Laboratorio de Probabilidad y Variables Aleatorias

Carmen Lancho - Isaac Martín - Víctor Aceña

Grado en Ciencia e Ingeniería de Datos - Inferencia Estadística - Curso 2024/2025

Índice

Objetivo

El objetivo principal de este documento es aprender a calcular probabilidades de distintas variables aleatorias. Como objetivos secundarios tenemos:

1. Identificar modelos de distribución de probabilidad a partir de unos datos
2. Saber manejar las funciones de R para calcular probabilidades
3. Saber interpretar los cálculos

1. Cálculos en variables aleatorias genéricas

La teoría de variables aleatorias ofrece un marco robusto para entender y modelizar fenómenos aleatorios en diversas disciplinas. En esta sección, abordaremos cómo realizar cálculos prácticos con variables aleatorias, tanto discretas como continuas, utilizando herramientas de R. Comenzaremos con las variables aleatorias discretas, donde la probabilidad de cada posible resultado se puede calcular directamente o acumularse. Luego, avanzaremos hacia las variables aleatorias continuas, cuya comprensión requiere integrar su función de densidad sobre un intervalo de interés.

1.1. Variables aleatorias discretas

Si tenemos la función de masa de probabilidad de una variable aleatoria discreta en forma de tabla, podemos guardar los valores x_i en un vector y los valores p_i en otro vector, y a partir de ahí realizar operaciones vectoriales para multiplicar y/o sumar, de forma que se obtengan los valores pedidos.

> Supongamos que tenemos la variable aleatoria "número de caras en el lanzamiento de 3 monedas". Los posibles valores son $\{0, 1, 2, 3\}$ y sus probabilidades $\{\frac{1}{8}, \frac{3}{8}, \frac{3}{8}, \frac{1}{8}\}$. Queremos evaluar las siguientes cuestiones:
>
> 1. **Verificación de la Suma de Probabilidades:** Es fundamental asegurar que las probabilidades sumen 1 para confirmar que están bien definidas.
> 2. **Probabilidad de un Evento:** Analizar la probabilidad de obtener menos de 2 caras nos permite entender mejor la distribución de la variable.
> 3. **Media de la Variable Aleatoria:** La media nos proporciona una medida central de la distribución.
> 4. **Varianza de la Variable Aleatoria:** La varianza nos da una idea de la dispersión de los valores alrededor de la media.

Para analizar esta variable aleatoria, definimos matemáticamente los posibles valores que puede tomar la variable $X = x_i$ y las correspondientes probabilidades $P(X = x_i) = p_i$, donde $i \in \{1, 2, 3, 4\}$. En R, almacenamos los posibles valores x_i en el vector `x` y las probabilidades asociadas p_i en el vector `p`, como se muestra a continuación:

$$X = \{x_i\} = \{0, 1, 2, 3\}$$

$$P = \{p_i\} = \left\{\frac{1}{8}, \frac{3}{8}, \frac{3}{8}, \frac{1}{8}\right\}$$

Estos vectores se utilizan luego para realizar cálculos estadísticos sobre la variable aleatoria:

```r
# Definición de los vectores de valores y probabilidades
x <- c(0, 1, 2, 3)
p <- c(1/8, 3/8, 3/8, 1/8)
```

Para confirmar que las probabilidades están correctamente normalizadas, verificamos que su suma sea igual a 1, lo cual se representa matemáticamente como:

$$\sum_{i=1}^{n} p_i = 1$$

Implementamos esta verificación en R de la siguiente manera:

```r
# Suma de probabilidades
suma_probabilidades <- sum(p)
cat(sprintf("La suma de las probabilidades es %.2f.\n", suma_probabilidades))
```

```
## La suma de las probabilidades es 1.00.
```

Calculamos la probabilidad de obtener menos de 2 caras. Matemáticamente, esta probabilidad se calcula sumando las probabilidades de los eventos donde el número de caras es menor que 2:

$$P(X < 2) = \sum_{x_i < 2} p_i = \sum_{i=1}^{2} p_i$$

Implementamos esta probabilidad en R de la siguiente manera:

```
# Probabilidad de obtener menos de 2 caras
probabilidad_menos_2 <- sum(p[x < 2])
cat(sprintf("La probabilidad de obtener menos de 2 caras es %.2f.\n", probabilidad_menos_2))
```

```
## La probabilidad de obtener menos de 2 caras es 0.50.
```

La media o esperanza matemática se calcula utilizando la expresión:

$$E(X) = \sum_{i=1}^{n} x_i p_i$$

Para determinar la media de la variable aleatoria:

```
# Media de la variable aleatoria
media <- sum(x * p)
cat(sprintf("La media de la variable aleatoria es %.2f.\n", media))
```

```
## La media de la variable aleatoria es 1.50.
```

La varianza se calcula con la expresión:

$$Var(X) = E(X^2) - [E(X)]^2 = \sum_{i=1}^{n} (x_i^2 p_i) - \left(\sum_{i=1}^{n} x_i p_i \right)^2$$

En R se puede implementar como:

```
# Varianza de la variable aleatoria
varianza <- sum(x^2 * p) - media^2
cat(sprintf("La varianza de la variable aleatoria es %.2f.\n", varianza))
```

```
## La varianza de la variable aleatoria es 0.75.
```

1.2. Variables aleatorias continuas

Los cálculos de variables aleatorias continuas se realizan a través de la función de densidad (con integrales) o a través de la función de distribución (sustituyendo valores en la función). En ambos casos, lo más cómodo es crear una función en R para esas funciones matemáticas.

Supongamos que tenemos una variable aleatoria definida por sus funciones de densidad y de distribución, definidas de la siguiente manera:

$$f(x) = \begin{cases} \frac{1}{8}x & \text{si} \quad 0 < x < 4 \\ \\ 0 & \text{resto} \end{cases} \quad ; \quad F(x) = \begin{cases} 0 & \text{si} \quad x \leq 0 \\ \frac{x^2}{16} & \text{si} \quad 0 < x < 4 \\ 1 & \text{si} \quad x \geq 4 \end{cases}$$

Sobre esta variable aleatoria se plantean las siguientes cuestiones:

1. **Verificación de la integral de la función de densidad:** Es fundamental asegurar que la función $f(x)$ es una función de densidad, para ello se verifica que la integral total es 1.
2. **Probabilidad de un Evento:** La probabilidad de que la variable aleatoria sea menor de 3.
3. **Probabilidad de un Evento:** La probabilidad de que la variable aleatoria esté entre 1.5 y 2.
4. **Media de la Variable Aleatoria:** La media nos proporciona una medida central de la distribución.
5. **Varianza de la Variable Aleatoria:** La varianza nos da una idea de la dispersión de los valores alrededor de la media.

A continuación, vamos a implementar estas funciones en R. La función `f` representa la función de densidad, mientras que `Fdist` representa la función de distribución.

```
f <- function(x) (1/8)*x
Fdist <- function(x) x^2/16
```

La sencillez de las expresiones nos sirven para esta práctica, poniendo cuidado en elegir los valores de `x` entre 0 y 4.

Si quisiéramos algo más elaborado podemos introducir condiciones, por ejemplo así:

```
Fdist2 <- function(x){
  if(x <= 0){
    0
  } else if(x < 4){
    x^2/16
  }
  else{
    1
  }
}
```

Las funciones creadas se pueden utilizar ahora para obtener valores sustituyendo la `x` por cualquier valor. Por ejemplo, para obtener valores de la función de distribución, que son probabilidades directamente, $F(1)$ sería `Fdist(1)`. Para obtener probabilidades con la función de densidad, utilizamos la función `integrate`, introduciendo como primer argumento la función, y después los límites de la integral. Admite el valor `inf`.

Para verificar que $f(x)$ es una función de densidad se evalúa que su integral total es 1:

$$\int_{-\infty}^{\infty} f(x)\,dx = \int_{0}^{4} \frac{1}{8}x\,dx = 1$$

En R, realizamos el cálculo mediante función `integrate` de R para evaluar la integral:

```r
integral_total <- integrate(f, 0, 4)

cat(sprintf("La integral de f(x) es %.2f.\n", integral_total$value))
```

```
## La integral de f(x) es 1.00.
```

> Fíjate que la salida de la integral incluye el error cometido. Esto es porque el ordenador utiliza métodos numéricos (no exactos) para obtener la integral. por eso después para usar el valor lo extraemos con `$value`. Por otra parte, podemos meter como argumento de la función `integrate` funciones creadas "al vuelo".

Para calcular la probabilidad de que la variable aleatoria sea menor que 3, es decir $P\{X \leq 3\}$, evaluamos $F(x)$ en $x = 3$:

```r
cat(sprintf("La probabilidad de que X sea menor o igual que 3 es %.2f.\n", Fdist(3)))
```

```
## La probabilidad de que X sea menor o igual que 3 es 0.56.
```

Para calcular la probabilidad de que la variable aleatoria esté entre 1.5 y 2, necesitamos integrar la función de densidad en ese intervalo. Matemáticamente, esto se expresa como:

$$P(1{,}5 \leq X \leq 2) = \int_{1{,}5}^{2} f(x)\,dx$$

En R, realizamos el cálculo mediante función `integrate` de R para evaluar la integral entre 1.5 y 2.

```r
cat(sprintf("La probabilidad de que X esté entre 1.5 y 2 es %.2f.\n", integrate(f, 1.5, 2)$value))
```

```
## La probabilidad de que X esté entre 1.5 y 2 es 0.11.
```

Podemos comprobar que este resultado es equivalente si empleamos la función de distribución:

$$P(1{,}5 \leq X \leq 2) = F(2) - F(1{,}5)$$

Donde $F(2)$ es la probabilidad de que la variable aleatoria sea menor o igual que 2, y $F(1{,}5)$ es la probabilidad de que la variable aleatoria sea menor o igual que 1.5. Restar estas dos probabilidades nos da la probabilidad de que la variable aleatoria esté entre 1.5 y 2.

```r
diferencia <- Fdist(2) - Fdist(1.5)

cat(sprintf("La probabilidad de que X esté entre 1.5 y 2 es %.2f.\n", diferencia))
```

```
## La probabilidad de que X esté entre 1.5 y 2 es 0.11.
```

En el contexto de la variable aleatoria definida por la función de densidad $f(x)$, la esperanza se calcula como:

$$E(X) = \int_{-\infty}^{\infty} x \cdot f(x)\,dx$$

Podemos calcular la esperanza de la variable aleatoria utilizando la función `integrate` de R para evaluar la integral $\int_0^4 x \cdot f(x)\,dx$:

```r
Ex <- integrate(function(x) x*f(x), 0, 4)$value

cat(sprintf("La media de la variable aleatoria es %.2f.\n", Ex))
```

```
## La media de la variable aleatoria es 2.67.
```

En el contexto de la variable aleatoria definida por la función de densidad $f(x)$, la varianza se calcula como:

$$Var(X) = E(X^2) - [E(X)]^2$$

Aplicando este concepto, primero calculamos $E(X^2)$ utilizando la función de densidad y luego calculamos la varianza utilizando la expresión anterior:

```r
Ex2 <- integrate(function(x) x^2*f(x), 0, 4)$value
Vx <- Ex2-Ex^2

cat(sprintf("La varianza de la variable aleatoria es %.2f.\n", Vx))
```

```
## La varianza de la variable aleatoria es 0.89.
```

2. Cálculos en modelos de distribución de probabilidad

En R, para cada modelo de distribución de probabilidad tenemos una función que empieza por `d` y devuelve la "densidad" (masa de probabilidad en el caso de discretas) y otra que empieza por `p` y devuelve la "probabilidad (acumulada)", es decir, la función de distribución (o su complementario, añadiendo el argumento `lower.tail = FALSE`). Después de la `d` o la `p` vendrá el nombre (o abreviatura) del modelo de probabilidad, por ejemplo para la distribución normal `norm`.

Para cada modelo de distribución de probabilidad tenemos otras dos funciones, una que empieza por `q`, que calcula el cuantil dada una probabilidad acumulada (es decir, es la función inversa de la función de distribución) y otra que empieza por `r`, con la que podemos obtener valores aleatorios (*random*) o simulaciones de una variable aleatoria.

2.1. Distribuciones discretas

Las distribuciones discretas juegan un papel crucial cuando tratamos con variables aleatorias que toman valores específicos o cuentas. En contextos donde los resultados son contables y no continuos, aplicamos modelos de distribución discreta para describir la probabilidad asociada a cada posible valor de la variable. En R, utilizamos funciones que comienzan con `d` para calcular la masa de probabilidad, que nos da la probabilidad de cada valor específico de la variable. Similarmente, las funciones que comienzan con `p` nos ayudan a obtener la función de distribución acumulada para evaluar la probabilidad de que la variable tome un valor menor o igual a un cierto límite.

En esta sección, abordaremos modelos específicos como la distribución binomial, la distribución geométrica, la distribución binomial negativa, la distribución hipergeométrica y la distribución de Poisson. A través de ejemplos prácticos, exploraremos cómo aplicar las funciones `dxxx`, `pxxx`, `qxxx`, y `rxxx` para comprender y analizar estas distribuciones en diversos contextos.

2.1.1. Funciones disponibles

En R, cada distribución discreta cuenta con cuatro funciones asociadas, que permiten realizar distintos cálculos estadísticos:

- `dxxx(x, ...)` calcula la función de masa de probabilidad, $f(x)$, para un valor dado de x.
- `pxxx(q, ...)` obtiene la función de distribución acumulada, $F(x)$, hasta un punto q.
- `qxxx(p, ...)` determina el cuantil para el cual la probabilidad $P(X \leq q)$ es igual a p.
- `rxxx(n, ...)` genera n números aleatorios siguiendo la distribución especificada.

Reemplace `xxx` por el identificador correspondiente de la distribución que desea utilizar. A continuación, se presentan los identificadores para las seis distribuciones discretas básicas en R:

- `binom`: Distribución Binomial.
- `geo`: Distribución Geométrica.
- `nbinom`: Distribución Binomial Negativa.
- `hyper`: Distribución Hipergeométrica.
- `pois`: Distribución de Poisson.
- `multinom`: Distribución Multinomial (Nota: `multinom` no sigue la misma nomenclatura de `dxxx`, `pxxx`, `qxxx`, `rxxx` ya que se utiliza principalmente para modelos multinomiales).

Las funciones en R están meticulosamente diseñadas para facilitar el análisis y la modelización de variables aleatorias. La consistencia en la nomenclatura de estas funciones ayuda a identificar rápidamente la herramienta adecuada para cada tipo de cálculo relacionado con distintas distribuciones estadísticas.

2.1.2. Distribución Binomial

La distribución binomial se utiliza para modelar el número de éxitos en una secuencia de n ensayos independientes entre sí, con una probabilidad fija p de ocurrencia del éxito en cada ensayo. La función `dbinom(x, size, prob)` en R proporciona la probabilidad de obtener exactamente x éxitos en n ensayos.

Para utilizar `dbinom`, necesitamos especificar:

- x: el número de éxitos que estamos investigando.
- `size` (n): el número total de ensayos.
- `prob` (p): la probabilidad de éxito en cada ensayo.

La probabilidad de obtener exactamente x éxitos se calcula como:

$$P(X = x) = \binom{n}{x} p^x (1-p)^{n-x}$$

> Supongamos que la probabilidad de que un estudiante acabe un grado en Ciencias es de 0,4. Tomamos al azar un grupo de 5 estudiantes. ¿Cuál es la probabilidad de que ninguno obtenga el grado? ¿Y la probabilidad de que al menos dos lo obtengan?

Para calcular la probabilidad de un número específico de éxitos en una distribución binomial, utilizamos la función `dbinom` en R. Esta función devuelve la probabilidad de obtener un número determinado de éxitos (`x`) en un número fijo de ensayos independientes (`size`), cada uno con la misma probabilidad de éxito (`prob`).

En el contexto de nuestro ejemplo, si queremos calcular la probabilidad de que ninguno de los 5 estudiantes termine su grado en Ciencias (considerando que la probabilidad de que un estudiante termine es de 0.4),

podemos configurar `dbinom` con los siguientes parámetros: `x = 0` (ningún estudiante termina), `size = 5` (cinco ensayos o estudiantes) y `prob = 0.4` (la probabilidad de éxito, es decir, que un estudiante termine).

La fórmula matemática que `dbinom` utiliza para calcular esta probabilidad es:

$$P(X = x) = \binom{n}{x} p^x (1-p)^{n-x}$$

Donde:

- X es la variable aleatoria que representa el número de éxitos (estudiantes que terminan el grado).
- $n = 5$ es el número total de ensayos (estudiantes).
- $x = 0$ es el número de éxitos para los que queremos calcular la probabilidad.
- $p = 0{,}4$ es la probabilidad de éxito en cada ensayo.

Al aplicar estos valores en la fórmula, calculamos la probabilidad específica para nuestro escenario:

```
prob_no_grado <- dbinom(x = 0, size = 5, prob = 0.4)
cat(sprintf("La probabilidad de que ninguno obtenga el grado es%.4f.\n", prob_no_grado))
```

```
## La probabilidad de que ninguno obtenga el grado es 0.0778.
```

Para calcular la probabilidad de que el número de éxitos sea mayor a un valor específico en una distribución binomial, podemos utilizar la función `pbinom` de R. Esta función devuelve la probabilidad acumulada de obtener un número de éxitos menor o igual a `q` en una serie de ensayos.

En nuestro ejemplo, si queremos saber la probabilidad de que más de un estudiante (es decir, al menos dos) termine su grado en Ciencias, podemos calcular la complementaria de la probabilidad de que uno o ninguno lo logre. Esto se puede hacer utilizando `1 - pbinom(q = 1, size = 5, prob = 0.4)`, donde `q = 1` representa el umbral máximo de estudiantes (uno) que no queremos superar para nuestro cálculo complementario.

La función `pbinom` utiliza la siguiente fórmula para calcular la probabilidad acumulada:

$$P(X \leq q) = \sum_{k=0}^{q} \binom{n}{k} p^k (1-p)^{n-k}$$

Al restar este resultado de 1, obtenemos la probabilidad de que más de un estudiante (al menos dos) termine el grado:

$$P(X > q) = 1 - P(X \leq q)$$

Por lo tanto, el siguiente bloque de código calcula la probabilidad de que al menos dos estudiantes terminen el grado:

```
prob_dos_grado <- 1 - pbinom(q = 1, size = 5, prob = 0.4)
cat(sprintf("La probabilidad de que al menos dos estudiantes terminen el grado es%.4f.\n", prob_dos_gra
```

```
## La probabilidad de que al menos dos estudiantes terminen el grado es 0.6630.
```

Otra forma de calcular la probabilidad de que más de un estudiante termine el grado es utilizar directamente la función `pbinom` con el argumento `lower.tail = FALSE`. Esto nos permite calcular la probabilidad de que el número de éxitos sea mayor que `q` directamente, sin necesidad de calcular la complementaria.

El argumento `lower.tail = FALSE` indica que estamos interesados en la probabilidad de que el número de éxitos en nuestra distribución binomial sea mayor que el valor `q` proporcionado. En este caso, queremos la probabilidad de que más de un estudiante, es decir, dos o más, termine el grado.

Por lo tanto, el código:

```
prob_dos_grado_v2 <- pbinom(q = 1, size = 5, prob = 0.4, lower.tail = FALSE)

cat(sprintf("La probabilidad de que al menos dos estudiantes terminen el grado es%.4f.\n", prob_dos_gra
```

```
## La probabilidad de que al menos dos estudiantes terminen el grado es 0.6630.
```

realiza directamente el cálculo de $P(X > 1)$ para nuestra distribución binomial, donde X es el número de estudiantes que terminan el grado, `size = 5` es el número total de estudiantes considerados, y `prob = 0.4` es la probabilidad de que un estudiante dado termine el grado.

Finalmente, otra manera de abordar el cálculo de la probabilidad de que al menos dos estudiantes terminen su grado es sumar directamente las probabilidades de obtener 2, 3, 4 o 5 estudiantes que terminan el grado. Esto se puede hacer sumando las probabilidades individuales para cada uno de estos casos.

Utilizamos `dbinom` para calcular la probabilidad de cada número específico de éxitos (en este caso, de 2 a 5 éxitos) y luego sumamos estos valores. Esto nos proporciona la probabilidad total de que al menos dos estudiantes, entre los cinco considerados, terminen el grado en Ciencias. El parámetro `size = 5` define el número total de ensayos (estudiantes), y `prob = 0.4` es la probabilidad de éxito individual (un estudiante terminando el grado).

El código siguiente ejecuta este cálculo:

```
prob_dos_grado_v3 <- sum(dbinom(x = 2:5, size = 5, prob = 0.4))

cat(sprintf("La probabilidad de que al menos dos estudiantes terminen el grado es%.4f.\n", prob_dos_gra
```

```
## La probabilidad de que al menos dos estudiantes terminen el grado es 0.6630.
```

Esta expresión suma las probabilidades calculadas por `dbinom` para 2, 3, 4 y 5 éxitos (estudiantes que terminan el grado) en 5 ensayos, ofreciendo otra vía para determinar la probabilidad de que al menos dos estudiantes logren terminar su grado.

2.1.3. Distribución de Bernoulli

La distribución de Bernoulli es la distribución más simple, modelando un experimento que tiene solo dos posibles resultados: éxito o fracaso (normalmente codificados como 1 y 0, respectivamente). La función `dbinom(x, size, prob)` en R puede utilizarse para una variable aleatoria de Bernoulli, especificando `size = 1` (un solo ensayo).

En la distribución de Bernoulli:

- x: el valor de éxito o fracaso (0 o 1).
- `size = 1`: ya que se trata de un solo ensayo.
- `prob` (p): la probabilidad de éxito en ese ensayo.

La probabilidad de obtener exactamente x en un experimento de Bernoulli se calcula como:

$$P(X = x) = p^x(1-p)^{1-x}$$

> Supongamos que lanzamos una moneda con una probabilidad de 0.6 de obtener cara (éxito). ¿Cuál es la probabilidad de que salga cara (éxito)? ¿Cuál es la probabilidad de que salga cruz (fracaso)?

Podemos utilizar la función `dbinom` en R para calcular las probabilidades de los dos resultados posibles (0 o 1) de una variable aleatoria de Bernoulli.

```
p <- 0.6  # Probabilidad de éxito

# Probabilidades para éxito (1) y fracaso (0)
prob_1 <- dbinom(x = 1, size = 1, prob = p)
prob_0 <- dbinom(x = 0, size = 1, prob = p)

cat(sprintf("La probabilidad de éxito (cara) es %.4f.\n", prob_1))
```

```
## La probabilidad de éxito (cara) es 0.6000.
```

```
cat(sprintf("La probabilidad de fracaso (cruz) es %.4f.\n", prob_0))
```

```
## La probabilidad de fracaso (cruz) es 0.4000.
```

2.1.4. Distribución de Poisson

La distribución de Poisson se utiliza para modelar el número de eventos que ocurren en un intervalo de tiempo fijo cuando estos eventos ocurren con una tasa media constante y de manera independiente entre sí. Esta distribución se define completamente por su parámetro λ (lambda), que representa la tasa media de ocurrencia de los eventos por intervalo.

Para un valor dado k, la probabilidad de observar exactamente k eventos se calcula como:

$$P(X = k) = \frac{e^{-\lambda}\lambda^k}{k!}$$

Donde:

- X es la variable aleatoria que representa el número de eventos,
- λ es la tasa media de eventos por intervalo,
- k es el número de ocurrencias del evento.

Las funciones principales en R para trabajar con esta distribución son `dpois` para la función de probabilidad y `ppois` para la función de distribución acumulada.

> En una parada de autobús llegan de media cuatro autobuses cada hora. ¿Cuál es la probabilidad de que en una hora pasen más de 8?

En el contexto de una parada de autobús donde llegan de media cuatro autobuses cada hora, podemos usar la distribución de Poisson para modelar la probabilidad de diferentes números de llegadas. Para calcular la probabilidad de que lleguen más de 8 autobuses en una hora, podemos usar la función `ppois` con `lower.tail = FALSE`, que nos da la probabilidad de que el número de llegadas sea mayor que un valor dado (en este caso, 8 autobuses).

La función `ppois(8, lambda = 4, lower.tail = FALSE)` calculará la probabilidad de que el número de autobuses que llegan en una hora sea mayor que 8, dado que la tasa media (λ) es de 4 autobuses por hora.

```
prob_8buses <- ppois(8, lambda = 4, lower.tail = FALSE)
cat(sprintf("La probabilidad de que el número de autobuses que llegan en una hora sea mayor que 8 es %.4
```

```
## La probabilidad de que el número de autobuses que llegan en una hora sea mayor que 8 es 0.0214.
```

Otro escenario común para aplicar la distribución de Poisson es el modelado del número de errores tipográficos que ocurren en una página de texto.

> Supongamos que, en promedio, se cometen 2 errores tipográficos por página en un libro. ¿Cuál es la probabilidad de que en una página no haya errores?

En este ejemplo, queremos calcular la probabilidad de que en una página específica no haya errores tipográficos, sabiendo que la tasa media (λ) es de 2 errores por página.

La función `dpois` nos permitirá calcular la probabilidad de observar exactamente $k = 0$ errores en una página, dado que la tasa media de errores es $\lambda = 2$.

```
prob_no_errors <- dpois(0, lambda = 2)
cat(sprintf("La probabilidad de que no haya errores en una página es %.4f.\n", prob_no_errors))
```

```
## La probabilidad de que no haya errores en una página es 0.1353.
```

2.1.5. Distribución Binomial Negativa

La distribución binomial negativa modela el número de fracasos que ocurren antes de que se alcancen un número fijo de éxitos en una secuencia de ensayos independientes con probabilidad constante de éxito. La función `dnbinom(x, size, prob)` en R devuelve la probabilidad de observar un número específico de fracasos antes de alcanzar un número determinado de éxitos.

Para utilizar `dnbinom`, necesitamos especificar:

- x: el número de fracasos antes de alcanzar los éxitos deseados.
- `size` (k): el número de éxitos que estamos esperando alcanzar.
- `prob` (p): la probabilidad de éxito en cada ensayo.

La probabilidad se calcula como:

$$P(X = x) = \binom{x + k - 1}{k - 1} p^k (1 - p)^x$$

> Supongamos que lanzamos una moneda con probabilidad de 0.4 de éxito (cara). Queremos saber la probabilidad de que ocurran 3 fracasos antes de que consigamos 2 éxitos.

Donde:

- X es la variable que representa el número de fracasos.
- $k = 2$ es el número de éxitos que se desean.
- $p = 0{,}4$ es la probabilidad de éxito en cada ensayo.

```
p <- 0.4  # Probabilidad de éxito
size <- 2  # Número de éxitos deseados

# Probabilidad de tener exactamente 3 fracasos antes de 2 éxitos
prob_3_failures <- dnbinom(x = 3, size = size, prob = p)
cat(sprintf("La probabilidad de que ocurran exactamente 3 fracasos antes de 2 éxitos es%.4f.\n", prob_3
```

```
## La probabilidad de que ocurran exactamente 3 fracasos antes de 2 éxitos es 0.1382.
```

2.1.6. Distribución Geométrica

La distribución geométrica modela el número de fracasos que ocurren antes del primer éxito en una secuencia de ensayos independientes con probabilidad constante de éxito. La función `dgeom(x, prob)` en R calcula la probabilidad de obtener exactamente x fracasos antes de obtener el primer éxito.

Para utilizar `dgeom`, necesitamos especificar:

- x: el número de fracasos antes del primer éxito.
- `prob` (p): la probabilidad de éxito en cada ensayo.

La probabilidad de obtener exactamente x fracasos antes del primer éxito se calcula como:

$$P(X = x) = (1 - p)^x p$$

Donde p es la probabilidad de éxito en cada ensayo y x es el número de fracasos.

> Supongamos que lanzamos una moneda con una probabilidad de 0.7 de obtener cara (éxito). Queremos calcular la probabilidad de que haya exactamente 2 fracasos antes de obtener la primera cara.

```
p <- 0.7  # Probabilidad de éxito

# Probabilidad de obtener 2 fracasos antes del primer éxito
prob_2_failures <- dgeom(x = 2, prob = p)
cat(sprintf("La probabilidad de que ocurran exactamente 2 fracasos antes del primer éxito es%.4f.\n", p
```

```
## La probabilidad de que ocurran exactamente 2 fracasos antes del primer éxito es 0.0630.
```

Otro caso en el que se puede aplicar la distribución geométrica es en la detección de defectos en un proceso de producción. Supongamos que en una fábrica, cada artículo producido tiene una probabilidad de 0.1 de ser defectuoso. Queremos calcular la probabilidad de que haya exactamente 3 productos defectuosos antes de encontrar el primer producto no defectuoso (éxito).

> En una fábrica, cada producto tiene una probabilidad de 0.1 de ser defectuoso. Calcula la probabilidad de que haya exactamente 3 productos defectuosos antes de encontrar el primer producto no defectuoso.

En este ejemplo, podemos usar la función `dgeom` para calcular la probabilidad de tener exactamente 3 fracasos (productos defectuosos) antes de encontrar el primer éxito (producto no defectuoso).

```
p <- 0.9  # Probabilidad de éxito (producto no defectuoso)

# Probabilidad de obtener 3 productos defectuosos antes del primer no defectuoso
prob_3_failures <- dgeom(x = 3, prob = p)
cat(sprintf("La probabilidad de que haya exactamente 3 productos defectuosos antes del primer no defectu
```

```
## La probabilidad de que haya exactamente 3 productos defectuosos antes del primer no defectuoso es 0.
```

2.1.7. Distribución Hipergeométrica

Para la distribución hipergeométrica, utilizamos `dhyper` en R, que requiere parámetros específicos para describir el escenario: el total de éxitos en la población (`m`), el total de no éxitos en la población (`n`), y el número de extracciones (`k`). La función devuelve la probabilidad de obtener un número dado de éxitos (`x`) en las extracciones realizadas.

> En un comité de dirección de una empresa medioambiental con 50 miembros, 30 están de acuerdo en crear una línea de negocio de vehículo eléctrico, y el resto no. En el descanso, cinco directivos (al azar) se salen a la máquina de café. ¿Cuál es la probabilidad de que de esos cinco solo uno esté de acuerdo en crear la línea de negocio?

Los parámetros para `dhyper` en este caso son: `x = 1` (un directivo a favor entre los seleccionados), `m = 30` (total de directivos a favor en el comité), `n = 20` (total de directivos en contra), y `k = 5` (número de directivos seleccionados para ir al café).

La probabilidad se calcula como:

$$P(X = x) = \frac{\binom{m}{x}\binom{N-m}{k-x}}{\binom{N}{k}}$$

Donde:

- $N = m + n$ es el total de la población o el tamaño del comité.
- X es la variable aleatoria que representa el número de directivos a favor seleccionados.

Aplicamos la función `dhyper` para obtener la probabilidad:

```
prob_no_acuerdo <- dhyper(x = 1, m = 30, n = 20, k = 5)

cat(sprintf("La probabilidad de que de esos cinco solo uno esté de acuerdo en crear la línea de negocio
```

```
## La probabilidad de que de esos cinco solo uno esté de acuerdo en crear la línea de negocio 0.0686.
```

2.2. Distribuciones continuas

Las distribuciones continuas son esenciales cuando abordamos variables aleatorias que pueden tomar cualquier valor dentro de un intervalo. Estas distribuciones nos permiten entender y modelar fenómenos donde

la precisión y la continuidad son claves, como tiempos, distancias o temperaturas. En este caso, las funciones que empiezan por `d` nos proporcionan la densidad de probabilidad, que, a diferencia de la masa de probabilidad para las discretas, nos ofrece la densidad en un punto específico o intervalo. Las funciones que inician con `p` siguen siendo cruciales para determinar la probabilidad acumulada, permitiéndonos calcular la probabilidad de que la variable aleatoria caiga por debajo de un cierto valor.

En esta parte, nos centraremos en distribuciones continuas claves como la distribución normal, la distribución exponencial, y otras distribuciones relevantes. Exploraremos el uso de `dxxx`, `pxxx`, `qxxx`, y `rxxx` para efectuar análisis detallados y aplicaciones prácticas de estas distribuciones, proporcionando así una base sólida para el modelado y la interpretación de datos continuos.

2.2.1. Funciones disponibles

En R, cada distribución continua tiene asociadas cuatro funciones principales que facilitan la realización de cálculos estadísticos importantes:

- `dxxx(x, ...)` calcula la función de densidad de probabilidad, $f(x)$, para un valor dado de x.
- `pxxx(q, ...)` obtiene la función de distribución acumulada, $F(x)$, hasta un punto q.
- `qxxx(p, ...)` determina el cuantil para el cual la probabilidad $P(X \leq q)$ es igual a p.
- `rxxx(n, ...)` genera n números aleatorios que siguen la distribución especificada.

Reemplace `xxx` por el identificador correspondiente de la distribución que desea utilizar. A continuación, se presentan los identificadores para algunas de las distribuciones continuas comunes en R:

- `beta`: Distribución Beta.
- `cauchy`: Distribución Cauchy.
- `chisq`: Distribución Chi-cuadrado.
- `exp`: Distribución Exponencial.
- `f`: Distribución F.
- `gamma`: Distribución Gamma.
- `lnorm`: Distribución Log-normal.
- `norm`: Distribución Normal.
- `t`: Distribución t-Student.
- `unif`: Distribución Uniforme.
- `weibull`: Distribución Weibull.

Las funciones en R para distribuciones continuas permiten calcular la densidad de probabilidad (`dxxx`), la probabilidad acumulada (`pxxx`), cuantiles (`qxxx`), y generar muestras aleatorias (`rxxx`). A diferencia de las distribuciones discretas, donde `dxxx` devuelve una probabilidad puntual, en las continuas proporciona la densidad en un punto, reflejando la concentración de probabilidad, no una probabilidad exacta. Este diseño consistente facilita la identificación y aplicación correcta de cada función para análisis y modelización estadística eficaz.

Para la distribución uniforme utilizamos las funciones `dunif` y `punif`, con los argumentos `min` y `max` para los parámetros a y b respectivamente. `dunif(x, min, max)` ofrece el valor de la densidad de probabilidad en el punto `x`, mientras que `punif(q, min, max)` calcula la probabilidad de que una variable aleatoria uniforme sea menor o igual a `q`.

2.2.2. Distribución Uniforme

La distribución uniforme continua se utiliza para modelar situaciones en las que todos los valores dentro de un intervalo están igualmente distribuidos, es decir, tienen la misma probabilidad de ocurrencia. La función

`dunif(x, min, max)` en R calcula la densidad de probabilidad para un valor específico dentro de un intervalo definido por `min` y `max`.

Para utilizar `dunif`, necesitamos especificar:

- x: el valor para el cual estamos calculando la densidad de probabilidad.
- `min`: el límite inferior del intervalo.
- `max`: el límite superior del intervalo.

La función de densidad de la distribución uniforme continua se define como:

$$f(x) = \frac{1}{b-a} \quad \text{para } a \leq x \leq b$$

Donde a es el límite inferior y b es el límite superior.

> Supongamos que seleccionamos al azar un número entre 0 y 10. Queremos calcular la probabilidad de que ese número caiga exactamente en 5, y luego queremos generar y visualizar 1000 números aleatorios que sigan una distribución uniforme entre 0 y 10.

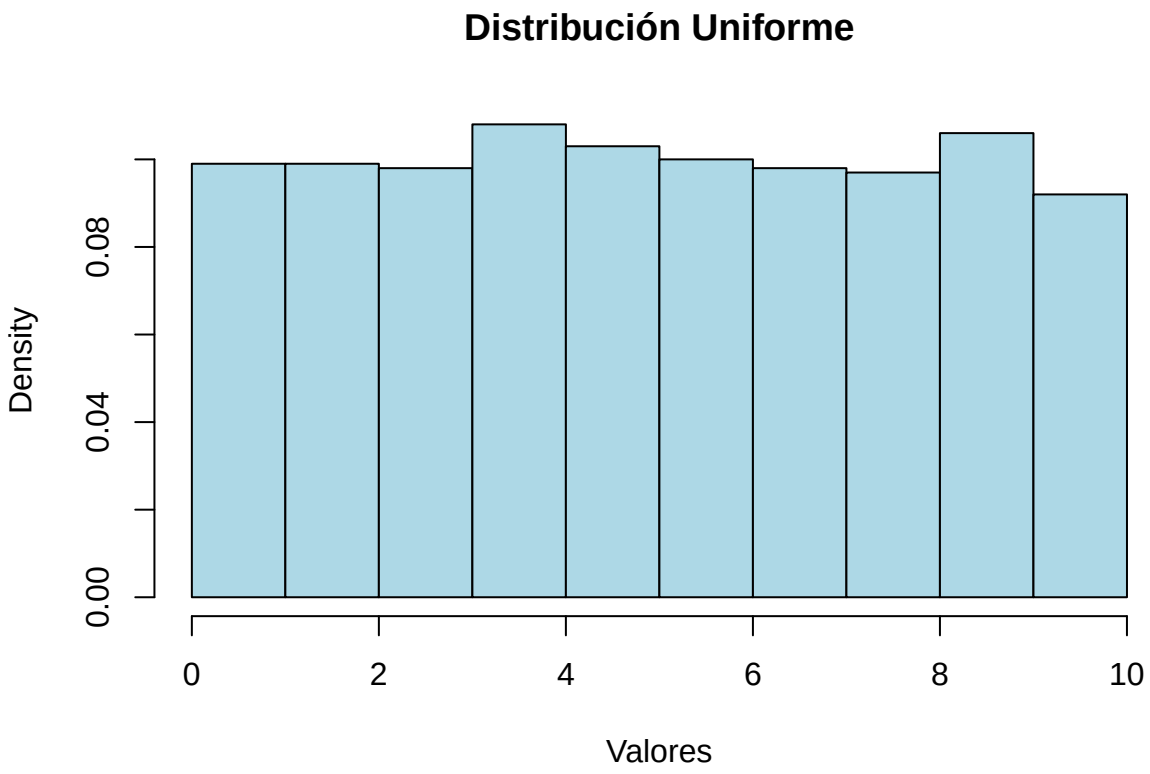
```r
# Parámetros de la distribución uniforme
min_val <- 0
max_val <- 10

# Probabilidad de que el número sea exactamente 5
# Nota: en la distribución continua, la probabilidad de un solo punto es 0, pero mostramos cómo usar la
prob_x_equals_5 <- dunif(5, min = min_val, max = max_val)
cat(sprintf("La densidad en x = 5 es%.4f.\n", prob_x_equals_5))
```

```
## La densidad en x = 5 es 0.1000.
```

```r
# Generar 1000 números aleatorios con distribución uniforme entre 0 y 10
set.seed(123)
random_uniform <- runif(1000, min = min_val, max = max_val)

# Visualizar histograma de los números aleatorios generados
hist(random_uniform, probability = TRUE, main = "Distribución Uniforme",
     xlab = "Valores", xlim = c(min_val, max_val), col = "lightblue", border = "black")
```



2.2.3. Distribución exponencial

La distribución exponencial es utilizada comúnmente para modelar el tiempo que transcurre entre eventos sucesivos en un proceso con tasa constante. En R, las funciones `dexp` para la densidad y `pexp` para la distribución acumulativa manejan estas consultas, utilizando el parámetro de tasa `rate`, que corresponde a $1/\beta$ en la formulación matemática de la distribución.

> El tiempo en horas que se tarda en llegar desde la empresa de mantenimiento de aerogeneradores hasta una determinada instalación sigue una distribución exponencial de parámetro $\beta = 2$. ¿Cuál es la probabilidad de que un operario tarde más de tres horas en llegar a la instalación?

La función `pexp(3, rate = 2, lower.tail = FALSE)` calculará esta probabilidad, donde `3` es el tiempo en horas que queremos evaluar, `rate = 2` es el parámetro de la distribución exponencial, y `lower.tail = FALSE` indica que buscamos la probabilidad de que el tiempo sea mayor que 3 horas.

La fórmula correspondiente para la función de distribución acumulativa de la exponencial en este contexto es:

$$P(T > t) = 1 - F(t) = e^{-\beta t}$$

Aplicando nuestros valores:

```
prob_mas_3h <- pexp(3, rate = 2, lower.tail = FALSE)

cat(sprintf("La probabilidad de que un operario tarde más de tres horas en llegar a la instalación%.4f.
```

```
## La probabilidad de que un operario tarde más de tres horas en llegar a la instalación 0.0025.
```

Este comando proporciona la probabilidad de que el tiempo hasta el próximo evento (en este caso, la llegada del operario) sea mayor que 3 horas.

Supongamos que el tiempo de respuesta de un servidor web sigue una distribución exponencial con una tasa de 0.5 respuestas por segundo, es decir, el tiempo medio de respuesta es de 2 segundos ($\beta = 2$). Queremos calcular la probabilidad de que el servidor responda en menos de 1 segundo.

> El tiempo de respuesta de un servidor web sigue una distribución exponencial con parámetro $\beta = 2$. ¿Cuál es la probabilidad de que el servidor responda en menos de 1 segundo?

Podemos usar la función `pexp` con el parámetro `lower.tail = TRUE`, que calcula la probabilidad acumulada de que el tiempo de respuesta sea **menor** que 1 segundo.

```r
rate <- 1/2  # Tasa inversa del tiempo medio de respuesta
time <- 1  # Tiempo que queremos evaluar

# Probabilidad de que el servidor responda en menos de 1 segundo
prob_menos_1s <- pexp(time, rate = rate, lower.tail = TRUE)

cat(sprintf("La probabilidad de que el servidor responda en menos de 1 segundo es%.4f.\n", prob_menos_1
```

```
## La probabilidad de que el servidor responda en menos de 1 segundo es 0.3935.
```

La distribución exponencial también se utiliza para modelar el tiempo hasta la falla de dispositivos, como bombillas. Supongamos que el tiempo de vida de una bombilla sigue una distribución exponencial con un promedio de vida útil de 1000 horas (lo que implica que el parámetro $\beta = 1000$). Queremos calcular la probabilidad de que una bombilla dure más de 1500 horas.

> El tiempo de vida útil de una bombilla sigue una distribución exponencial con parámetro $\beta = 1000$. ¿Cuál es la probabilidad de que una bombilla dure más de 1500 horas?

En este caso, podemos usar la función `pexp` para calcular la probabilidad de que la vida útil de una bombilla sea mayor a 1500 horas. Dado que el parámetro de tasa `rate` en R es la inversa de β, el valor de `rate` será 1/1000.

```r
rate <- 1/1000  # Tasa inversa del tiempo medio de vida
time <- 1500  # Tiempo que queremos evaluar

# Probabilidad de que una bombilla dure más de 1500 horas
prob_mas_1500h <- pexp(time, rate = rate, lower.tail = FALSE)
```

2.2.4. Distribución normal

La distribución normal es una de las distribuciones más importantes en estadística, utilizada para modelar fenómenos naturales, sociales y de investigación científica cuando los datos se distribuyen en forma de campana alrededor de un promedio. En R, las funciones `dnorm` y `pnorm` nos permiten trabajar con la densidad y la distribución acumulativa de la normal, respectivamente, sin necesidad de tipificar manualmente las variables, ya que podemos especificar directamente la media (`mean`) y la desviación estándar (`sd`). Su función de densidad de probabilidad es:

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

Para calcular probabilidades acumuladas y encontrar cuantiles en una distribución normal, podemos usar las funciones `pnorm` y `qnorm` en R, que incorporan estos cálculos matemáticos internamente:

- `pnorm(x, mean, sd)` calcula la probabilidad $P(X \leq x)$ para una variable normal X con media `mean` y desviación estándar `sd`.
- `qnorm(p, mean, sd)` encuentra el valor x tal que $P(X \leq x) = p$.

> El peso de los paquetes que contienen los pedidos que recibe un laboratorio se distribuye según una distribución normal de media 1,8 y desviación típica 0,5 kg. ¿Cuál es la probabilidad de que un paquete esté entre 1 y 2 kilos?¿Por debajo de qué peso estarán probablemente al menos el 95 % de los paquetes? Por último, realiza una simulación de 100 paquetes y haz un histograma.

Para calcular la probabilidad de que el peso de un paquete se encuentre en un intervalo específico, utilizamos la función `pnorm` para la distribución normal. En este caso, queremos saber la probabilidad de que un paquete pese entre 1 y 2 kilos, dados una media de 1.8 kg y una desviación estándar de 0.5 kg. La función `pnorm` nos da la probabilidad acumulada hasta un punto dado, por lo que calcularemos la probabilidad de que un paquete pese menos o igual a 2 kilos y restaremos la probabilidad de que pese menos o igual a 1 kilo:

```
prob_paquete_12 <- pnorm(2, 1.8, 0.5) - pnorm(1, 1.8, 0.5)

cat(sprintf("La probabilidad de que un paquete esté entre 1 y 2 kilos%.4f.\n", prob_paquete_12))
```

```
## La probabilidad de que un paquete esté entre 1 y 2 kilos 0.6006.
```

En cuanto al segundo cálculo, si deseamos encontrar un umbral de peso por debajo del cual se encuentre al menos el 95 % de los paquetes, utilizaremos la función `qnorm`. Esta función devuelve el cuantil o valor crítico asociado a una probabilidad acumulada dada. Para obtener el peso que no supera el 95 % de los paquetes, establecemos `p = 0.95`, junto con la media (`mean = 1.8`) y desviación estándar (`sd = 0.5`) especificadas:

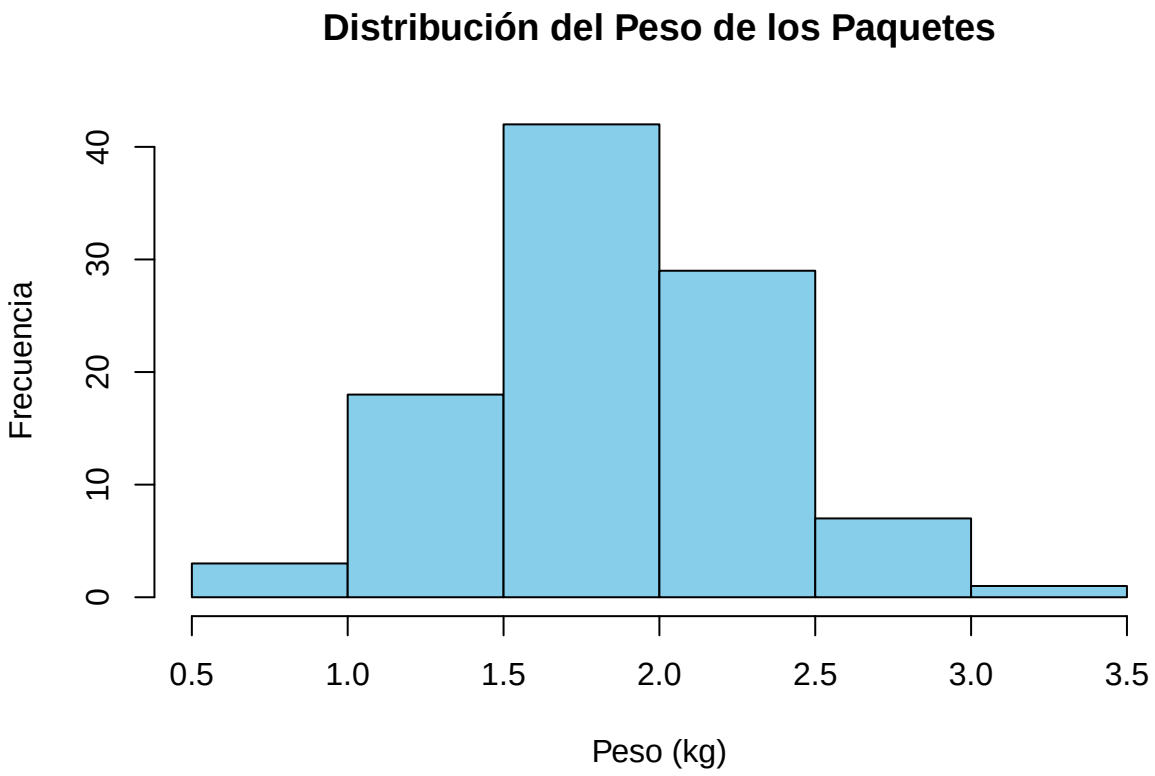
```
menos_95_kg <- qnorm(p = 0.95, 1.8, 0.5)

cat(sprintf("El 95%% de los paquetes se encuentran por debajo de%.2fkg.\n", menos_95_kg))
```

```
## El 95% de los paquetes se encuentran por debajo de 2.62kg.
```

Para visualizar cómo se distribuye el peso de los paquetes basándonos en nuestros parámetros especificados de la distribución normal, podemos realizar una simulación. Utilizamos la función `rnorm` para generar 100 valores aleatorios que siguen una distribución normal con una media de 1.8 kg y una desviación estándar de 0.5 kg. Para garantizar que la simulación sea reproducible, establecemos una semilla inicial con `set.seed(1)`. Después de generar los datos simulados, creamos un histograma para visualizar la distribución del peso de los 100 paquetes simulados:

```
set.seed(1)
simu <- rnorm(n = 100, mean = 1.8, sd = 0.5)
hist(simu, main = "Distribución del Peso de los Paquetes", xlab = "Peso (kg)", ylab = "Frecuencia", col
```



Este histograma nos ofrece una representación gráfica de la distribución esperada del peso de los paquetes, permitiéndonos observar la variabilidad y la tendencia central de los datos simulados basados en la distribución normal.