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 **BioestadisticaR** formado por:
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,
Juan Manuel Praena Fernández,
Miguel Ángel Luque Fernández.

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 **BioestadisticaR**.

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 la presente práctica aprenderemos a calcular probabilidades con R de las distribuciones más empleadas, Normal en el caso de las continuas y, Binomial y Poisson, en el caso de las discretas.

4.1.1 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, aunque sea de forma apróximada, dicha distribución. Además, presenta unas propiedades muy interesantes y sencillas que hace que sea muy utilizada dentro de este área. 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 de probabilidad y con las distribuciones continuas se usa la *función de densidad*. A continuación se detallan 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* $N(3, 0.3)$.

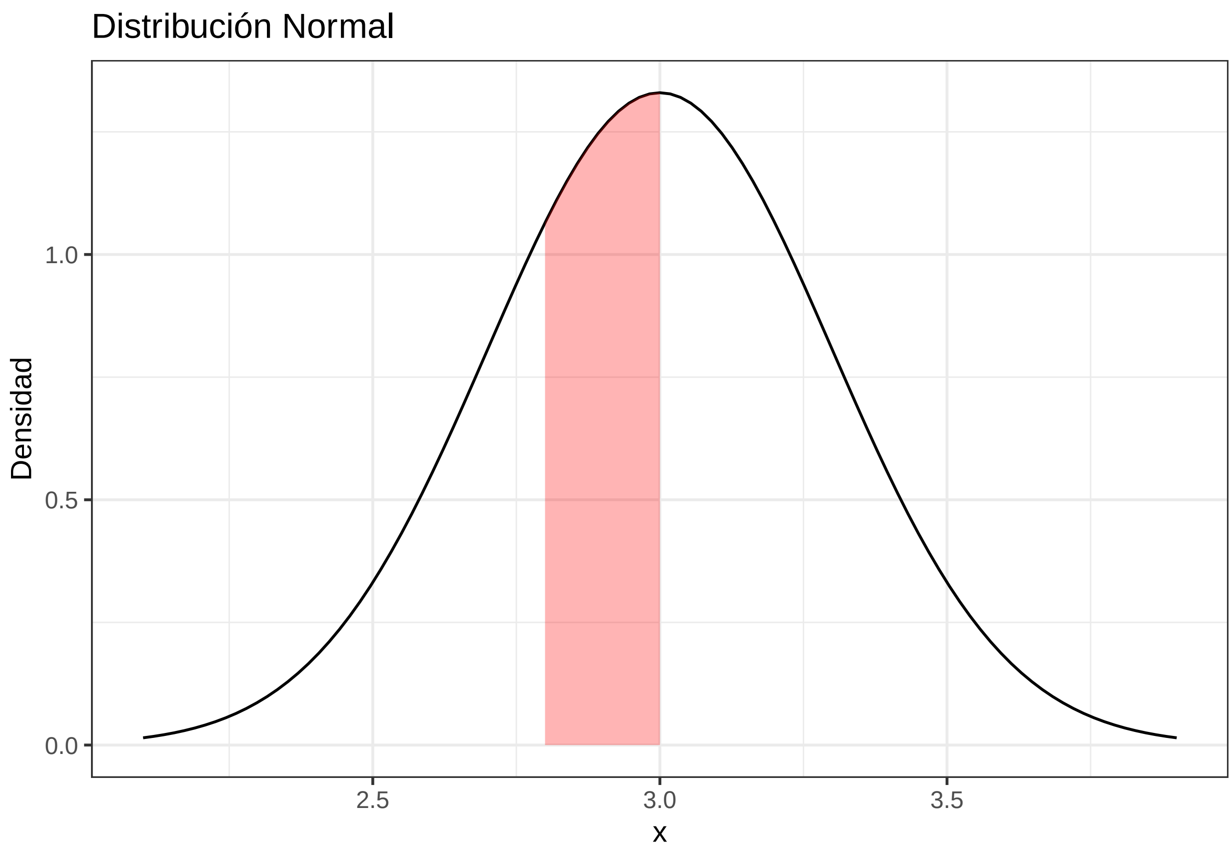
1. *Calcular la probabilidad de que un bebé varón pese entre 2.8 kg y 3 kg (ambos inclusive)*. Se pide calcular la probabilidad de que $P[2.8 \leq X \leq 3] = P[X \leq 3] - P[X < 2.8] = P[X \leq 3] - P[X \leq 2.8]$, siendo X una variable aleatoria con distribución Normal que representa el peso en gramos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

```
library(ggplot2)
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()
```



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

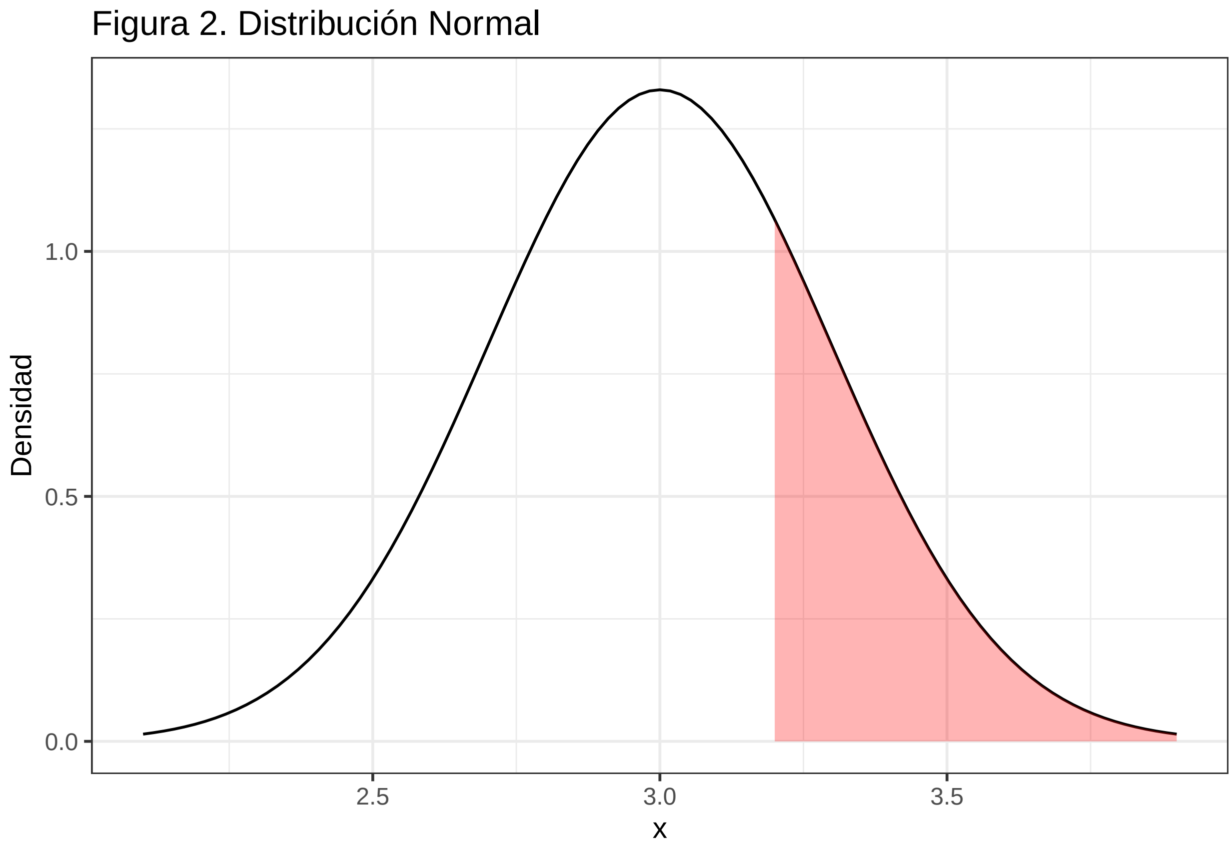
```
valor=3.2
pnorm(valor,mean=media,sd=desv.tip,lower.tail = FALSE)
```

```
## [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("Figura 2. Distribución Normal") + xlab("x") + theme_bw()
```

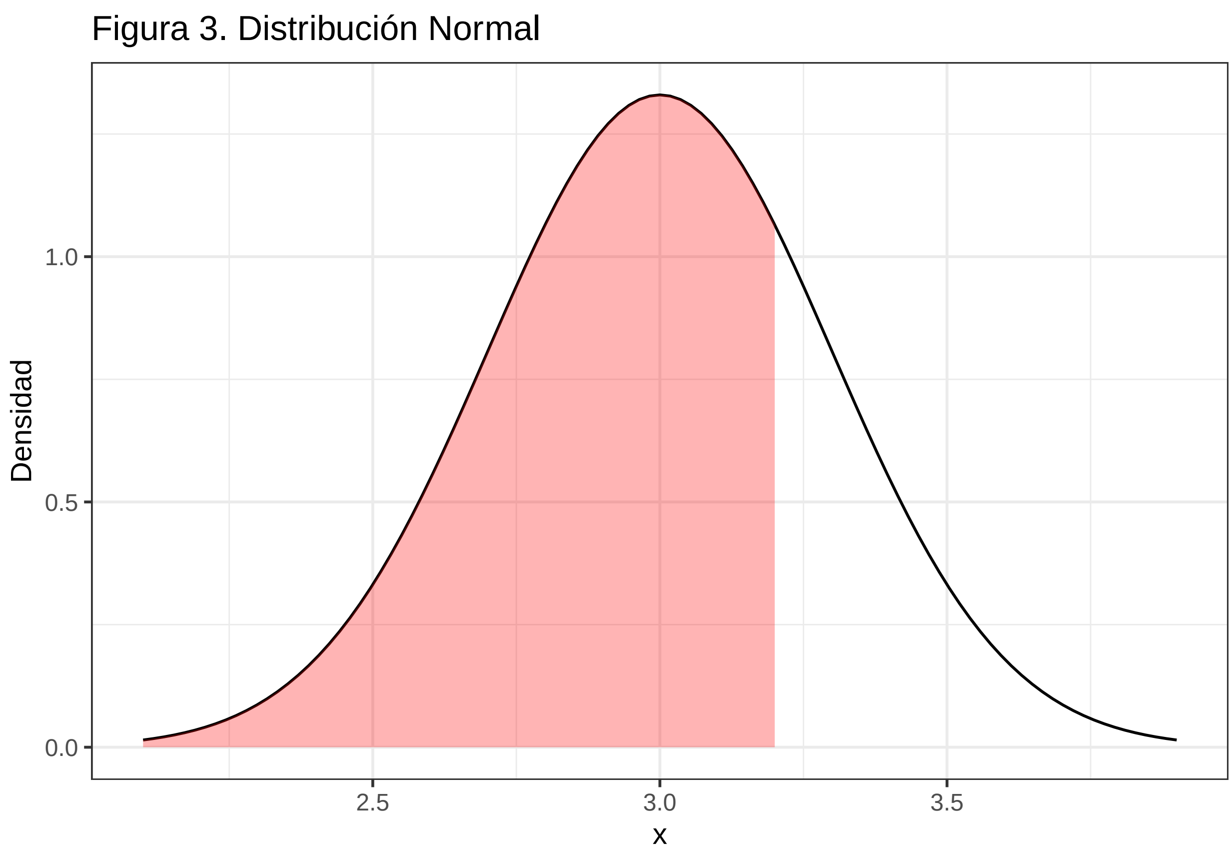


3. *La probabilidad de que un bebé varón pese 3.2 kg o menos.* Se pide calcular $P[X \leq 3.2]$.

```
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("Figura 3. Distribución Normal") + xlab("x") + theme_bw()
```



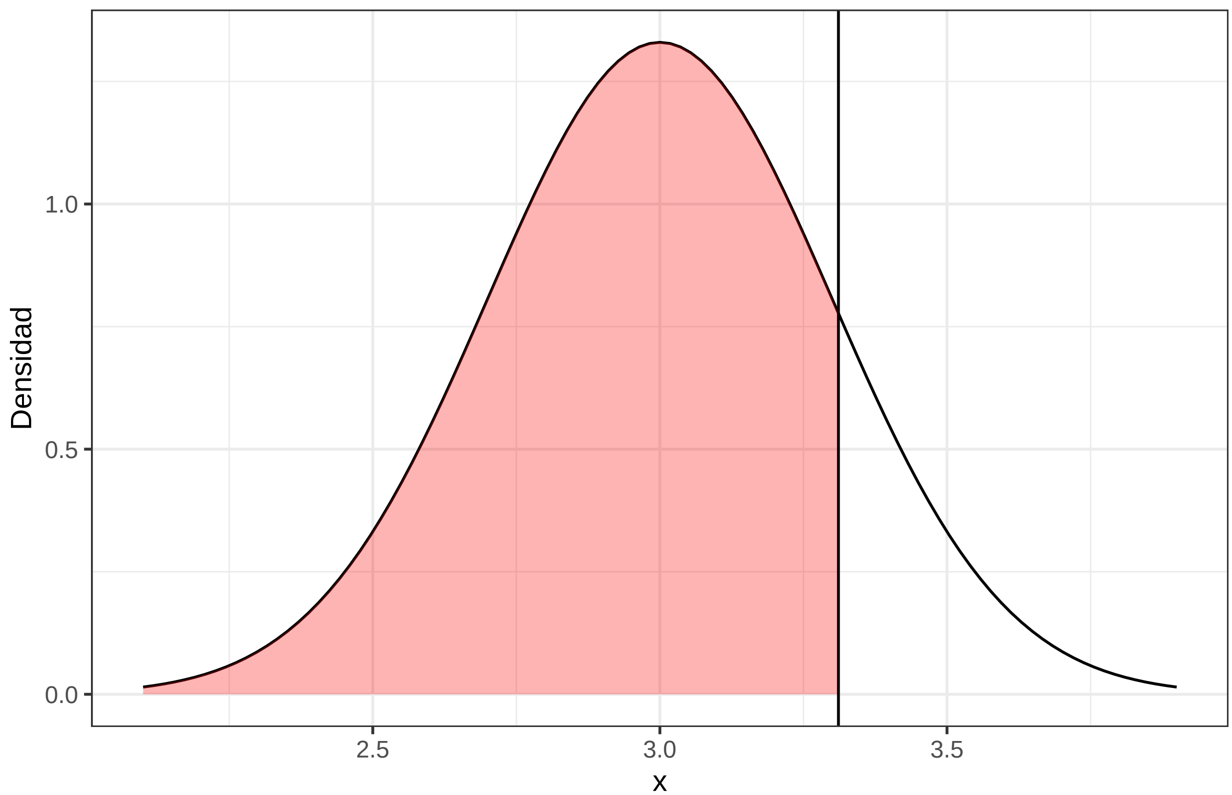
4. *El peso mínimo del 15% de los bebés varones que más pesan.* Esto es aquel valor que deje a su derecha el 15% de las observaciones, que a su vez, será también aquel que deja a su izquierda el 85% de las observaciones restantes.

```
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("Figura 4. Distribución Normal") + xlab("x") + theme_bw()
```

Figura 4. Distribución Normal



4.1.2 Distribución Binomial

Recuérdese que se define la variable aleatoria *Binomial* como el número de individuos que tiene una característica dicotómica en una muestra de individuos de un población muy grande, en teoría infinita, siendo constante para todos los individuos de la población la probabilidad p. En esa situación podemos decir que $X \sim B(n, p)$. Partiendo de aquí aprenderemos a calcular probabilidad, e inversos de las probabilidades, de una Binomial con **R**, destacando aquí cuatro funciones:

- **dbinom(x, size=n, prob=p)**. Devuelve el valor (valores) x de la *función 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 $p = 0.35$.

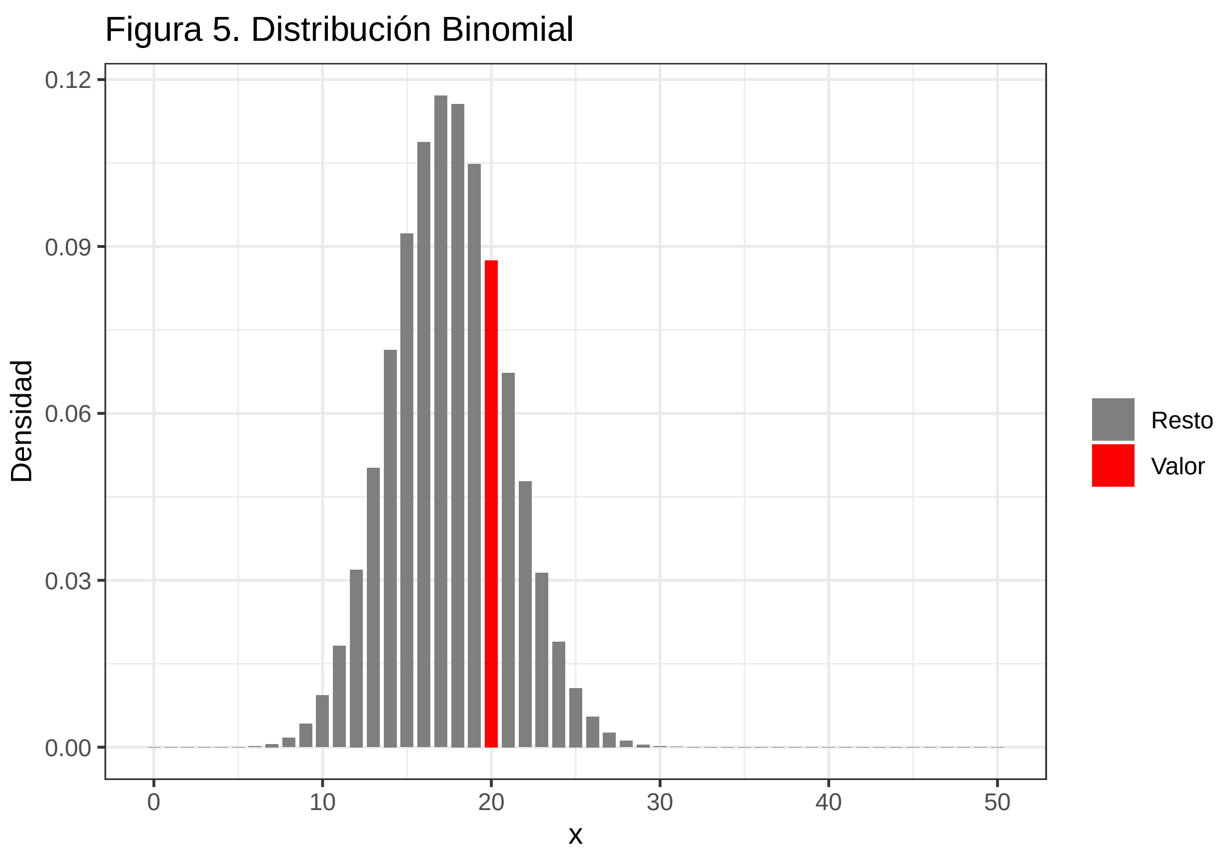
1. *Calcular la probabilidad de que exactamente 20 pacientes sean fumadores.* Aquí se pide calcular

$P[X = 20]$ 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("Figura 5. 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.

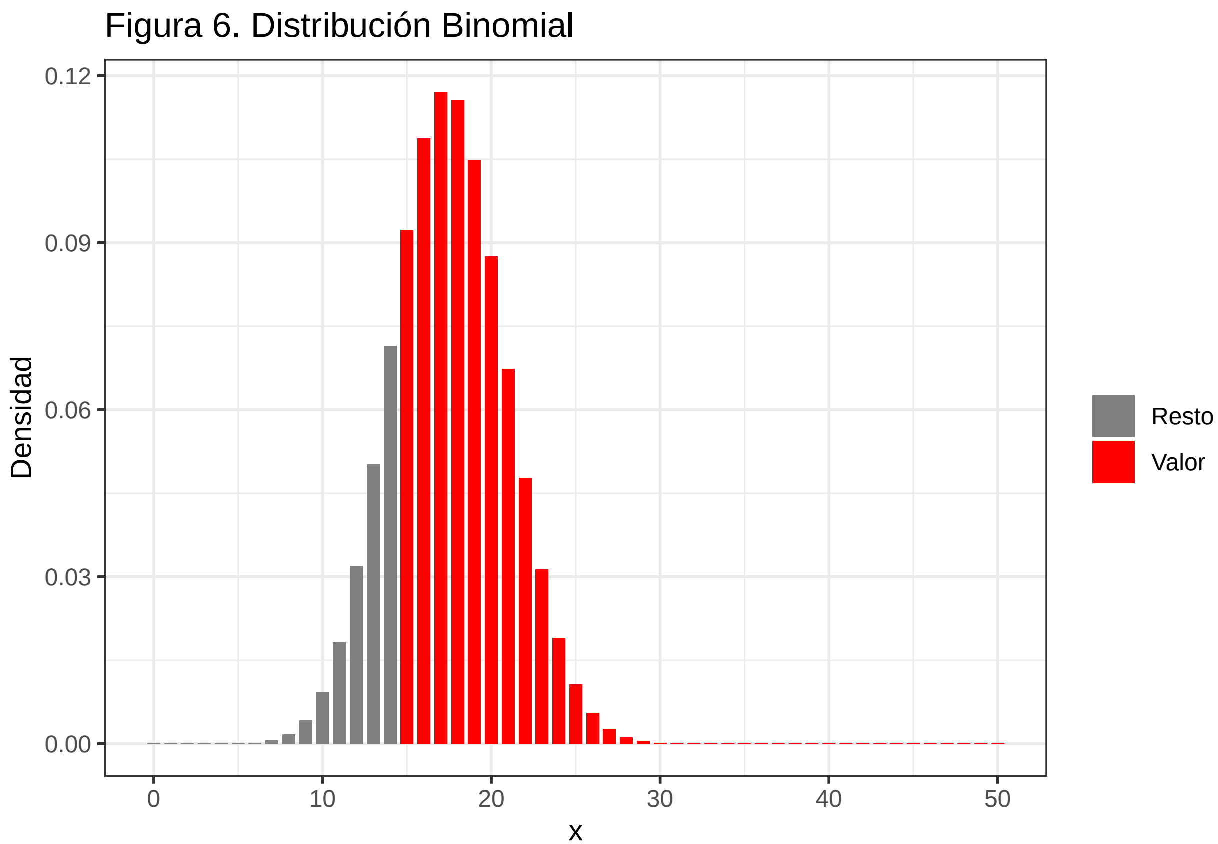
```
valor=14
pbinom(valor,size=n,prob=p,lower.tail=FALSE)
```

```
## [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("Figura 6. Distribución Binomial") +
  scale_fill_manual(name = "", values=c("grey50","red")) + theme_bw()
```

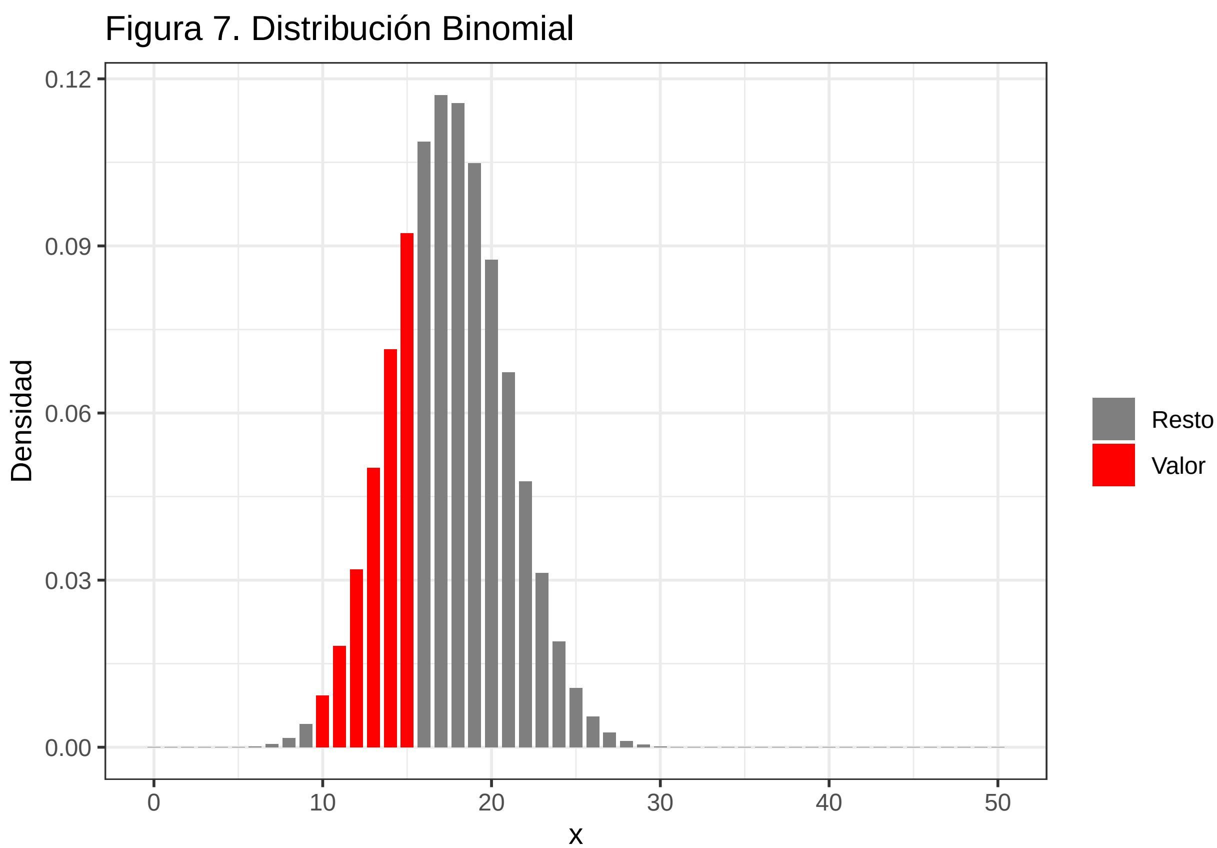


3. *Obtener la probabilidad de que entre 10 y 15 pacientes (ambos inclusive) sean fumadores.* Se pide calcular $P[10 \leq X \leq 15] = P[X \leq 15] - P[X < 10] = P[X \leq 15] - P[X \leq 9]$

```
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("Figura 7. Distribución Binomial") +
  scale_fill_manual(name = "", values=c("grey50","red")) + theme_bw()
```

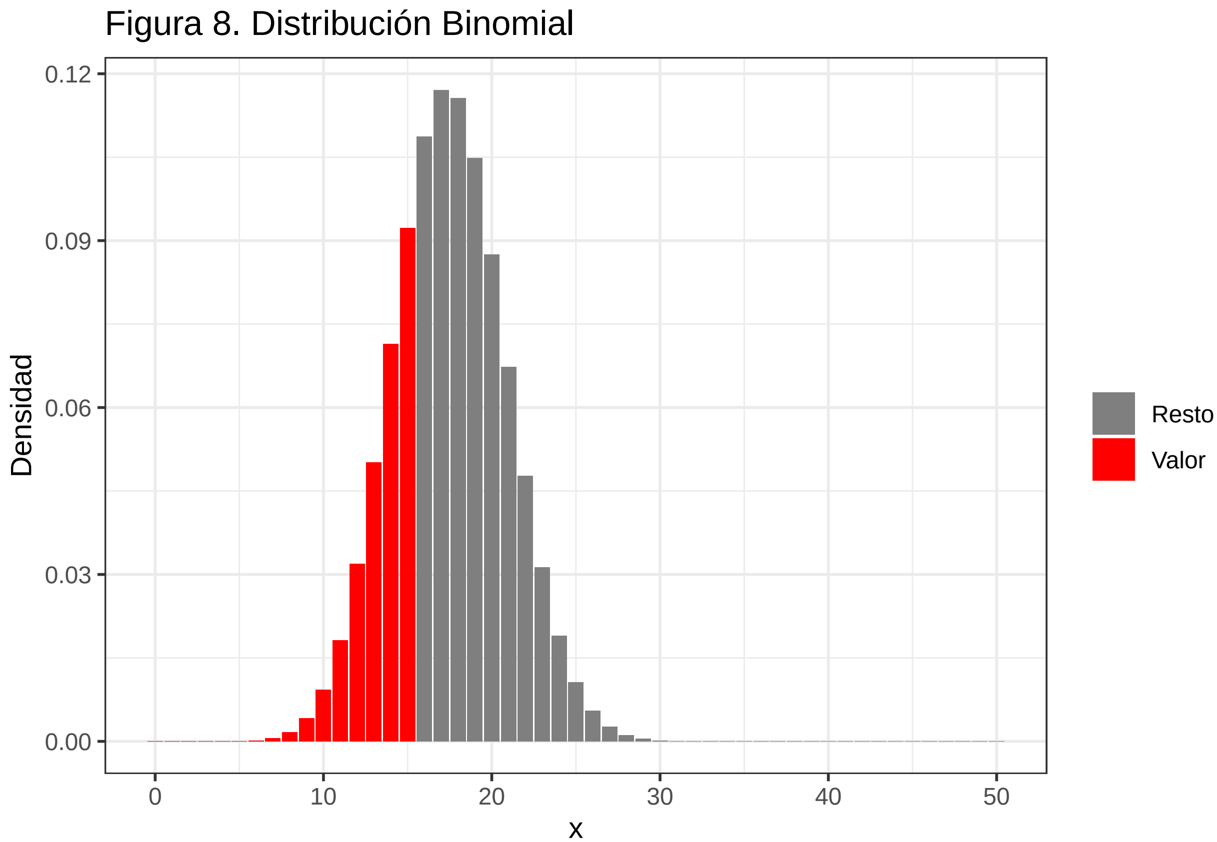


4. *Calcular el valor de la variable tal que deja a su derecha un 70% de las observaciones.* El valor de la variable que deja a su derecha un 70% de las observaciones es el mismo que deja a su izquierda el 30% restante, por lo que se pide calcular el valor q tal que $P[X \leq q] = 0.30$.

```
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("Figura 8. Distribución Binomial") +
  scale_fill_manual(name = "", values=c("grey50","red")) + theme_bw()
```



4.1.3 Distribución Poisson

Atendiendo a las definiciones que se han visto en las clases de Teoría de la variable aleatoria de Poisson, una variable aleatoria X sigue una distribución de Poisson de parámetro λ, abreviadamente $X \sim P(\lambda)$. Aprenderemos aquí a resolver problemas de tal variable con **R**, siendo las funciones aplicables en este caso:

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 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.

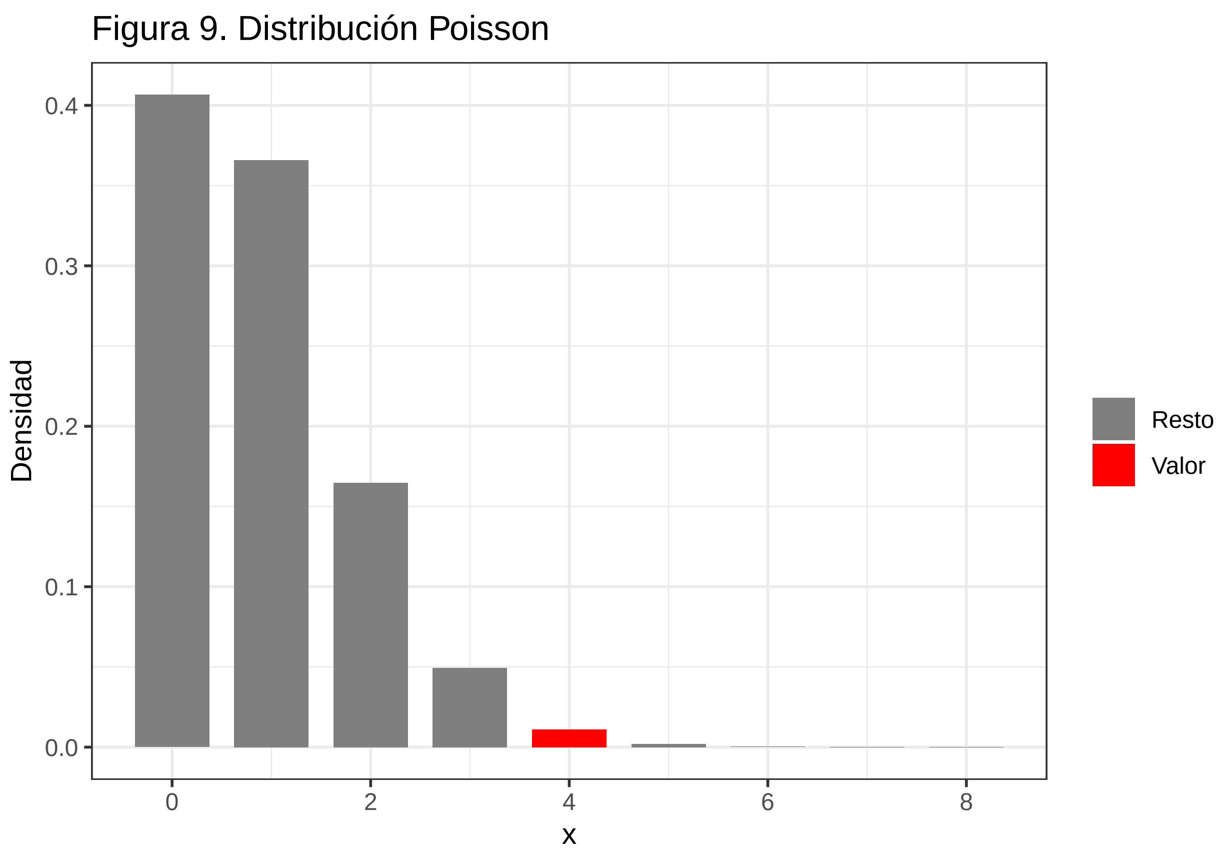
Para poner en práctica estas funciones se propone el siguiente ejercicio. *En un hospital el número medio de pancreatitis agudas atendidas por día es 0.90. Es fácil comprobar que la variable aleatoria que cuenta el número de pancreatitis agudas atendidas por día sigue una distribución Poisson de parámetro*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

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("Figura 9. Distribución Poisson") +
  scale_fill_manual(name = "", values=c("grey50","red")) + theme_bw()
```

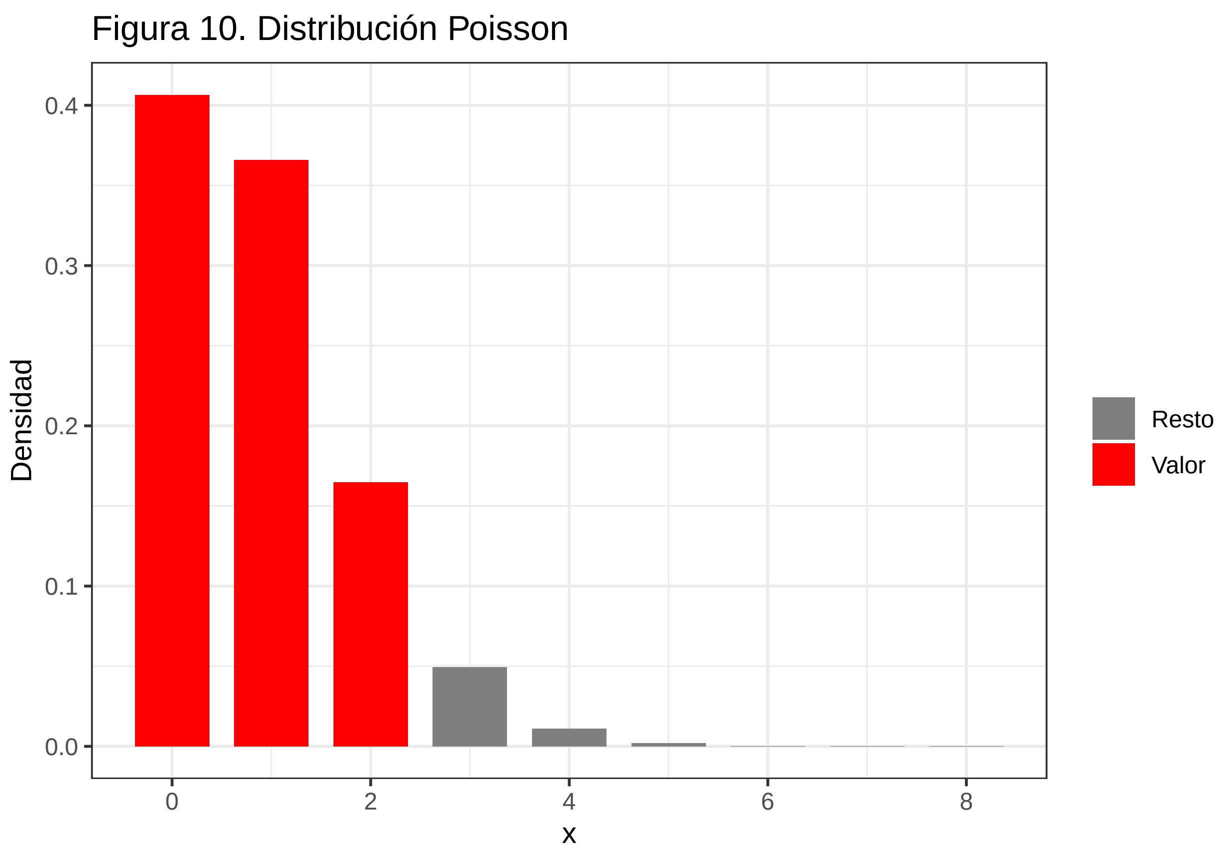


2. *Calcular la probabilidad de que, en un día cualquiera, el hospital atienda como máximo 2 clientes.* Se pide calcular $P[X \leq 2]$.

```
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("Figura 10. Distribución Poisson") +
  scale_fill_manual(name = "", values=c("grey50","red")) + theme_bw()
```

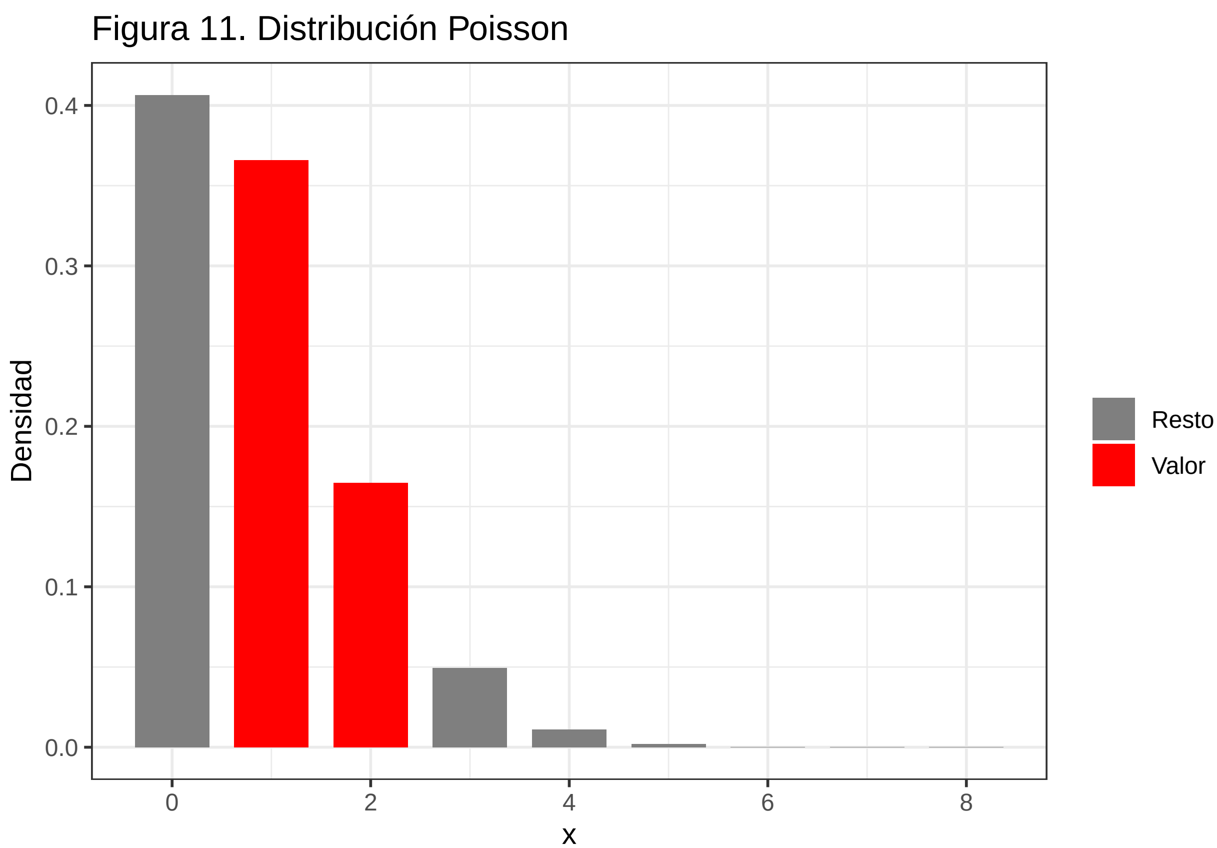


3. *Calcular la probabilidad de que, en un día dado, el hospital reciba entre 1 y 2 pacientes (ambos inclusive).* Se pide calcular $P[1 \leq X \leq 2] = P[X \leq 2] - P[X < 1] = P[X \leq 2] - P[X \leq 0]$.

```
valor1=2
valor2=0
ppois(valor1,lambda=lambda)-ppois(valor2,lambda=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("Figura 11. 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("Figura 12. Distribución Poisson") +
  scale_fill_manual(name = "", values=c("grey50","red")) + theme_bw()
```

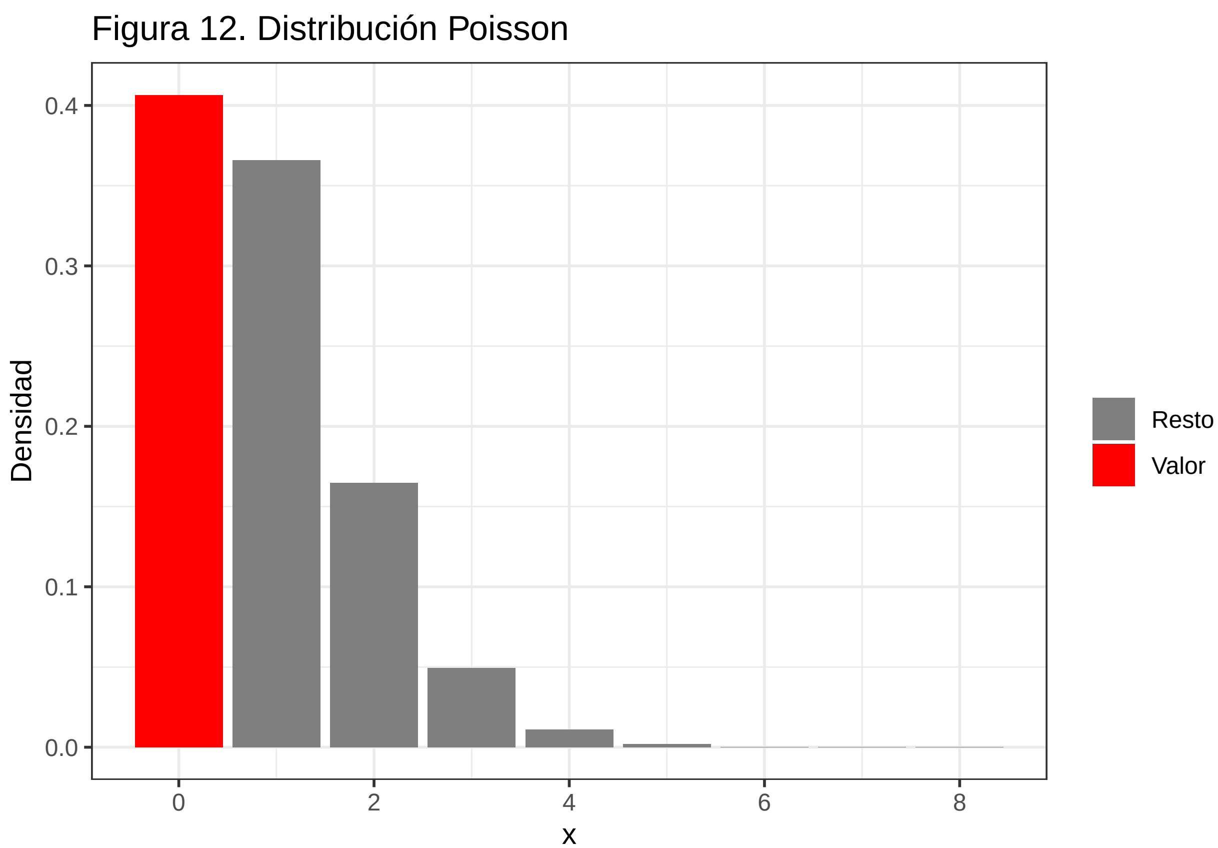


Figura 12. Distribución Poisson
Densidad
0.0
0.1
0.2
0.3
0.4
0
2
4
6
8
x
Resto
Valor