

Práctica 4. Cálculo de probabilidades en R.
Christian J. Acal González y Miguel Ángel Montero Alonso



**UNIVERSIDAD
DE GRANADA**

Todo el material para el conjunto de actividades de este curso ha sido elaborado y es propiedad intelectual del grupo **BioestadísticaR** formado por:

Antonio Martín Andrés
Juan de Dios Luna del Castillo,
Pedro Femia Marzo,
Miguel Ángel Montero Alonso,
Christian José Acal González,
Pedro María Carmona Sáez,
Juan Manuel Melchor Rodríguez,
José Luis Romero Béjar,
Manuela Expósito Ruíz,
Juan Antonio Villatoro García.

Todos los integrantes del grupo han participado en todas las actividades, en su elección, construcción, correcciones o en su edición final, no obstante, en cada una de ellas, aparecerán uno o más nombres correspondientes a las personas que han tenido la máxima responsabilidad de su elaboración junto al grupo de **BioestadísticaR**.

Todos los materiales están protegidos por la Licencia Creative Commons **CC BY-NC-ND** que permite "descargar las obras y compartirlas con otras personas, siempre que se reconozca su autoría, pero no se pueden cambiar de ninguna manera ni se pueden utilizar comercialmente".

Práctica 4. Cálculo de probabilidades en R.

Christian J. Acal González y Miguel Ángel Montero Alonso

4.1 Cálculo de probabilidades en R

En la práctica anterior se vieron los distintos tipos de datos que pueden representarse, así como el modo de describirlos y resumirlos en el caso de disponer de una muestra de una población. En este Tema el objetivo es describir y resumir los datos de toda la población. Tanto para la práctica anterior (descripción de poblaciones) como para el actual (y los que siguen), es crucial la teoría de la probabilidad en la que existen múltiples modelos teóricos que resultan de utilidad para modelizar un gran número de situaciones reales. La teoría de probabilidad es una rama de la Estadística que abarcaría muchísimo tiempo estudiarla al detalle. En el presente curso se intenta dar una visión general de esta rama, enfocando la metodología desde un modo muy utilitario y evitando complejidades innecesarias. En particular, en este tema, se introduce el cálculo de probabilidades con R. El concepto de probabilidad surge cuando los resultados de un fenómeno se producen al azar (aleatoriamente). Cuando tales resultados son numéricos (o se pueden convertir en números) se dice que el resultado observado es una variable aleatoria que será discreta o continua según qué valores posibles pueda tomar. Estas variables aleatorias siguen exactamente (o de forma aproximada) unas leyes bien conocidas que son llamadas distribuciones de probabilidad teóricas. Cada una de ellas difieren una de otras sólo en sus parámetros pudiéndose las estudiar de un modo global. Existen muchas distribuciones de probabilidad pero, dado el carácter introductorio de este curso, se limitará a estudiar la distribución binomial y la distribución de Poisson como ejemplos de distribuciones discretas y la distribución Normal para ilustrar las distribuciones continuas.

4.1.1 Distribución Binomial

Casi siempre que un investigador se pregunta por la proporción de individuos que verifican una cierta característica dicotómica (estar enfermo o no, ser varón o no, etc.) subyace en el problema una distribución Binomial. Si de una población se toma una muestra de tamaño n y se anota el número X de individuos de ella que sí verifican la característica, entonces la variable aleatoria X sigue distribución Binomial. Asociada a si cada individuo sigue o no la característica de interés, aparece el término probabilidad/proporción (p) que denota la probabilidad de que el individuo presente la característica o la proporción de individuos que presentan la característica en la población. En consecuencia, esta distribución que se representa como $B(n, p)$, puede verse como el número de éxitos en n repeticiones del experimento con probabilidad p , donde el éxito sería sí se verifica la condición y el experimento sería el hecho de comprobar si el individuo cumple la característica en cuestión. Con respecto al cálculo de probabilidades de una variable aleatoria con distribución Binomial, en R se distinguen cuatro funciones:

- **dbinom(x, size=n, prob=p)**. Devuelve el valor (valores) x de la función masa de probabilidad de una variable con distribución Binomial de parámetros n y p . La función masa de probabilidad es una función que caracteriza a las variables aleatorias discretas y que asigna una determinada probabilidad a cada uno de los valores de la variable.
- **pbinom(x, size=n, prob=p, lower.tail = TRUE)**. Calcula el valor (valores) x de la función de distribución de una distribución Binomial de parámetros n y p . Se define la función de distribución de la variable aleatoria X como la probabilidad de que X tome un valor menor o igual que x , es decir, $P[X \leq x]$. Si **lower.tail=FALSE**, R calcula $P[X > x]$, lo cual ya no sería la función de distribución.

- **qbinom(a, size=n, prob=p, lower.tail = TRUE)**. Determina el cuantil (cuantiles) de una variable con distribución Binomial de parámetros n y p . Por definición, un cuantil es aquel que deja a su izquierda una proporción de valores a , es decir, es aquel valor q tal que $P[X \leq q] = a$.
- **rbinom(r, size=n, prob=p)**. Genera r valores aleatorios de una distribución Binomial, $B(n,p)$.

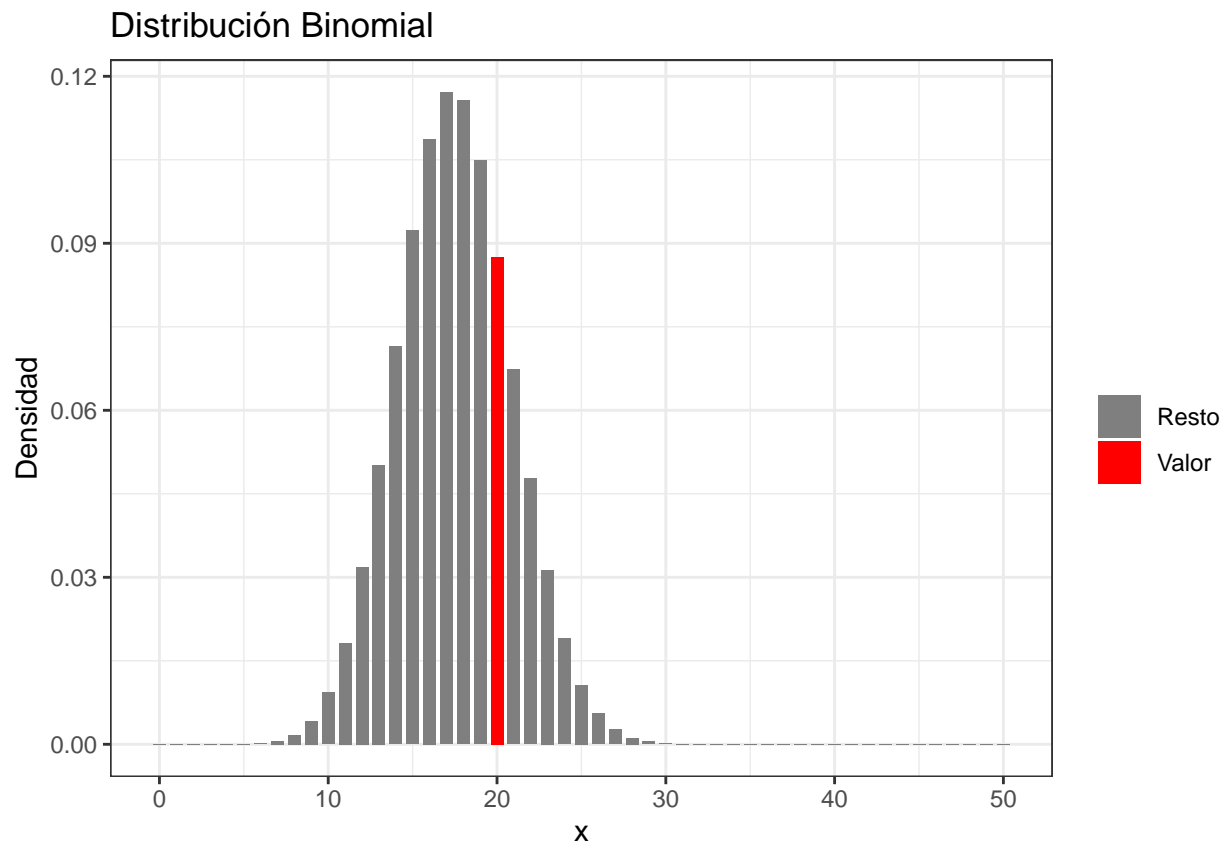
Para poner en práctica estas funciones se propone el siguiente ejercicio. *En un ensayo clínico se ha tomado un total de 50 pacientes y la probabilidad de que el paciente fume es 0.35. Claramente, la variable aleatoria que cuenta el número de pacientes fumadores sigue una distribución Binomial de parámetros $n = 50$ y $p = 0.35$.*

1. *Calcular la probabilidad de que exactamente 20 pacientes sean fumadores.* Aquí se pide calcular $P[X = 20]$ siendo X una variable aleatoria que representa el número de pacientes fumadores.

```
n=50
p=0.35
valor=20
dbinom(valor,size=n,prob=p)
```

```
## [1] 0.08750881
```

```
library(ggplot2)
rango = 0:n
df = data.frame(x = rango, y = dbinom(rango, size=n, prob=p))
ggplot(df, aes(x = x, y = y, fill=factor(ifelse(x==valor,"Valor","Resto")))) +
  geom_bar(stat = "identity",width = 0.75) + xlab("x") + ylab("Densidad") +
  ggtitle("Distribución Binomial") +
  scale_fill_manual(name = "", values=c("grey50","red")) + theme_bw()
```



2. Obtener la probabilidad de que al menos 15 pacientes sean fumadores. Aquí se pide calcular $P[X \geq 15] = P[X > 14]$, que sería equivalente a $P[X \geq 15] = 1 - P[X < 15] = 1 - P[X \leq 14]$. Se procede a calcular las dos posibilidades con las opciones habilitadas en R.

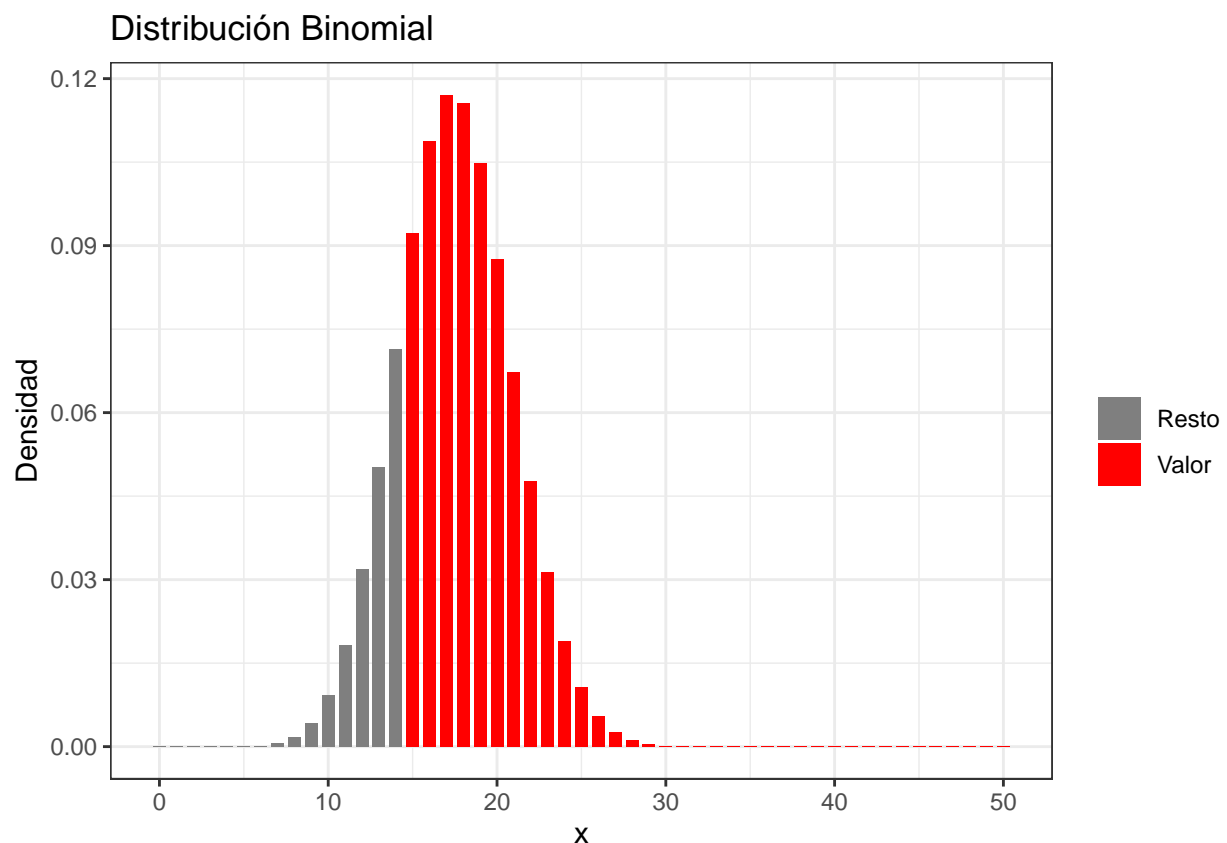
```
valor=14  
pbinom(valor,size=n,prob=p,lower.tail=FALSE)
```

```
## [1] 0.812223
```

```
1-pbinom(valor,size=n,prob=p)
```

```
## [1] 0.812223
```

```
ggplot(df, aes(x = x, y = y, fill=factor(ifelse(x>valor,"Valor","Resto")))) +  
  geom_bar(stat = "identity",width=0.75) + xlab("x") + ylab("Densidad") +  
  ggtitle("Distribución Binomial") +  
  scale_fill_manual(name = "", values=c("grey50","red")) + theme_bw()
```

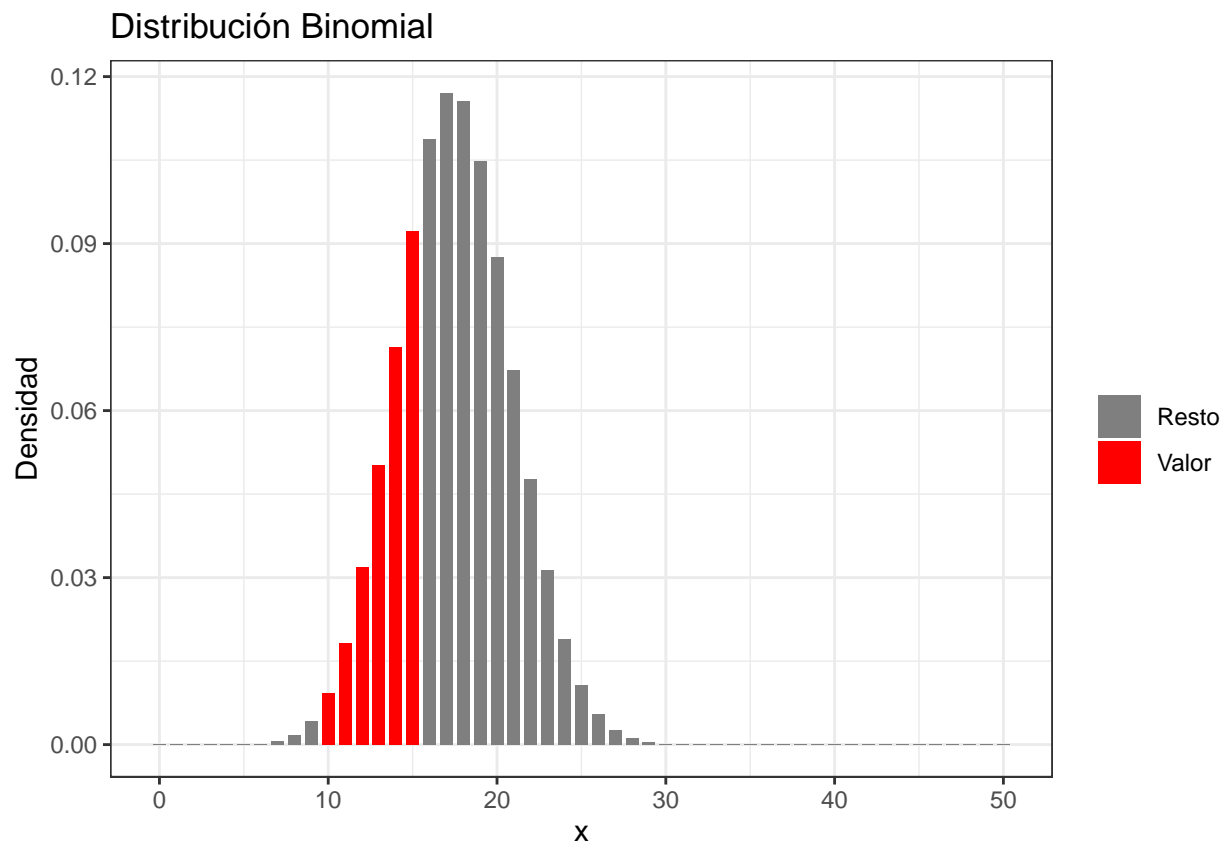


3. Obtener la probabilidad de que entre 10 y 15 pacientes (ambos inclusive) sean fumadores. Se pide calcular $P[10 \leq X \leq 15] = P[X \leq 15] - P[X < 10] = P[X \leq 15] - P[X \leq 9]$

```
valor1=15
valor2=9
pbinom(valor1,size=n,prob=p)-pbinom(valor2,size=n,prob=p)
```

```
## [1] 0.2734065
```

```
ggplot(df, aes(x = x, y = y, fill=factor(ifelse(x>=valor2+1 & x<=valor1,"Valor","Resto"))))+
  geom_bar(stat = "identity",width=0.75) + xlab("x") + ylab("Densidad") +
  ggtitle("Distribución Binomial") +
  scale_fill_manual(name = "", values=c("grey50","red")) + theme_bw()
```

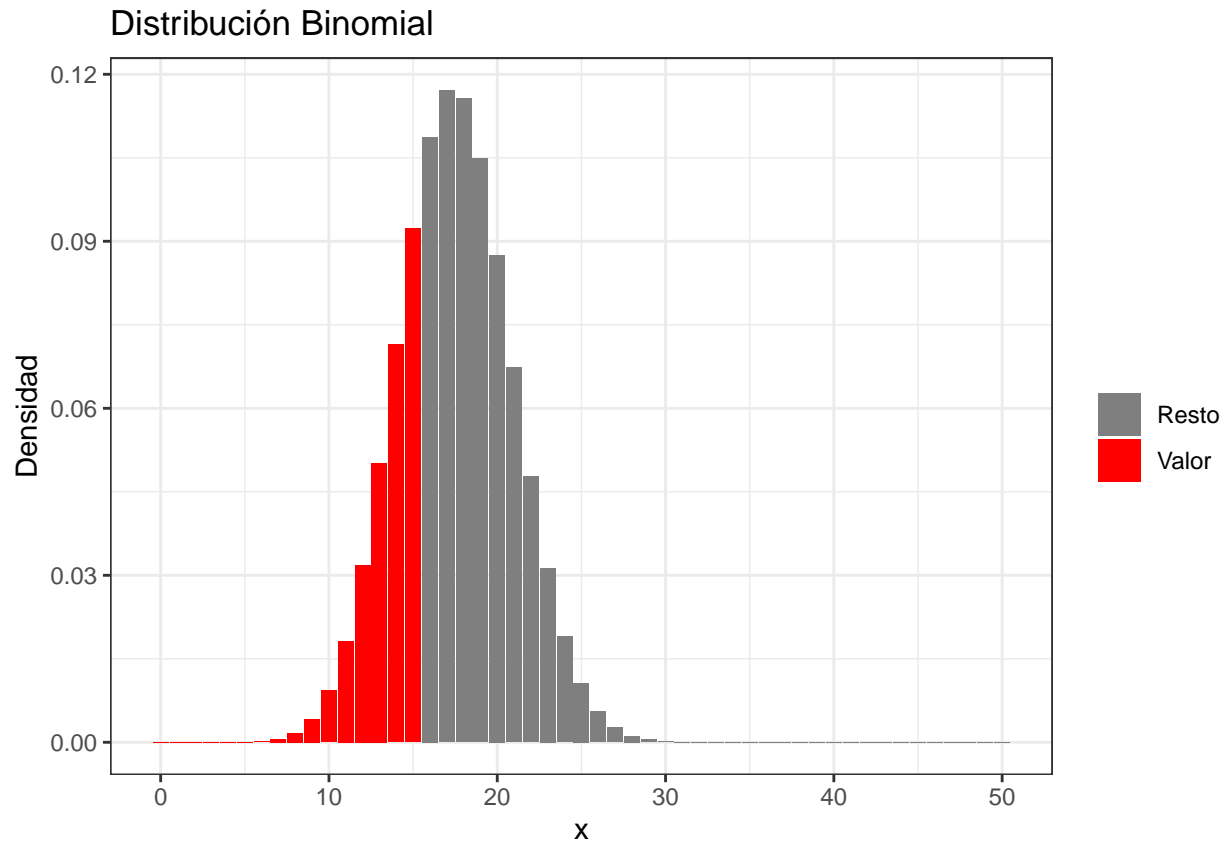


4. Calcular el valor de la variable tal que deja a su derecha un 70% de las observaciones. El valor de la variable que deja a su derecha un 70% de las observaciones es el mismo que deja a su izquierda el 30% restante, por lo que se pide calcular el valor q tal que $P[X \leq q] = 0.30$.

```
a=0.30
qbinom(a,size=n,prob=p)
```

```
## [1] 16
```

```
ggplot(df, aes(x = x, y = y, fill=factor(ifelse(cumsum(y)<=a,"Valor","Resto")))) +
  geom_bar(stat = "identity") + xlab("x") + ylab("Densidad") +
  ggtitle("Distribución Binomial") +
  scale_fill_manual(name = "", values=c("grey50","red")) + theme_bw()
```



5. Generar una muestra de 10 valores aleatorios de esta distribución.

```
rbinom(10,size=n,prob=p)
```

```
## [1] 19 19 14 24 21 10 11 14 13 10
```

Nota: Como se están generando números aleatorios, cada vez que se ejecute la orden anterior, R generará valores distintos. Si se está interesado en generar siempre los mismos números, se puede fijar una semilla mediante la función `set.seed()`. Dentro de la función se incluirá un número entero (la semilla) en el que para cada valor, R generará una secuencia de números distinta.

```
set.seed(1) #Se fija la semilla para producir los mismos valores
rbinom(10,size=n,prob=p)
```

```
## [1] 15 16 18 22 15 22 23 19 19 12
```

```
set.seed(1)
rbinom(10,size=n,prob=p)
```

```
## [1] 15 16 18 22 15 22 23 19 19 12
```

```
set.seed(2)
rbinom(10,size=n,prob=p)
```

```
## [1] 14 19 18 14 23 23 14 21 17 18
```

4.1.2 Distribución Poisson

Se dice que una variable aleatoria X sigue una distribución de Poisson de parámetro λ , abreviadamente $X \sim P(\lambda)$, si representa el número de sucesos independientes que ocurren a una velocidad constante en un intervalo de tiempo o en una región del espacio, siendo λ la media de ocurrencias en el intervalo considerado o en la región del espacio considerada. Al ser la distribución de Poisson también una distribución discreta, los valores que se pueden calcular son los mismos que los estudiados para la distribución binomial (función masa de probabilidad, función de distribución, cuantiles y generación de valores aleatorios). La única salvedad radica en que hay que especificar la distribución Poisson y sustituir el parámetro λ por los parámetros n y p de la distribución Binomial, es decir:

- **dpois(x, lambda)**. Devuelve el valor (valores) x de la función masa de probabilidad de una variable con distribución Poisson de parámetro λ .
- **ppois(x, lambda, lower.tail = TRUE)**. Calcula el valor (valores) x de la función de distribución de una distribución Poisson de parámetro λ . Si **lower.tail=FALSE**, R calcula $P[X > x]$, lo cual ya no sería la función de distribución.
- **qpois(a, lambda, lower.tail = TRUE)**. Determina el cuantil (cuantiles) de una variable con distribución Poisson de parámetro λ .
- **rpois(r, lambda)**. Genera r valores aleatorios de una distribución Poisson de parámetro λ .

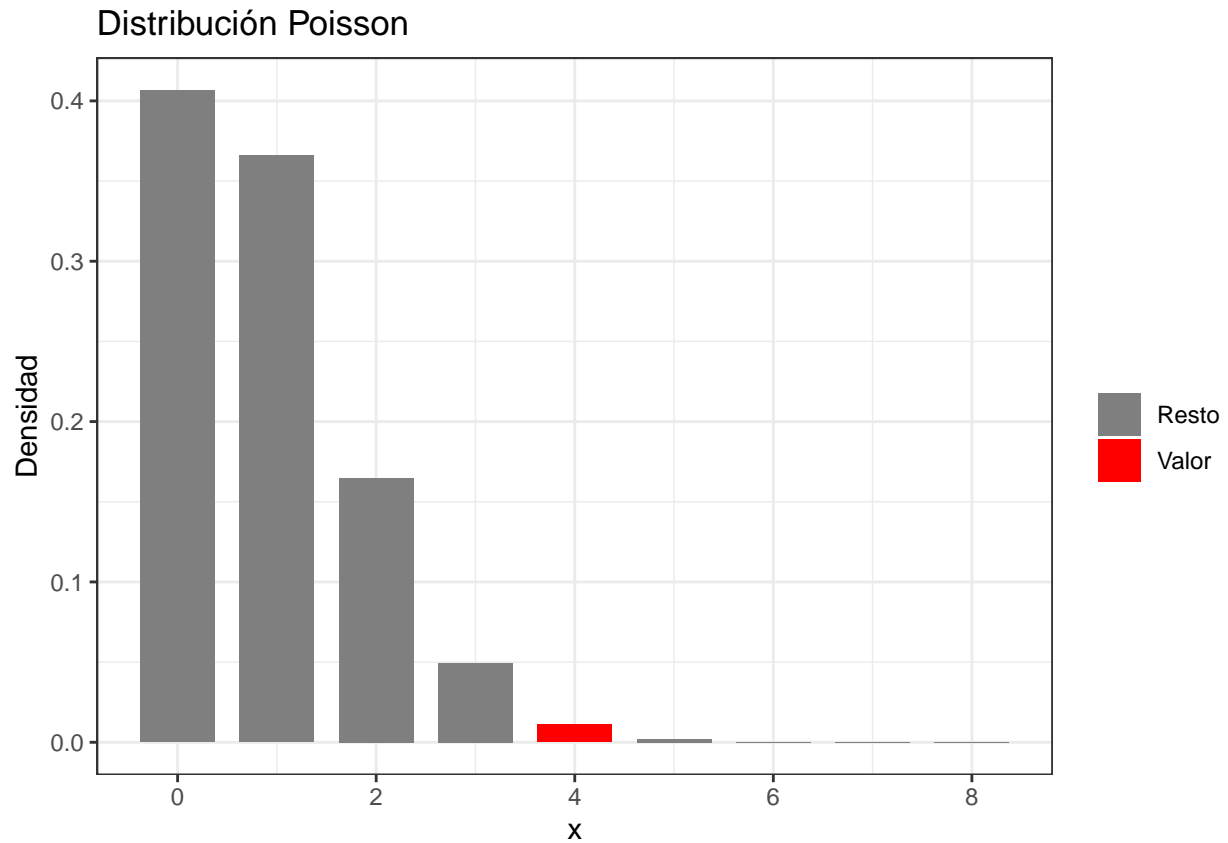
Para poner en práctica estas funciones se propone el siguiente ejercicio. *En un hospital el número medio de pancreatitis agudas atendidas por día es 0.90. Es fácil comprobar que la variable aleatoria que cuenta el número de pancreatitis agudas atendidas por día sigue una distribución Poisson de parámetro $\lambda = 0.90$.*

1. *Calcular la probabilidad de que en un día dado, el hospital atienda exactamente a 4 pacientes.* Se pide calcular $P[X = 4]$.

```
lambda=0.90
valor=4
dpois(valor,lambda = lambda)
```

```
## [1] 0.0111146
```

```
rango = 0:8
df = data.frame(x = rango, y = dpois(rango, lambda=lambda))
ggplot(df, aes(x = x, y = y, fill=factor(ifelse(x==valor,"Valor","Resto")))) +
  geom_bar(stat = "identity",width = 0.75) + xlab("x") + ylab("Densidad") +
  ggtitle("Distribución Poisson") +
  scale_fill_manual(name = "", values=c("grey50","red")) + theme_bw()
```

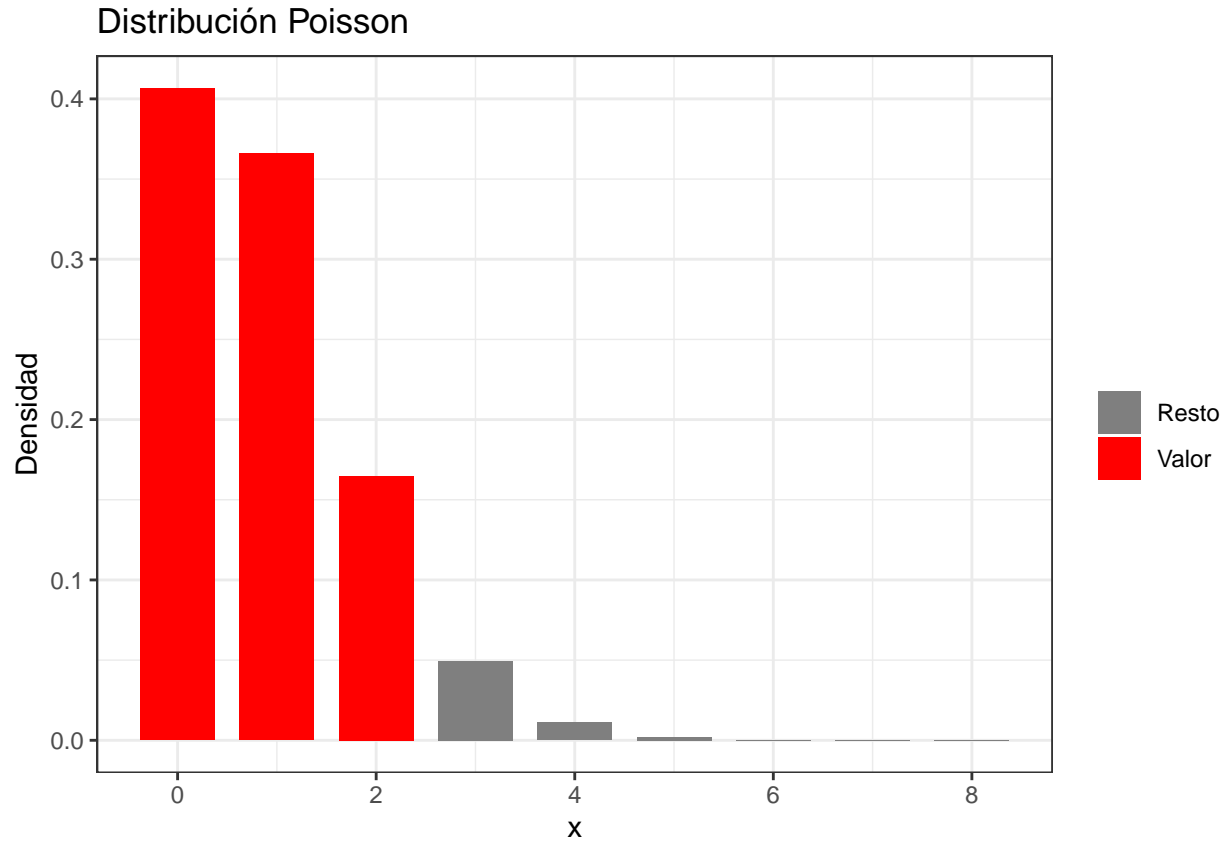


2. Calcular la probabilidad de que, en un día cualquiera, el hospital atienda como máximo 2 clientes. Se pide calcular $P[X \leq 2]$.

```
valor=2  
ppois(valor,lambda = lambda)
```

```
## [1] 0.9371431
```

```
ggplot(df, aes(x = x, y = y, fill=factor(ifelse(x<=valor,"Valor","Resto")))) +  
  geom_bar(stat = "identity",width = 0.75) + xlab("x") + ylab("Densidad") +  
  ggtitle("Distribución Poisson") +  
  scale_fill_manual(name = "", values=c("grey50","red")) + theme_bw()
```

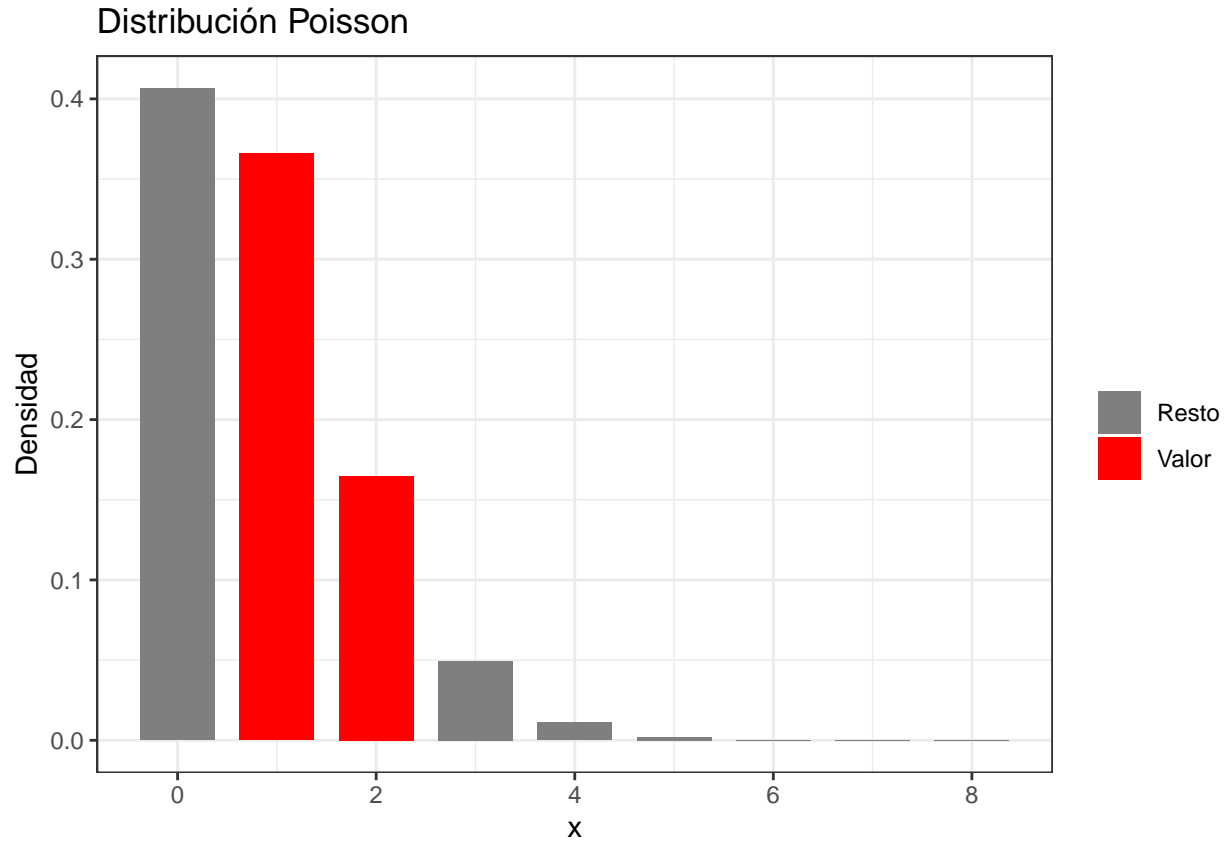



3. Calcular la probabilidad de que, en un día dado, el hospital reciba entre 1 y 2 pacientes (ambos inclusive). Se pide calcular $P[1 \leq X \leq 2] = P[X \leq 2] - P[X < 1] = P[X \leq 2] - P[X \leq 0]$.

```
valor1=2
valor2=0
ppois(valor1,lambda=lambda)-ppois(valor2,lambda)
```

```
## [1] 0.5305734
```

```
ggplot(df, aes(x = x, y = y, fill=factor(ifelse(x>=valor2+1 & x<=valor1,"Valor","Resto")))) +
  geom_bar(stat = "identity",width = 0.75) + xlab("x") + ylab("Densidad") +
  ggtitle("Distribución Poisson") +
  scale_fill_manual(name = "", values=c("grey50","red")) + theme_bw()
```

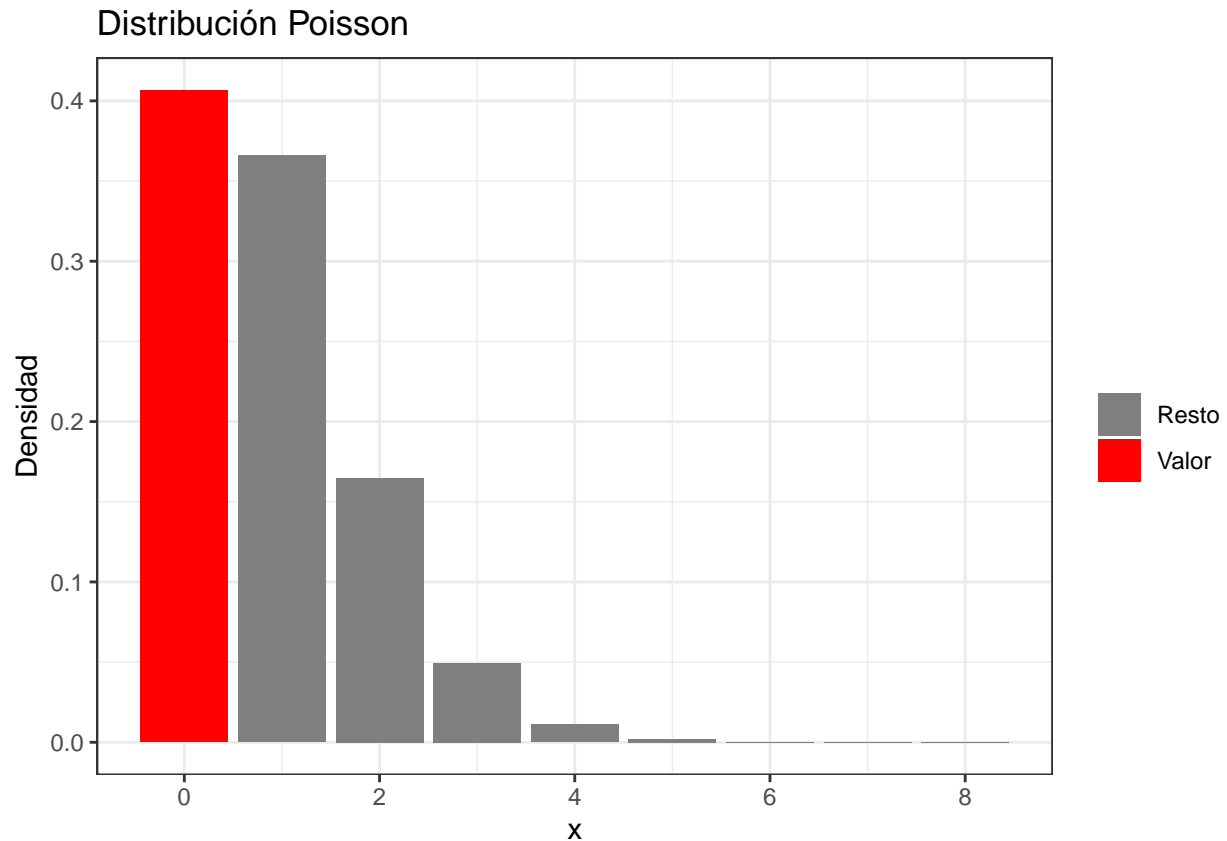


4. *Obtener la mediana de la variable.* La mediana coincide con el cuartil 2 que es aquel que divide a la población (muestra) en dos partes iguales, es decir, a la izquierda queda el 50% de los valores y a la derecha el 50% restante.

```
a=0.50
qpois(a,lambda=lambda)
```

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

```
ggplot(df, aes(x = x, y = y, fill=factor(ifelse(cumsum(y)<=a,"Valor","Resto")))) +
  geom_bar(stat = "identity") + xlab("x") + ylab("Densidad") +
  ggtitle("Distribución Poisson") +
  scale_fill_manual(name = "", values=c("grey50","red")) + theme_bw()
```



5. Generar una muestra de 8 valores aleatorios de la distribución.

```
r=8
rpois(r,lambda=lambda)
```

```
## [1] 1 0 1 0 0 2 3 0
```

4.1.3 Distribución Normal

La distribución Normal es la distribución continua más importante en estadística, ya que una gran mayoría de las variables aleatorias de la Naturaleza siguen dicha distribución y porque presenta unas propiedades muy interesantes y sencillas que hace que sea muy utilizada dentro de esta rama. La distribución Normal queda definida por los parámetros μ y σ que representan la media y la desviación típica, respectivamente, de la distribución. En consecuencia, una variable aleatoria con distribución Normal se denotará como $X \sim N(\mu, \sigma)$.

El tratamiento computacional con R de la distribución Normal es muy similar al que se utiliza con las distribuciones discretas. La principal diferencia radica en que con las distribuciones discretas se trabaja con la función masa de probabilidad y con las distribuciones continuas se usa la función de densidad. La función de densidad de la distribución Normal se calcula a partir de la función `dnorm()` la cual es análoga a las funciones `dbinom()` y `dpois()` de las distribuciones Binomial y Poisson, respectivamente. Sin embargo, el cálculo de probabilidades para valores concretos de una variable continua no tiene sentido, ya que para cualquier variable aleatoria X continua (siga o no distribución Normal) $P[X = x] = 0$ para cualquier valor de x . En consecuencia, `dnorm` devuelve el valor de la función de densidad en un punto (o puntos). Asimismo,

se detallan a continuación los argumentos de las funciones que se usan en R para el cálculo de probabilidades de una distribución Normal:

- **dnorm(x, mean=media, sd=desv.tip)**. Devuelve el valor (valores) x de la función de densidad de una variable con distribución Normal de parámetros μ y σ .
- **pnorm(x, mean=media, sd=desv.tip, lower.tail = TRUE)**. Calcula el valor (valores) x de la función de distribución de una distribución Normal de parámetros μ y σ . Si **lower.tail=FALSE**, R calcula $P[X > x]$, lo cual ya no sería la función de distribución.
- **qnorm(a, mean=media, sd=desv.tip, lower.tail = TRUE)**. Determina el cuantil (cuantiles) de una variable con distribución Normal de parámetros μ y σ .
- **rqnorm(r, mean=media, sd=desv.tip)**. Genera r valores aleatorios de una distribución Normal de parámetros μ y σ .

Notar que si no se especifica los valores de μ y σ , R tomará por defecto $\mu = 0$ y $\sigma = 1$.

Para poner en práctica estas funciones se propone el siguiente ejercicio. *Se sabe que el peso en kg de los recién nacidos varones sigue una $N(3, 0.3)$.*

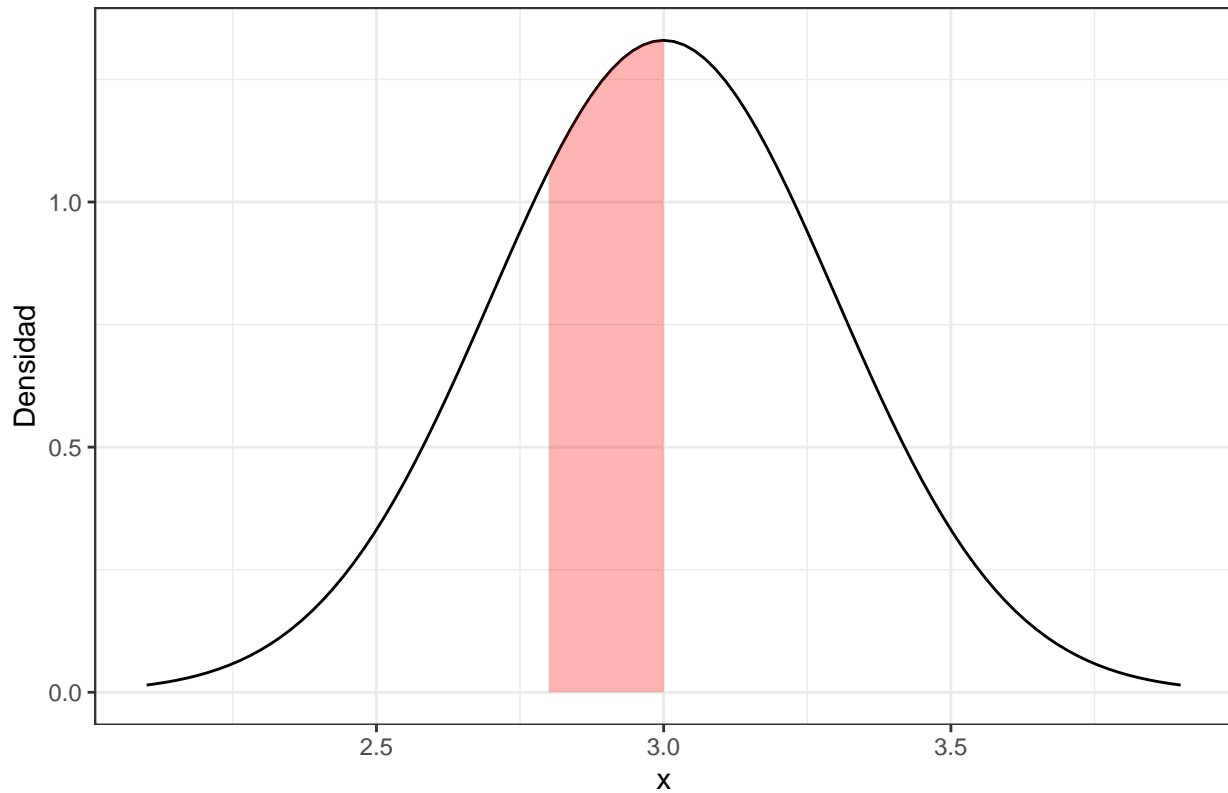
1. *Calcular la probabilidad de que un bebé varón pese entre 2.8 kg y 3 kg (ambos inclusive).* Se pide calcular la probabilidad de que $P[2.8 \leq X \leq 3] = P[X \leq 3] - P[X < 2.8] = P[X \leq 3] - P[X \leq 2.8]$, siendo X una variable aleatoria con distribución Normal que representa el peso en gramos de los recién nacidos varones.

```
media=3
desv.tip=0.30
valor1=3
valor2=2.8
pnorm(valor1,mean=media,sd=desv.tip)-pnorm(valor2,mean=media,sd=desv.tip)
```

```
## [1] 0.2475075
```

```
rango=c(media-3*desv.tip, media+3*desv.tip)
ggplot(data.frame(x = rango), aes(x=rango)) +
  stat_function(fun = dnorm, n = 101, args = list(mean = media, sd = desv.tip))+
  geom_area(stat = 'function',fun = dnorm,args = list(mean = media, sd =desv.tip),
           fill = 'red',xlim = c(valor2, valor1),alpha = 0.3)+
  ylab("Densidad") + ggtitle("Distribución Normal") + xlab("x") + theme_bw()
```

Distribución Normal



2. La probabilidad de que un bebé varón pese más de 3.2 kg. Se pide calcular $P[X > 3.2] = 1 - P[X \leq 3.2]$.

```
valor=3.2
pnorm(valor,mean=media,sd=desv.tip,lower.tail = FALSE)
```

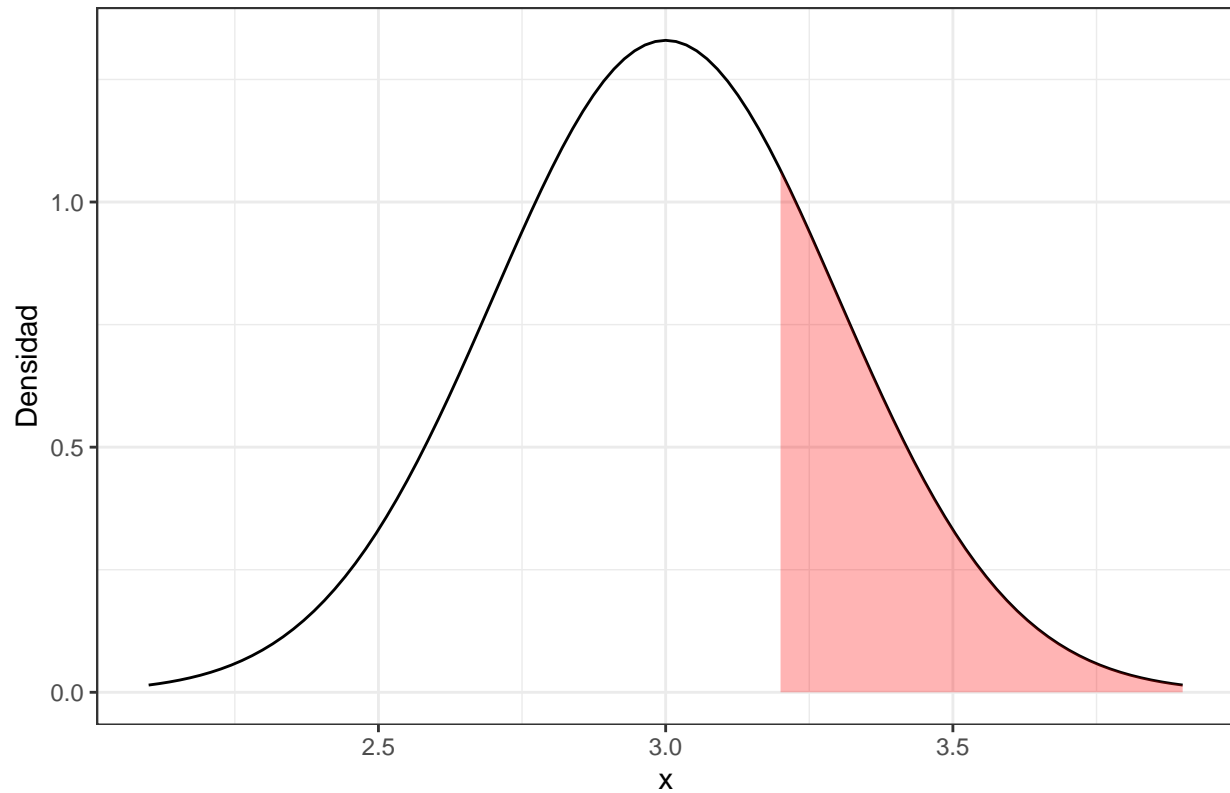
```
## [1] 0.2524925
```

```
1-pnorm(valor,media,sd=desv.tip)
```

```
## [1] 0.2524925
```

```
rango=c(media-3*desv.tip, media+3*desv.tip)
ggplot(data.frame(x = rango), aes(x=rango)) +
  stat_function(fun = dnorm, n = 101, args = list(mean = media, sd = desv.tip))+
  geom_area(stat = 'function',fun = dnorm,args = list(mean = media, sd =desv.tip),
           fill = 'red',xlim = c(valor, max(rango)),alpha = 0.3)+
  ylab("Densidad") + ggtitle("Distribución Normal") + xlab("x") + theme_bw()
```

Distribución Normal



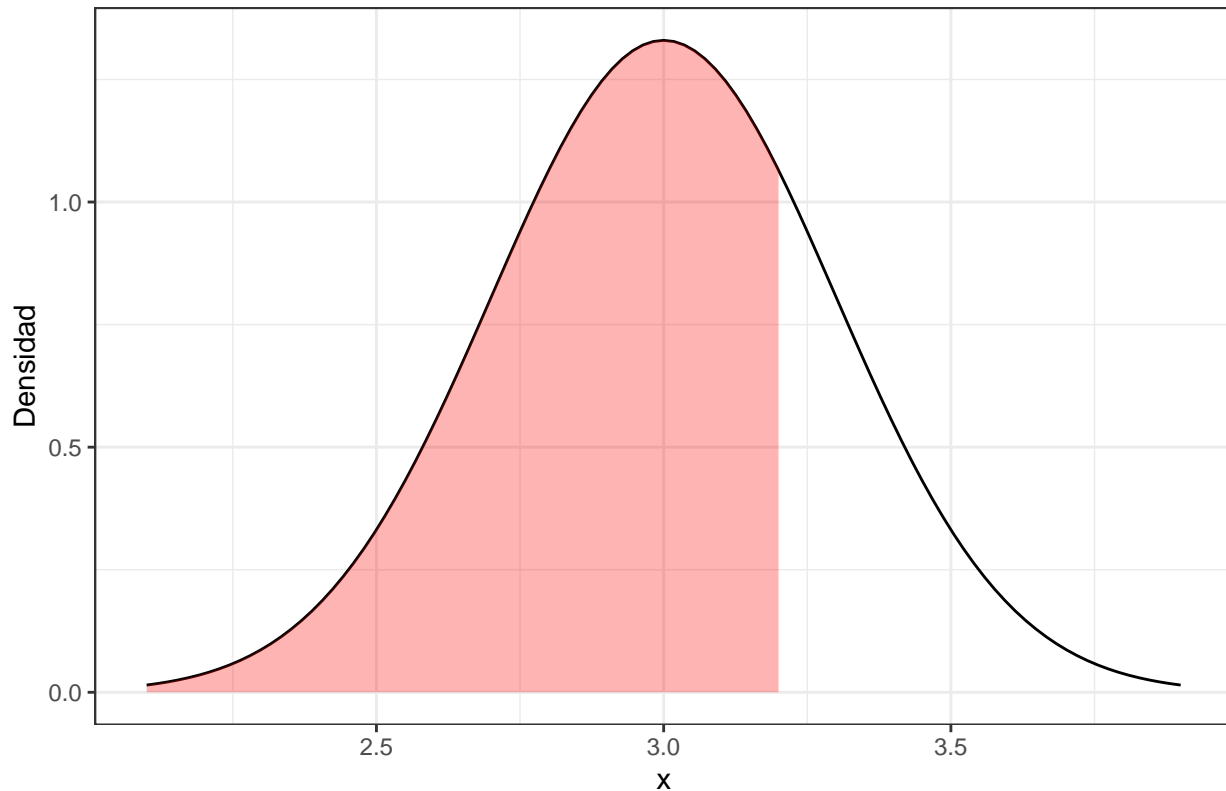
3. La probabilidad de que un bebé varón pese 3.2 kg o menos. Se pide calcular $P[X \leq 3.2]$.

```
valor=3.2
pnorm(valor,mean=media,sd=desv.tip)
```

```
## [1] 0.7475075
```

```
rango=c(media-3*desv.tip, media+3*desv.tip)
ggplot(data.frame(x = rango), aes(x=rango)) +
  stat_function(fun = dnorm, n = 101, args = list(mean = media, sd = desv.tip))+
  geom_area(stat = 'function',fun = dnorm,args = list(mean = media, sd =desv.tip),
           fill = 'red',xlim = c(min(rango), valor),alpha = 0.3)+
  ylab("Densidad") + ggtitle("Distribución Normal") + xlab("x") + theme_bw()
```

Distribución Normal



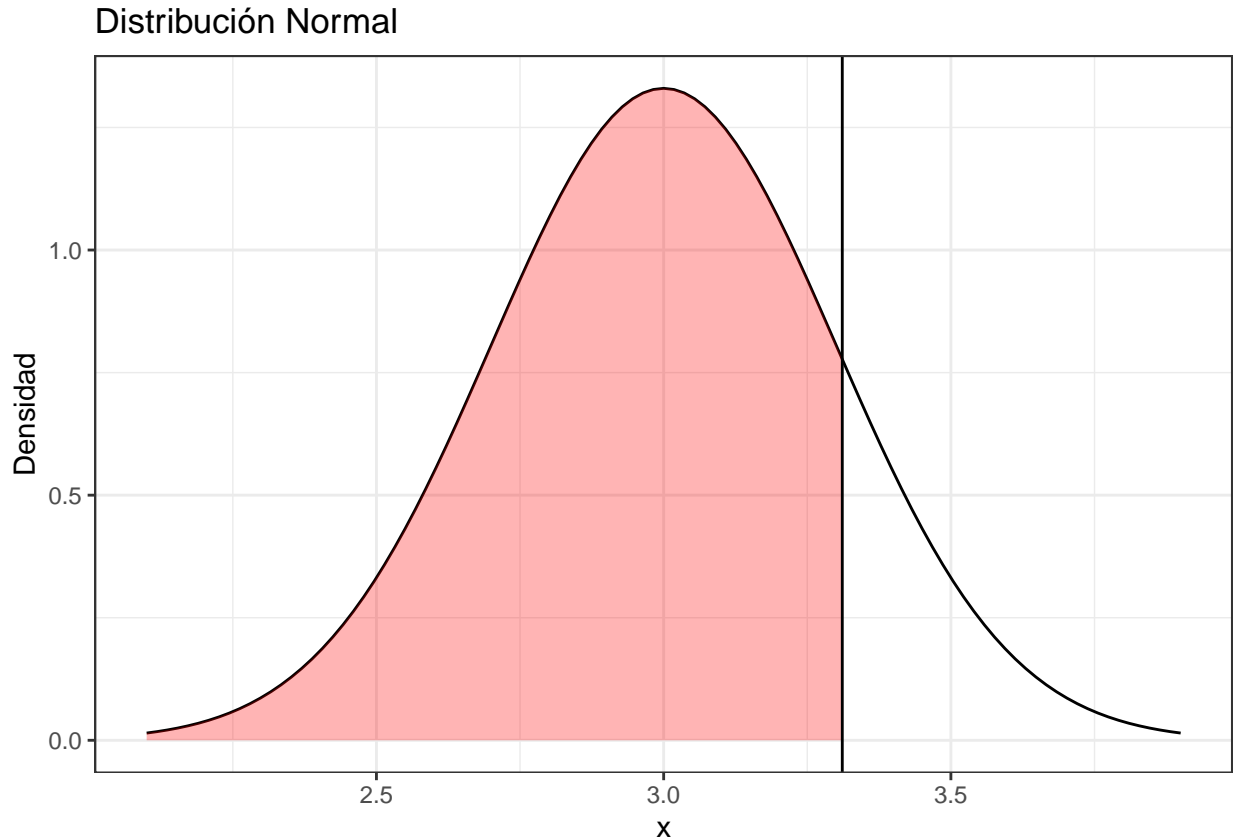
4. *El peso mínimo del 15% de los bebés varones que más pesan.* Se quiere calcular el 15% de los valores más grandes del peso de los bebés varones recién nacidos. Esto es aquel valor que deje a su derecha el 15% de las observaciones, que a su vez, será también aquel que deja a su izquierda el 85% de las observaciones restantes.

```
a=0.85
```

```
qnorm(a,mean=media,sd=desv.tip)
```

```
## [1] 3.31093
```

```
rango=c(media-3*desv.tip, media+3*desv.tip)
ggplot(data.frame(x = rango), aes(x=rango)) +
  stat_function(fun = dnorm, n = 101, args = list(mean = media, sd = desv.tip))+
  geom_area(stat = 'function',fun = dnorm,args = list(mean = media, sd =desv.tip),
           fill = 'red',xlim = c(min(rango), qnorm(a,mean=media,sd=desv.tip)),alpha = 0.3)+
  geom_vline(xintercept=qnorm(a,mean=media,sd=desv.tip))+
  ylab("Densidad") + ggtitle("Distribución Normal") + xlab("x") + theme_bw()
```



5. Generar 15 valores aleatorios que sigan la distribución de la variable aleatoria X .

```
r=15
rnorm(r,mean=media,sd=desv.tip)
```

```
## [1] 2.958364 3.125295 3.294526 2.882191 2.688099 3.534669 2.306679 3.263581
## [9] 3.010742 3.303849 3.129680 3.627246 2.640022 3.476891 3.586395
```

4.2 Extracción de una muestra aleatoria. Generación de números aleatorios

Lo usual es que la *matriz de datos* esté constituida por una muestra de datos, pero si **la matriz de datos fuese toda la población**, y de ella se tuviera que extraer una *muestra aleatoria*, el procedimiento es sencillo. Para el caso general de una población, en la que a sus unidades se les pueden asignar códigos numéricos, el procedimiento habitual para extraer una *muestra aleatoria* está basado en el método de los *números aleatorios*. Más concretamente, y utilizando un ejemplo, si se dispone de la base de datos **osteo.sav** que contiene la información de 94 pacientes diabéticos insulín-dependientes (suponiendo que cada unidad tiene asignado un código numérico entre 1 y 94, variable **num**), y se desea extraer una *muestra aleatoria* de 40 pacientes de dicha población, se necesitaría generar 40 números aleatorios entre 1 y 94, los cuáles determinarían las 40 unidades de la muestra.

Para la generación de los números aleatorios es aconsejable utilizar la función `sample(x,size,replace=FALSE)`, donde x es un vector de la forma $1:n$ siendo n el tamaño total de la población, `size` representa la cantidad

de números a simular y `replace` indica si la extracción de números aleatorios se hace con reemplazamiento (**TRUE**) o no (**FALSE**). Para no extraer el número de un mismo individuo más de una vez, hay que indicar `replace=FALSE`. A continuación se genera 40 números aleatorios entre 1 y 94 que servirán para determinar las 40 unidades de la muestra.

```
set.seed(1)
num.aleatorios=sample(1:94,40,replace=FALSE)
num.aleatorios=sort(num.aleatorios) #Ordenación de los valores
num.aleatorios
```

```
## [1] 1 6 7 14 20 21 25 28 33 34 35 37 38 39 40 42 43 44 51 54 55 56 58 59 60
## [26] 66 68 69 70 71 73 74 75 78 79 82 85 87 91 93
```

NOTA: Se ha fijado la semilla `set.seed(1)` para generar siempre los mismos números aleatorios, puesto que en caso contrario se reproducirían números diferentes cada vez que se ejecute el código. También se ha utilizado la función `sort()` para ordenar los números aleatorios generados de menor a mayor.

Una vez generados los números aleatorios, se procede a extraer las unidades de la muestra, cuyo fichero contiene la información de los 40 pacientes seleccionados.

```
library(foreign)
osteo=read.spss("osteo.sav",to.data.frame = TRUE)
osteo.muestra=osteo[num.aleatorios,]
nrow(osteo.muestra)
```

```
## [1] 40
```

```
head(osteo.muestra)
```

```
##   num edad grupo_edad  sexo peso talla      imc tevol tabaco  alcohol
## 1    1   23      < 25 Hombre 72.3  175 23.60816    2     Sí Moderado
## 6    6   36      > 33 Mujer 61.5  158 24.63548    2     Sí Moderado
## 7    7   21      < 25 Hombre 73.8  180 22.77778    3     Sí Moderado
## 14   14   32   25 - 33 Mujer 51.0  156 20.95661   24     No Moderado
## 20   20   25   25 - 33 Hombre 60.0  165 22.03857   10     No Moderado
## 21   21   20      < 25 Mujer 65.0  158 26.03749    5     Sí Moderado
##           ingca acfis retin nefro neuro hba1c  ca  p  cr pthm pthi bmdcue
## 1 Suficiente  Sí    No    No    No    4.6  9.6 4.2 1.02  2.9 49.0  22.0
## 6 Suficiente  Sí    No    No    No    8.4  9.9 4.7 0.80  1.1 59.9  38.3
## 7 Suficiente  No    No    No    No   10.1  9.8 4.7 1.00  3.4 20.9   6.2
## 14 Suficiente  No    No    No    No    8.1  9.3 3.4 0.84  0.9  NA   NA
## 20 Suficiente  No    No    No    No    8.1  9.8 4.2 1.00  3.3 45.0  15.6
## 21 Suficiente  No    No    No    No   12.1 10.4 3.8 0.80  1.6 24.7  13.4
##           szl24 sztri szcue osteo_cue osteo_tri
## 1    1.07  0.55  1.44      No      No
## 6    0.92  2.05  0.35      No      No
## 7    1.00 -0.75  0.08      No      No
## 14   0.87 -0.30  0.48      No      No
## 20   0.67 -3.13 -2.22      Sí      Sí
## 21   0.77  0.02 -0.77      No      No
```