, prob = c(0.65, 0.35))

# Tabla de contingencia
tab_bi <- table(Educacion = educ, Empleo = empleo)
tab_bi
```

```
##           Empleo
## Educacion  Formal Informal
##   Básica       43       12
##   Media        47       30
##   Superior     42       26
```

```r
# Totales de margen (fila/columna)
addmargins(tab_bi)
```

```
##           Empleo
## Educacion  Formal Informal Sum
##   Básica       43       12  55
##   Media        47       30  77
##   Superior     42       26  68
##   Sum         132       68 200
```

```
# Proporciones globales (sobre N total)
round(prop.table(tab_bi), 3)
```

```
##           Empleo
## Educacion  Formal Informal
##   Básica    0.215    0.060
##   Media     0.235    0.150
##   Superior  0.210    0.130
```

```
# Proporciones por fila (condicionales en Educación)
round(prop.table(tab_bi, margin = 1), 3)
```

```
##           Empleo
## Educacion  Formal Informal
##   Básica    0.782    0.218
##   Media     0.610    0.390
##   Superior  0.618    0.382
```

```
# Proporciones por columna (condicionales en Empleo)
round(prop.table(tab_bi, margin = 2), 3)
```

```
##           Empleo
## Educacion  Formal Informal
##   Básica    0.326    0.176
##   Media     0.356    0.441
##   Superior  0.318    0.382
```

Tablas con 3 o más variables (multivía)

```
set.seed(123)
region <- sample(c("Norte","Centro","Sur"), 200, replace = TRUE, prob = c(0.3,0.5,0.2))

tab_tri <- table(Educacion = educ, Empleo = empleo, Region = region)
tab_tri
```

```
## , , Region = Centro
##
##           Empleo
## Educacion  Formal Informal
##   Básica        0        0
##   Media        47       30
##   Superior     18        8
##
## , , Region = Norte
##
##           Empleo
## Educacion  Formal Informal
```

```
##   Básica       14        4
##   Media         0        0
##   Superior     24       18
##
## , , Region = Sur
##
##           Empleo
## Educacion  Formal Informal
##   Básica       29        8
##   Media         0        0
##   Superior      0        0
```

```r
# Sumar por un margen (p. ej., colapsar Región)
tab_bi_por_region <- addmargins(tab_tri, margin = 3)  # agrega totales por región
tab_bi_por_region[, , "Sum"]                          # tabla Educación x Empleo totalizada
```

```
##           Empleo
## Educacion  Formal Informal
##   Básica       43       12
##   Media        47       30
##   Superior     42       26
```

```r
# Proporciones condicionadas por Región (margin=3)
round(prop.table(tab_tri, margin = 3), 3)  # para cada región, proporciones de Educ x Empleo
```

```
## , , Region = Centro
##
##           Empleo
## Educacion  Formal Informal
##   Básica    0.000    0.000
##   Media     0.456    0.291
##   Superior  0.175    0.078
##
## , , Region = Norte
##
##           Empleo
## Educacion  Formal Informal
##   Básica    0.233    0.067
##   Media     0.000    0.000
##   Superior  0.400    0.300
##
## , , Region = Sur
##
##           Empleo
## Educacion  Formal Informal
##   Básica    0.784    0.216
##   Media     0.000    0.000
##   Superior  0.000    0.000
```

Manejo de NA y ordenación

```r
x_cat <- c("A","B","A", NA, "C", "B", "A", NA)

# Incluir NA como categoría explícita
tab_na <- table(x_cat, useNA = "ifany")
tab_na
```

```
## x_cat
##    A    B    C <NA>
##    3    2    1    2
```

```
prop.table(tab_na)
```

```
## x_cat
##     A     B     C  <NA>
## 0.375 0.250 0.125 0.250
```

```
# Ordenar la tabla por frecuencia descendente
tab_ord <- sort(tab_na, decreasing = TRUE)
tab_ord
```

```
## x_cat
##    A    B <NA>    C
##    3    2    2    1
```

Nota sobre pesos (frecuencias ponderadas)

Si tus observaciones traen pesos w_i, usa tapply() para sumar pesos por categoría:

```
cat <- sample(c("A","B","C"), 20, replace = TRUE)
w   <- runif(20, 0.5, 2)  # pesos

# Frecuencia ponderada (suma de pesos por categoría)
freq_w <- tapply(w, INDEX = cat, FUN = sum)
freq_w
```

```
##        A        B        C
## 9.428589 8.934537 6.701228
```

```
round(freq_w / sum(w), 3)  # proporciones ponderadas
```

```
##     A     B     C
## 0.376 0.356 0.267
```

4. Sumas aleatorias

El estudio de sumas de variables aleatorias es fundamental en probabilidad y estadística. Muchas leyes y teoremas centrales (como la Ley de los Grandes Números y el Teorema del Límite Central) se basan en analizar cómo se comporta la suma de variables aleatorias independientes.

Sea un conjunto de variables aleatorias independientes $X_1, X_2, \ldots, X_n$. Definimos la suma:

$$S_n = \sum_{i=1}^{n} X_i$$

La distribución de S_n depende de la distribución de cada X_i. Algunos casos notables:

- Si $X_i \sim \text{Bernoulli}(p)$, entonces $S_n \sim \text{Binomial}(n, p)$.
- Si $X_i \sim \text{Poisson}(\lambda)$, entonces $S_n \sim \text{Poisson}(n\lambda)$.
- Si $X_i \sim \text{N}(\mu, \sigma^2)$, entonces $S_n \sim \text{N}(n\mu, n\sigma^2)$.

4.1. Distribución de sumas de variables independientes

Cuando las variables son independientes e idénticamente distribuidas (i.i.d.), la distribución de la suma puede obtenerse de dos formas principales:

1. Por convolución:

Si X e Y son independientes, la densidad de $Z = X + Y$ es:

$$f_Z(z) = \int f_X(x) f_Y(z - x)\, dx$$

Para variables discretas, la suma se da con convoluciones de las funciones de probabilidad.

2. Por funciones generadoras de momentos (mgf):

La fgm de la suma es el producto de las fgm individuales:

$$M_{S_n}(t) = \big(M_X(t)\big)^n$$

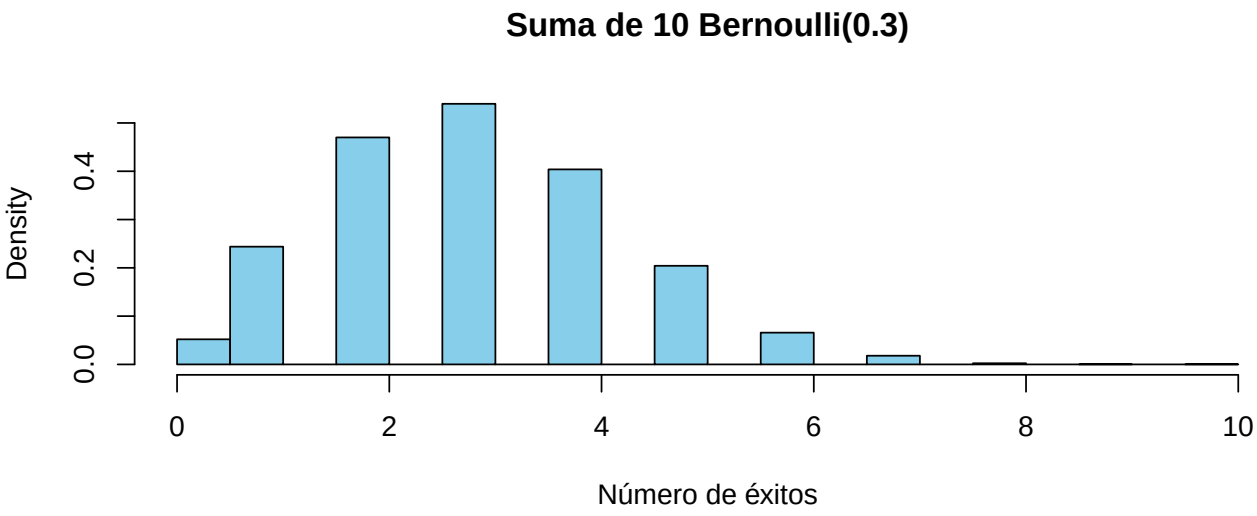
Esto permite identificar la distribución de la suma en casos clásicos (Poisson, Normal, Binomial).

Ejemplo: Suma de Bernoulli → Binomial

Si $X_i \sim \text{Bernoulli}(0.3)$, entonces:

$$S_{10} = \sum_{i=1}^{10} X_i \sim \text{Binomial}(10, 0.3)$$

```r
set.seed(123)
S <- replicate(10000, sum(rbinom(10, size = 1, prob = 0.3)))
hist(S, probability = TRUE, breaks = 15, col = "skyblue",
     main = "Suma de 10 Bernoulli(0.3)",
     xlab = "Número de éxitos")
```



4.2. Ejemplos de aplicación

Ejemplo 1:

Teoría (fgm): Sea $X_i \sim \text{Poisson}(\lambda)$. Su función generadora de momentos es:

$$M_X(t) = \exp\left(\lambda(e^t - 1)\right)$$

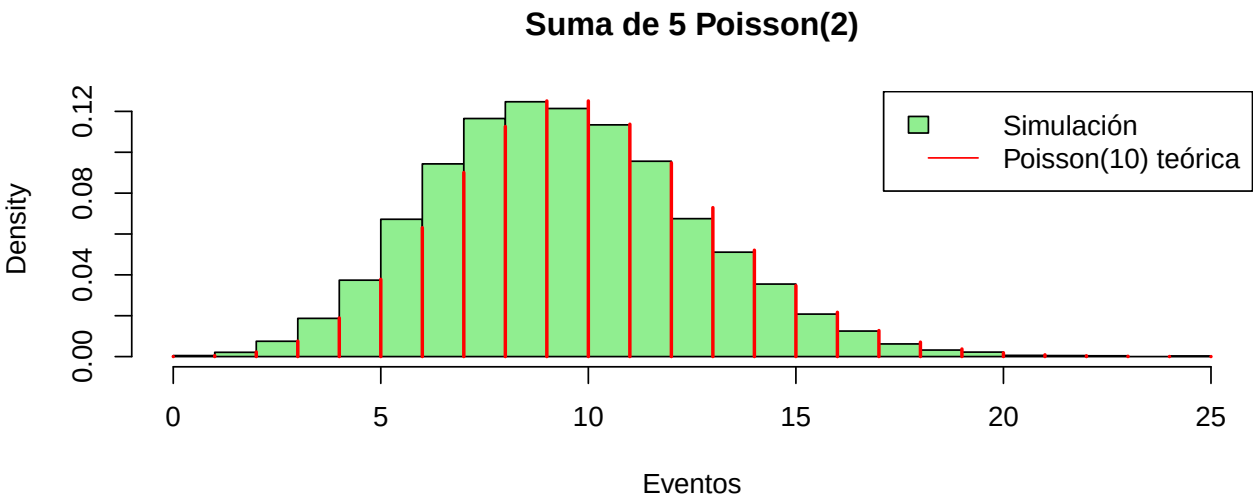
Para la suma de n variables independientes:

$$M_{S_n}(t) = \left(M_X(t)\right)^n = \left[\exp\left(\lambda(e^t - 1)\right)\right]^n = \exp\left(n\lambda(e^t - 1)\right)$$

Pero esa es justamente la mgf de una Poisson con parámetro $n\lambda$.

$$S_n \sim \text{Poisson}(n\lambda)$$

```r
set.seed(123)
# Suma de 5 Poisson(2)
S_pois <- replicate(10000, sum(rpois(5, lambda = 2)))

hist(S_pois, probability = TRUE, col = "lightgreen", breaks = 20,
     main = "Suma de 5 Poisson(2)", xlab = "Eventos")

# Superposición de la distribución teórica
x_vals <- 0:max(S_pois)
lines(x_vals, dpois(x_vals, lambda = 5*2), type = "h", lwd = 2, col = "red")
legend("topright", legend = c("Simulación", "Poisson(10) teórica"),
       fill = c("lightgreen", NA), border = c("black", NA),
       lty = c(NA, 1), col = c("black", "red"))
```



Ejemplo 2:

Teoría (fgm): Sea $X_i \sim \mathrm{N}(\mu, \sigma^2)$. Su fgm es:

$$M_X(t) = \exp\left(\mu t + \tfrac{1}{2}\sigma^2 t^2\right)$$

Para la suma de n variables:

$$M_{S_n}(t) = \left(M_X(t)\right)^n = \exp\left(n\mu t + \tfrac{1}{2}n\sigma^2 t^2\right)$$

Esto corresponde exactamente a la fgm de una normal con parámetros:

$$S_n \sim \mathrm{N}(n\mu,\, n\sigma^2)$$

Ejemplo en R:

```r
set.seed(123)
# Suma de 5 N(10, 2^2)
S_norm <- replicate(10000, sum(rnorm(5, mean = 10, sd = 2)))

hist(S_norm, probability = TRUE, breaks = 30, col = "skyblue",
     main = "Suma de 5 N(10, 2^2)", xlab = "Valor")

# Curva de densidad teórica N(5*10, 5*2^2)
curve(dnorm(x, mean = 50, sd = sqrt(5*4)), add = TRUE, col = "red", lwd = 2)
legend("topright", legend = c("Simulación", "Teórica N(50,20)"),
       fill = c("skyblue", NA), border = c("black", NA),
       lty = c(NA, 1), col = c("black", "red"))
```