

Unidad 3: Variables aleatorias y sus distribuciones.

Modelo Normal

Script para obtener una probabilidad “menor que” numéricamente y gráficamente:

```
# =====
# PROBABILIDADES NORMALES EN R- Cálculo y representación gráfica.

Caso menor que

# =====
# Definimos los parámetros de la distribución normal
media <- 100 # Media ( $\mu$ )
desv <- 15 # Desviación estándar ( $\sigma$ )
valor <- 110

# Creamos una secuencia de valores para graficar la curva. La siguiente línea crea un vector de 200 valores equidistantes que van desde media - 4*desv hasta media + 4*desv
# Es decir que la secuencia de valores va desde: media - 4*desv hasta media + 4*desv
# Esta secuencia cubre prácticamente toda la distribución (el 99.99% de los valores)
# El valor length = 200 permite obtener una curva que gráficamente tenga continuidad, mientras que un valor length = 10 daría por resultado una curva con aspecto dentado.

x <- seq(media - 4*desv, media + 4*desv, length = 200)
y <- dnorm(x, mean = media, sd = desv)
# -----
# Probabilidad menor que
# -----
# Ejemplo: P(X < valor)
# Cálculo numérico: El valor de la probabilidad queda guardado en la variable P_menor
P_menor <- pnorm(valor, mean = media, sd = desv)
cat('Probabilidad P(X < valor) =', P_menor, '\n')
# Gráfico con el área sombreada a la izquierda
# type = 'l' define que el gráfico será una línea continua
# lwd = 2 define el grosor de la línea
```

```
plot(x, y, type = 'l', lwd = 2, col = 'black', main = paste0('Probabilidad: P(X < ', valor, ') = ', round(P_menor, 4)), xlab = 'X', ylab = 'Y')

# El valor length = 100 es suficiente para lograr un sombreado adecuado.

x_sombreado <- seq(media - 4*desv, valor, length = 100)

y_sombreado <- dnorm(x_sombreado, mean = media, sd = desv)

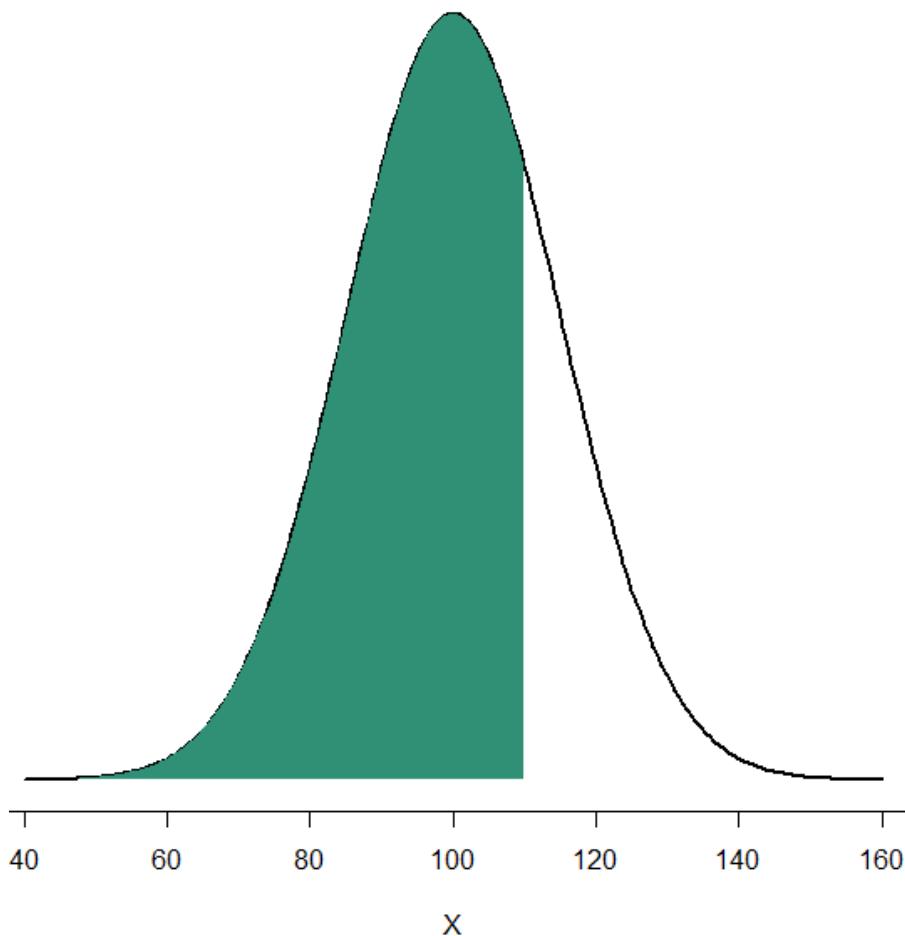
# A continuación se definen las coordenadas x del gráfico:

# Para una probabilidad menor que los valores de x van desde media - 4*desv, continúan con los valores intermedios x sombreado y finalizan en valor

polygon(c(media - 4*desv, x_sombreado, valor), c(0, y_sombreado, 0), col = '#309076', border = NA)
```

Salida obtenida:

Probabilidad: $P(X < 110) = 0.7475$



Script para calcular una probabilidad “**mayor que**” numéricamente y gráficamente:

```
# =====
# PROBABILIDADES NORMALES EN R- Cálculo y representación gráfica.

Caso mayor que

# =====

# Definimos los parámetros de la distribución normal

media <- 100    # Media ( $\mu$ )
desv <- 15      # Desviación estándar ( $\sigma$ )
valor <- 110

# Creamos una secuencia de valores para graficar la curva. La siguiente línea crea un vector de
# 200 valores equidistantes que van desde media - 4*desv hasta media + 4*desv
# Es decir que la secuencia de valores va desde: media - 4*desv hasta media + 4*desv
# Esta secuencia cubre prácticamente toda la distribución (el 99.99% de los valores)
# El valor length = 200 permite obtener una curva que gráficamente tenga continuidad,
# mientras que un valor length = 10 daría por resultado una curva con aspecto dentado.

x <- seq(media - 4*desv, media + 4*desv, length = 200)
y <- dnorm(x, mean = media, sd = desv)
#
# Probabilidad mayor que
#
# Ejemplo: P(X > valor)
# Cálculo numérico:
P_mayor <- pnorm(valor, mean = media, sd = desv, lower.tail = F)
cat('Probabilidad P(X > valor) =', P_mayor, '\n')
# Gráfico con el área sombreada a la derecha
# type = 'l' define que el gráfico será una línea continua
# lwd = 2 define el grosor de la línea
plot(x, y, type = 'l', lwd = 2, col = 'black', main = paste0('Probabilidad: P(X > ', valor, ') = ',
round(P_mayor, 4)),
xlab = 'X', ylab = 'Y')
```

Para una probabilidad **mayor que**, los valores de x van desde **valor** hasta **media + 4*desv**

El valor **length = 100** es suficiente para lograr un sombreado adecuado.

```
x_sombreado <- seq(valor, media + 4*desv, length = 100)
```

```
y_sombreado <- dnorm(x_sombreado, mean = media, sd = desv)
```

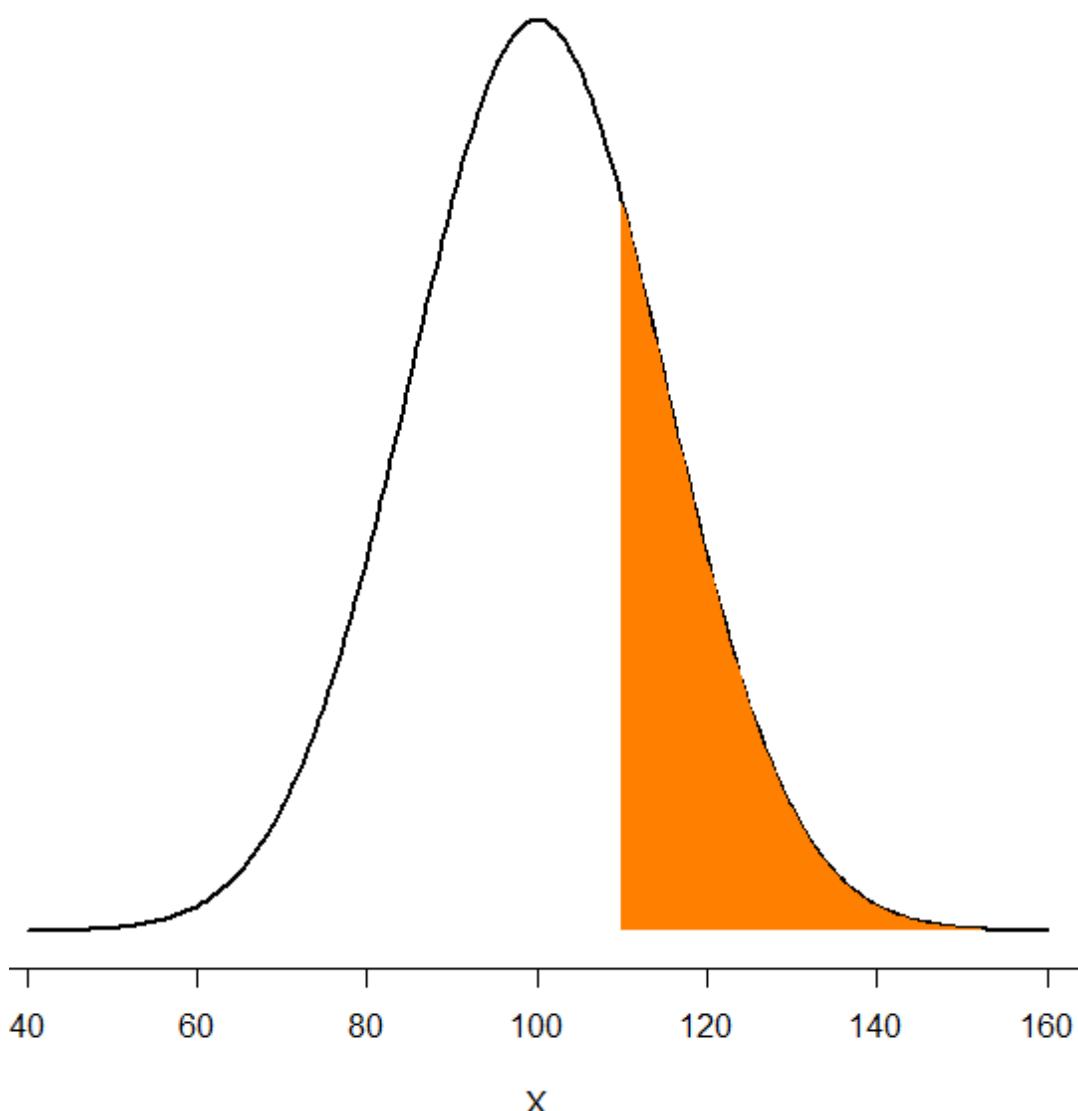
A continuación se definen las coordenadas x del polígono:

Para una probabilidad **mayor que** los valores de x parten desde **valor**, continúan con los valores intermedios x sombreado y finalizan en **media + 4*desv**

```
polygon(c(valor, x_sombreado, media + 4*desv), c(0, y_sombreado, 0), col = '#FF7F00', border = NA)
```

Salida obtenida:

Probabilidad: $P(X > 110) = 0.2525$



Script para obtener una probabilidad “entre” numéricamente y gráficamente

```
# =====
# PROBABILIDADES NORMALES EN R- Cálculo y representación gráfica.

# Caso entre: P(valor1 < x < valor2)
# =====

# Definimos los parámetros de la distribución normal

media <- 100    # Media ( $\mu$ )
desv <- 15      # Desviación estándar ( $\sigma$ )
```

```
valor1 <- 80
```

```
valor2 <- 120
```

```
# Creamos una secuencia de valores para graficar la curva. La siguiente línea crea un vector de  
200 valores equidistantes que van desde media - 4*desv hasta media + 4*desv
```

```
# Es decir que la secuencia de valores va desde: media - 4*desv hasta media + 4*desv
```

```
# Esta secuencia cubre prácticamente toda la distribución (el 99.99% de los valores)
```

```
# El valor length = 200 permite obtener una curva que gráficamente tenga continuidad,  
mientras que un valor length = 10 daría por resultado una curva con aspecto dentado.
```

```
x <- seq(media - 4*desv, media + 4*desv, length = 200)
```

```
y <- dnorm(x, mean = media, sd = desv)
```

```
# -----
```

```
# Probabilidad entre dos valores
```

```
# -----
```

```
# Ejemplo: P(valor1 < X < valor2)
```

```
# Cálculo numérico:
```

```
P_entre <- pnorm(valor2, mean = media, sd = desv) - pnorm(valor1, mean = media, sd = desv)
```

```
cat('Probabilidad P(valor1 < X < valor2) =', P_entre, '\n')
```

```
# Gráfico con el área central sombreada
```

```
# type = 'l' define que el gráfico será una línea llena, por defecto será una línea de puntos
```

```
# lwd = 2 define el grosor de la línea
```

```
plot(x, y, type = 'l', lwd = 2, col = 'black',
```

```
main = paste0('Probabilidad: P(', valor1, '< X <', valor2, ') = ', round(P_entre, 4)),
```

```
  xlab = 'X', ylab = 'Y')
```

```
# Para una probabilidad entre, los valores de x van desde valor1 hasta valor2
```

```
# El valor length = 100 es suficiente para lograr un sombreado adecuado.
```

```
x_sombreado <- seq(valor1, valor2, length = 100)
```

```
y_sombreado <- dnorm(x_sombreado, mean = media, sd = desv)
```

```
# A continuación se definen las coordenadas x del gráfico:
```

Para una probabilidad **entre**, los valores de x van desde **valor1**, continúan con los valores intermedios x sombreado y finalizan en **valor2** porque debe calcularse **P(valor1 < X < valor2)**

```
polygon(c(valor1, x_sombreado, valor2), c(0, y_sombreado, 0), col = '#b8860b', border = NA)
```

Salida obtenida:

Probabilidad: $P(80 < X < 120) = 0.8176$

