

# Graficas distribuciones normales

Julio Ladron de Guevara

```
options(repos = c(CRAN = "https://cran.rstudio.com"))
```

```
install.packages("carData")
```

```
## package 'carData' successfully unpacked and MD5 sums checked
```

```
##
```

```
## The downloaded binary packages are in
```

```
## C:\Users\julio\AppData\Local\Temp\RtmpySpwcf\downloaded_packages
```

```
install.packages("prettyR")
```

```
## package 'prettyR' successfully unpacked and MD5 sums checked
```

```
##
```

```
## The downloaded binary packages are in
```

```
## C:\Users\julio\AppData\Local\Temp\RtmpySpwcf\downloaded_packages
```

```
install.packages('latexpdf', repos= "http://cran.us.r-project.org")
```

```
## package 'latexpdf' successfully unpacked and MD5 sums checked
```

```
##
```

```
## The downloaded binary packages are in
```

```
## C:\Users\julio\AppData\Local\Temp\RtmpySpwcf\downloaded_packages
```

```
install.packages('tinytex', repos= "http://cran.us.r-project.org")
```

```
## package 'tinytex' successfully unpacked and MD5 sums checked
```

```
##
```

```
## The downloaded binary packages are in
```

```
## C:\Users\julio\AppData\Local\Temp\RtmpySpwcf\downloaded_packages
```

## Funcion de densidad

Dada una variable aleatoria  $X \sim N(\mu, \sigma^2)$ , crearemos una funcion de dos variables que devuelva su funcion de densidad.

En este caso, `qnorm(0.999)` nos devuelve el cuantil correspondiente al percentil 99.9 de la distribución normal estándar, que representa un valor muy alejado de la media y, por lo tanto, proporciona un rango amplio de valores para los límites del eje x.

```
graf_densidadN <- function(a,b){
  q <- qnorm(0.999)
  xlim_left <- a - q*b
  xlim_right <- a + q*b
  curve(dnorm(x,a,b),
        xlim = c(xlim_left, xlim_right),
        col = "blue",
        lwd = 2,
        xlab = "x",
        ylab = "f(x)",
        main = paste("Funcion de densidad N(",a,",",b,")"))
}
```

## Funcion de distribución

Dada una variable aleatoria  $X \sim N(\mu, \sigma^2)$ , crearemos una funcion de dos variables que devuelva su funcion de distribucion.

En este caso, `qnorm(0.999)` nos devuelve el cuantil correspondiente al percentil 99.9 de la distribución normal estándar, que representa un valor muy alejado de la media y, por lo tanto, proporciona un rango amplio de valores para los límites del eje x

```
graf_distribucionN <- function(a, b) {
  q <- qnorm(0.999)
  xlim_left <- a - q * b
  xlim_right <- a + q * b
  curve(pnorm(x, a, b),
        xlim = c(xlim_left, xlim_right),
        ylim = c(0, 1),
        col = "blue",
        lwd = 2,
        xlab = "x",
        ylab = "F(x)",
        main = paste("Funcion de distribucion N(", a, ",", b, ")"))
}
```

## Area bajo la curva

Dada una variable aleatoria  $X \sim N(\mu, \sigma^2)$  y dados dos puntos  $x, y \in \mathbb{R}$  tales que  $x \neq y$  crearemos una funcion de cuatro variables que devuelva el area bajo la funcion de densidad entre esas dos variables.

```
areadeN_entre <- function(a,b,c,d){
  z = max(c,d)
  w = min(c,d)

  region = seq(w,z,0.01)

  xP = c(w,region,z)

  yP = c(0,dnorm(region, a, b), 0)
```

```

y_max = max(dnorm(region, a, b))

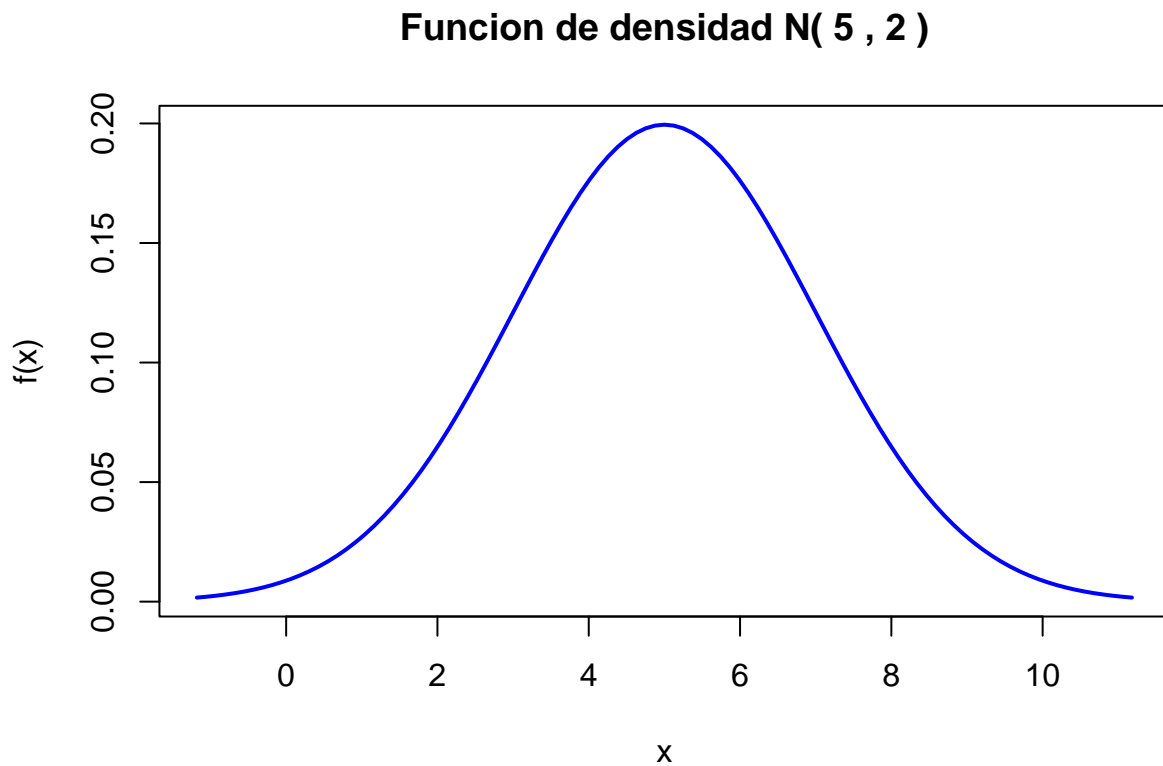
curve(dnorm(x,a,b),
      xlim = c(w - abs(w)*0.1, z + abs(z)*0.1),
      yaxs = "i",
      ylim = c(0, y_max + y_max * 0.1),
      col = "blue",
      lwd = 2,
      xlab = "f(x)",
      main = paste("Area bajo N(",a,",",b,"")"))
polygon(xP, yP, col = "orange")
}

```

```

# Función normal 1
mu1 <- 5
sigma1 <- 2
graf_densidadN(mu1, sigma1)

```

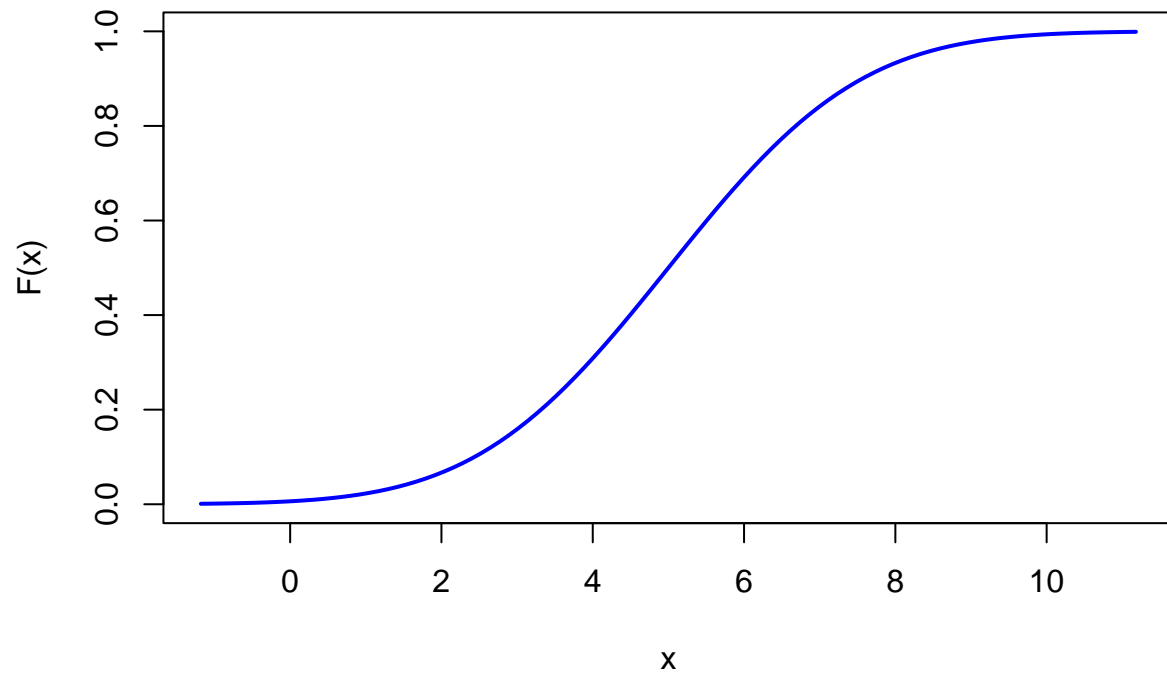


```

graf_distribucionN(mu1, sigma1)

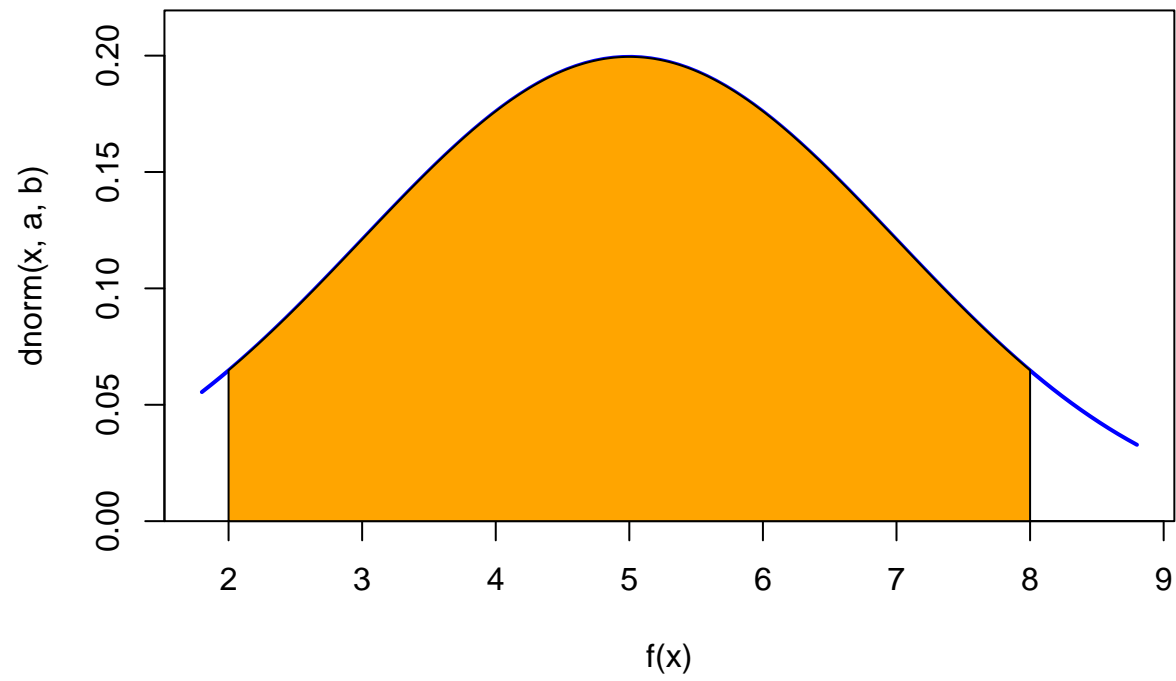
```

### Funcion de distribucion N( 5 , 2 )



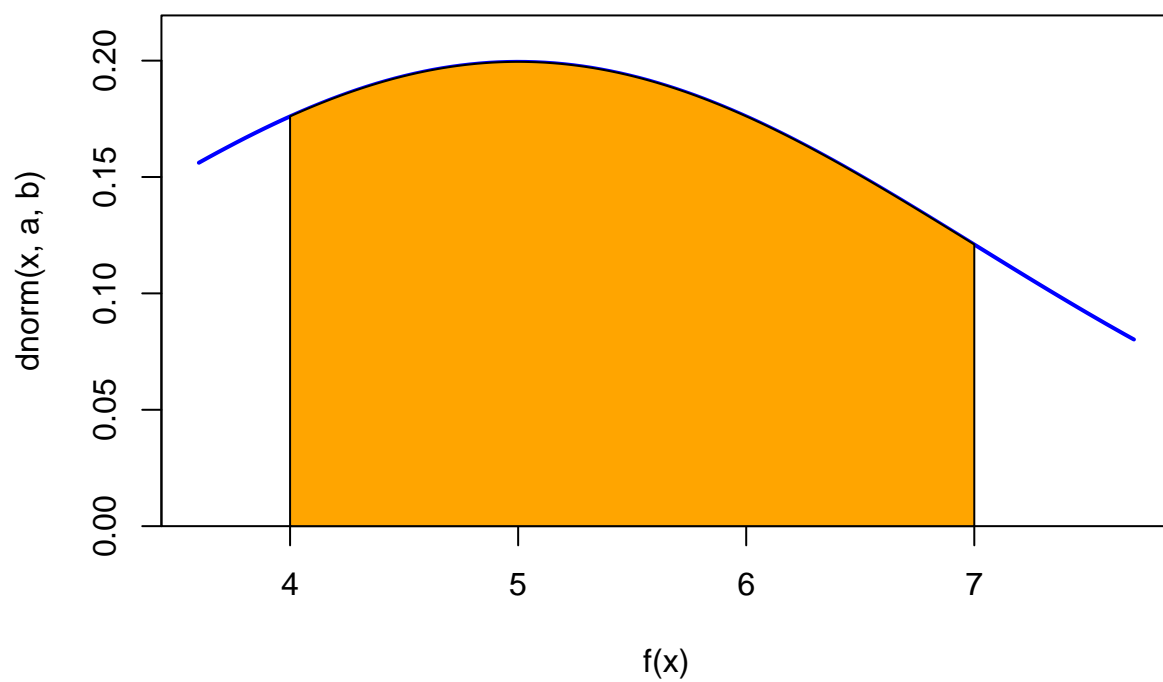
```
areadeN_entre(mu1, sigma1, 2, 8)
```

### Area bajo N( 5 , 2 )



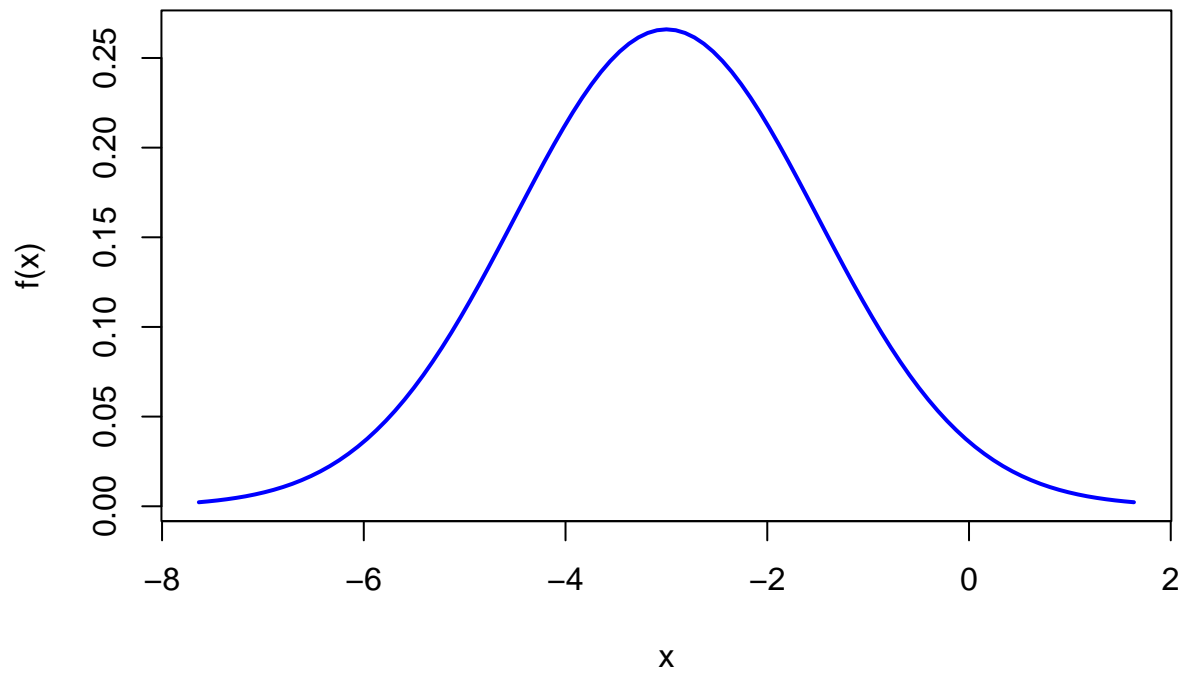
```
areadeN_entre(mu1, sigma1, 4, 7)
```

### Area bajo N( 5 , 2 )



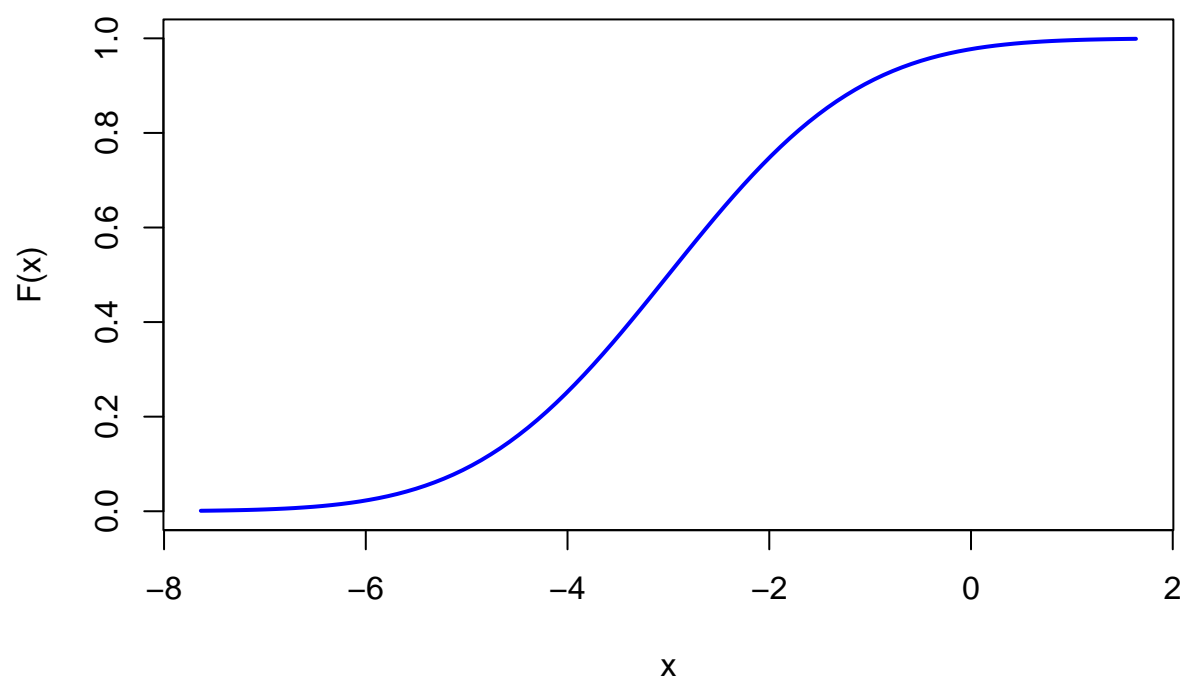
```
# Función normal 2  
mu2 <- -3  
sigma2 <- 1.5  
graf_densidadN(mu2, sigma2)
```

### Funcion de densidad N( -3 , 1.5 )



```
graf_distribucionN(mu2, sigma2)
```

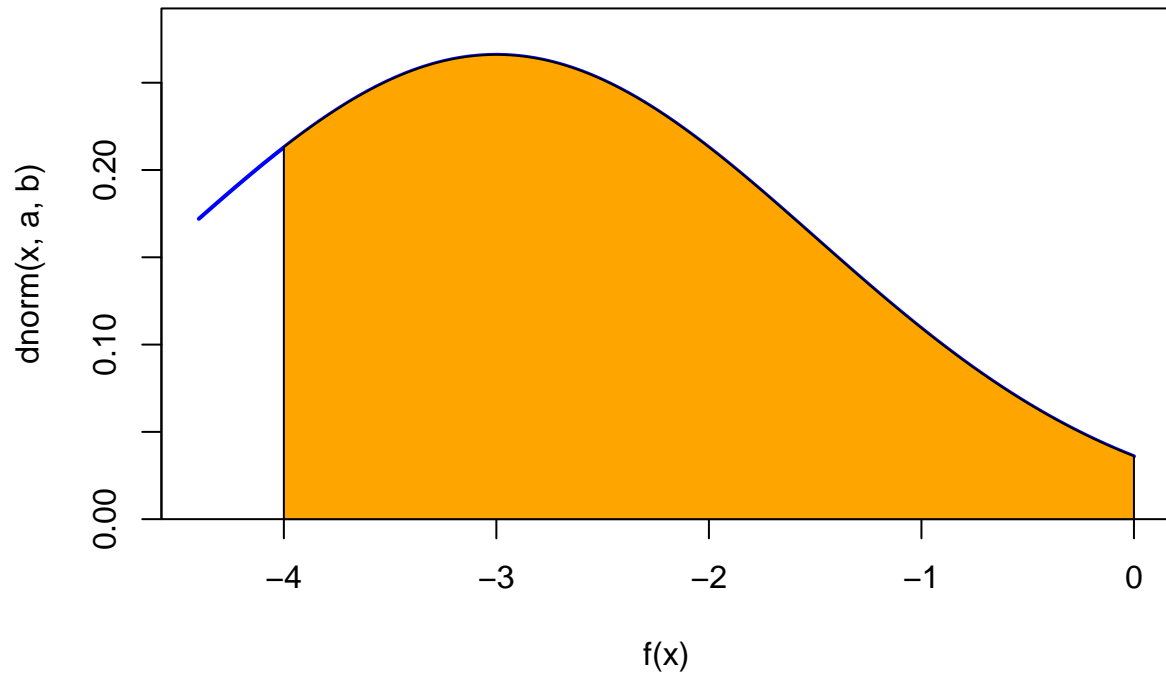
### Funcion de distribucion $N(-3, 1.5)$



```
areadeN_entre(mu2, sigma2, -4, 0)
```

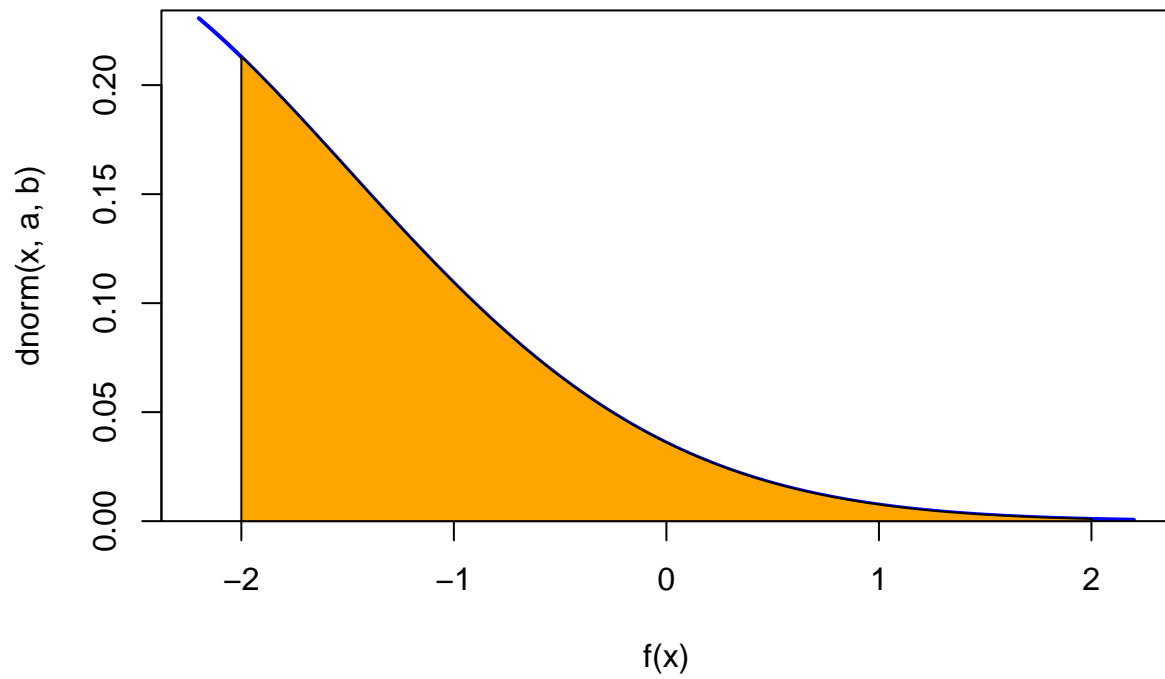


### Area bajo $N(-3, 1.5)$



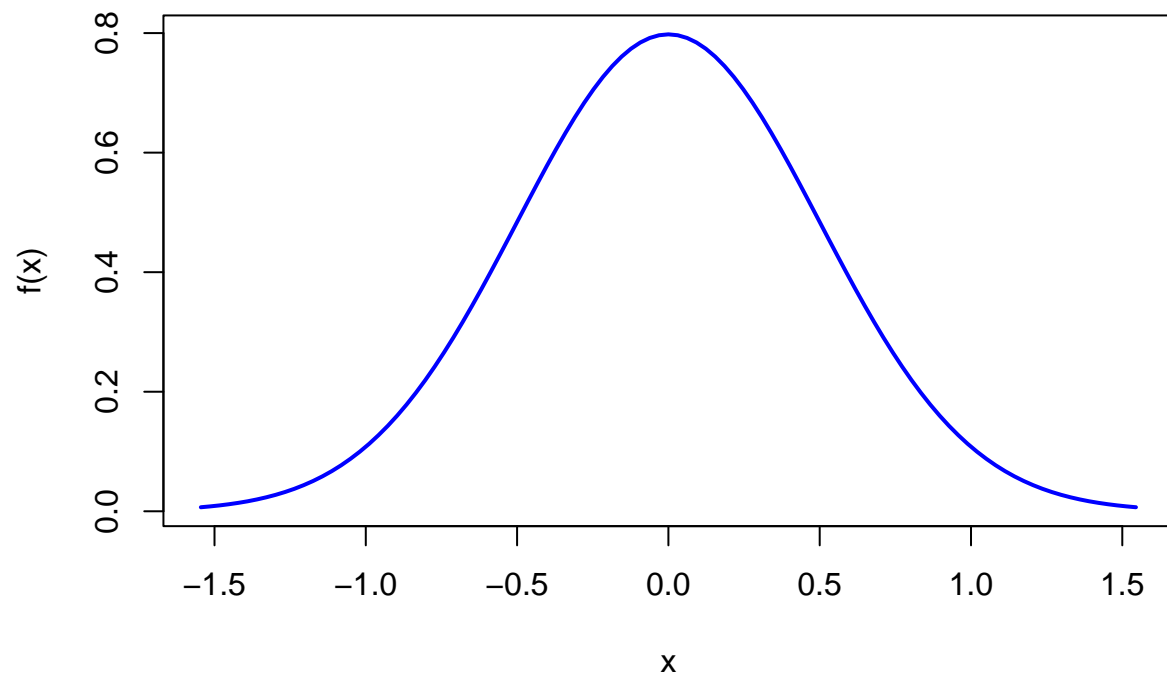
```
areadeN_entre(mu2, sigma2, -2, 2)
```

### Area bajo $N(-3, 1.5)$



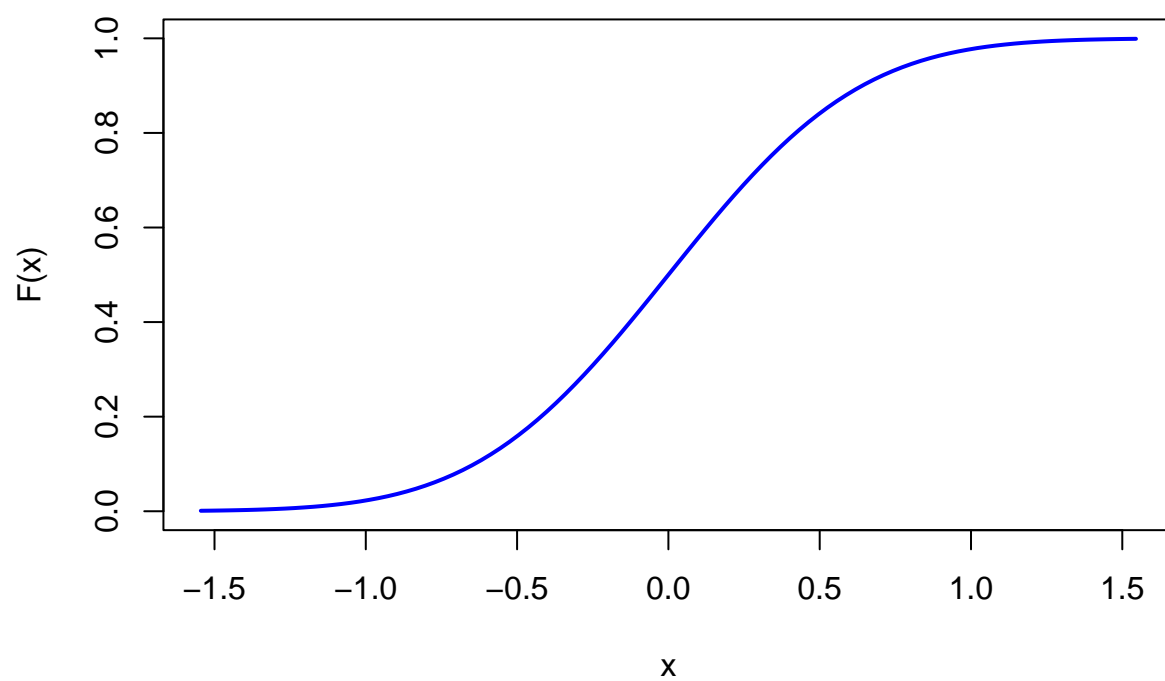
```
# Función normal 3  
mu3 <- 0  
sigma3 <- 0.5  
graf_densidadN(mu3, sigma3)
```

### Funcion de densidad N( 0 , 0.5 )



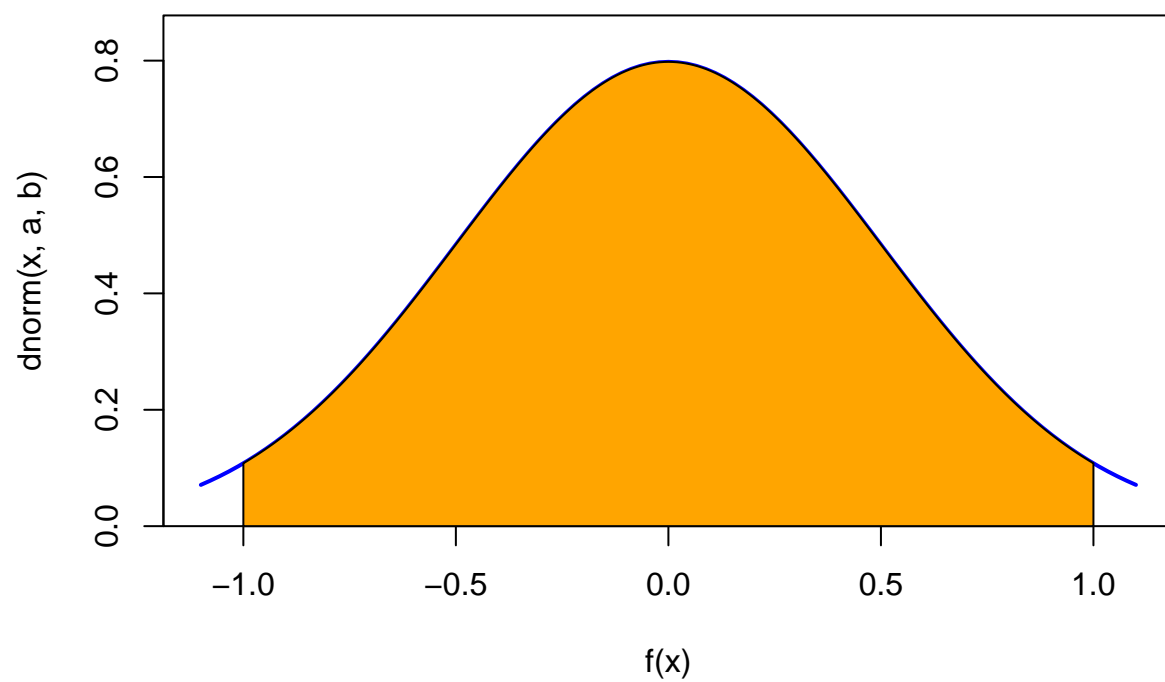
```
graf_distribucionN(mu3, sigma3)
```

### Funcion de distribucion N( 0 , 0.5 )



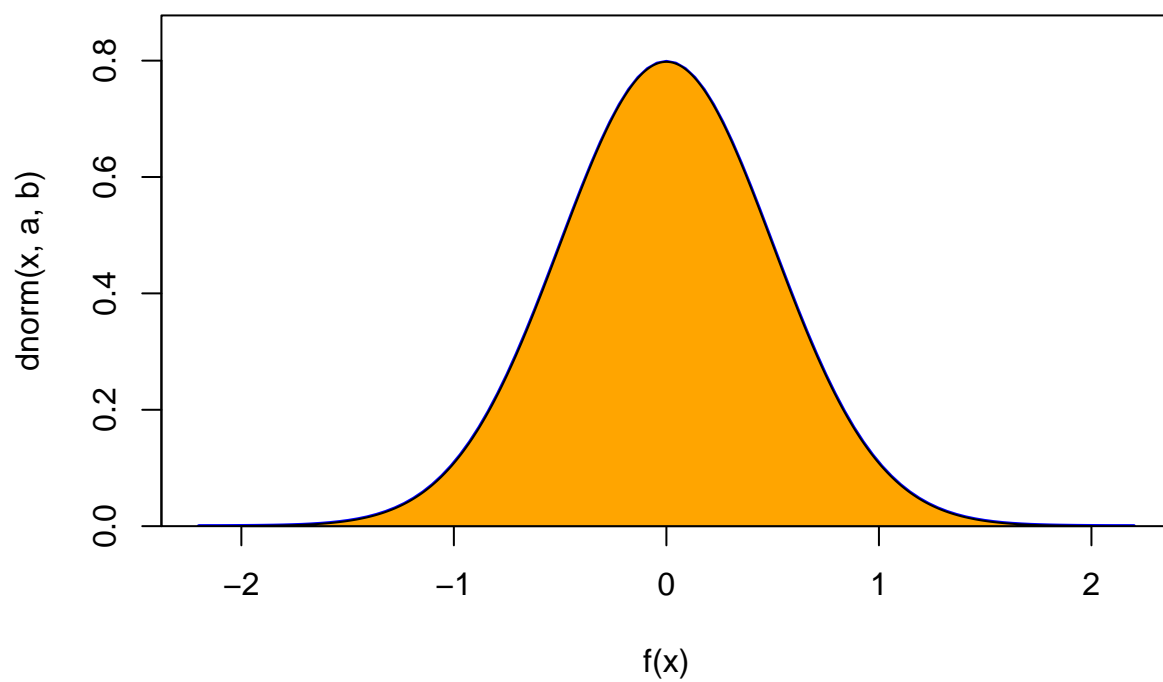
```
areadeN_entre(mu3, sigma3, -1, 1)
```

### Area bajo $N(0, 0.5)$



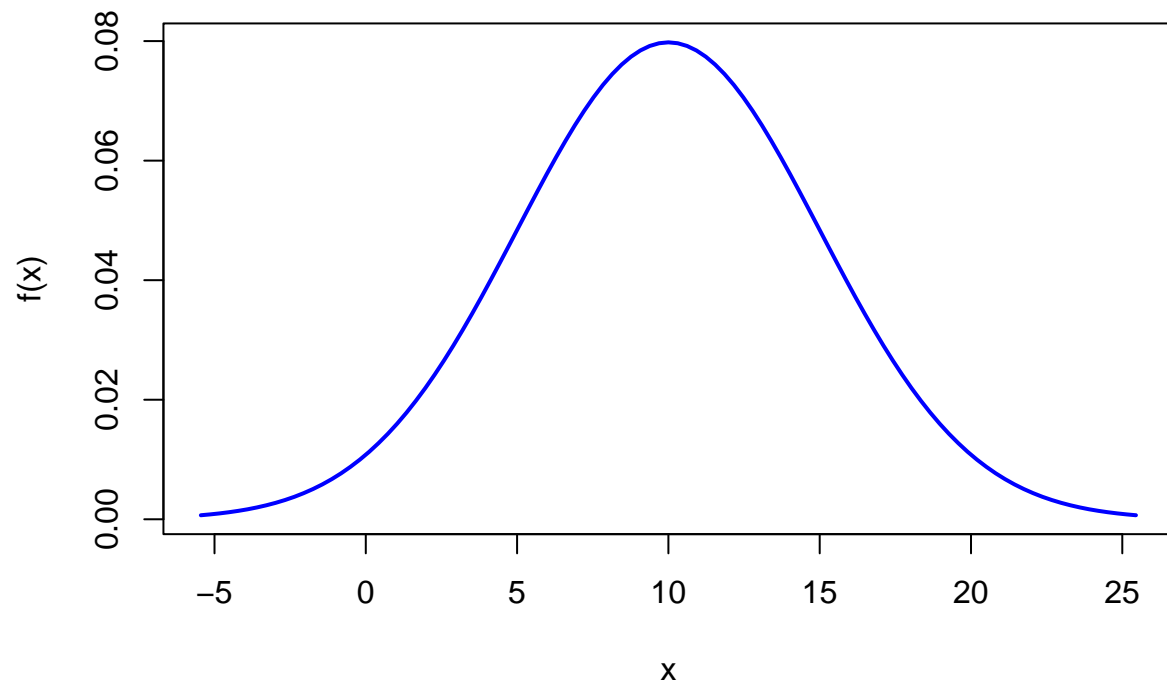
```
areadeN_entre(mu3, sigma3, -2, 2)
```

### Area bajo N( 0 , 0.5 )



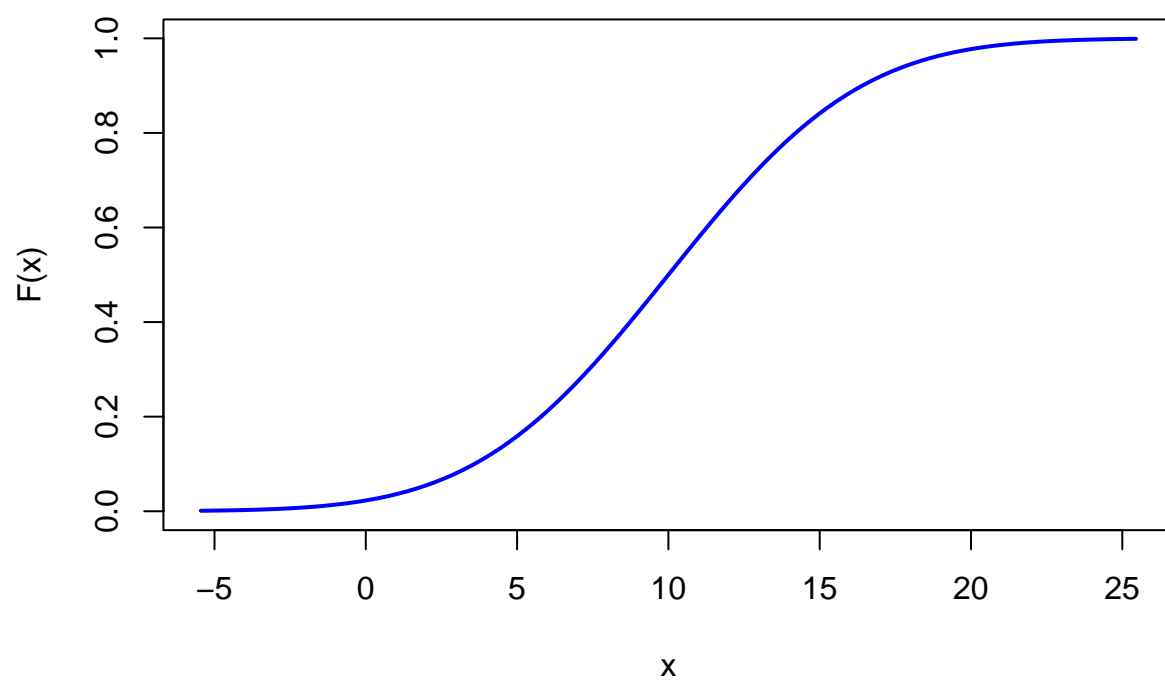
```
# Función normal 4  
mu4 <- 10  
sigma4 <- 5  
graf_densidadN(mu4, sigma4)
```

### Funcion de densidad N( 10 , 5 )



```
graf_distribucionN(mu4, sigma4)
```

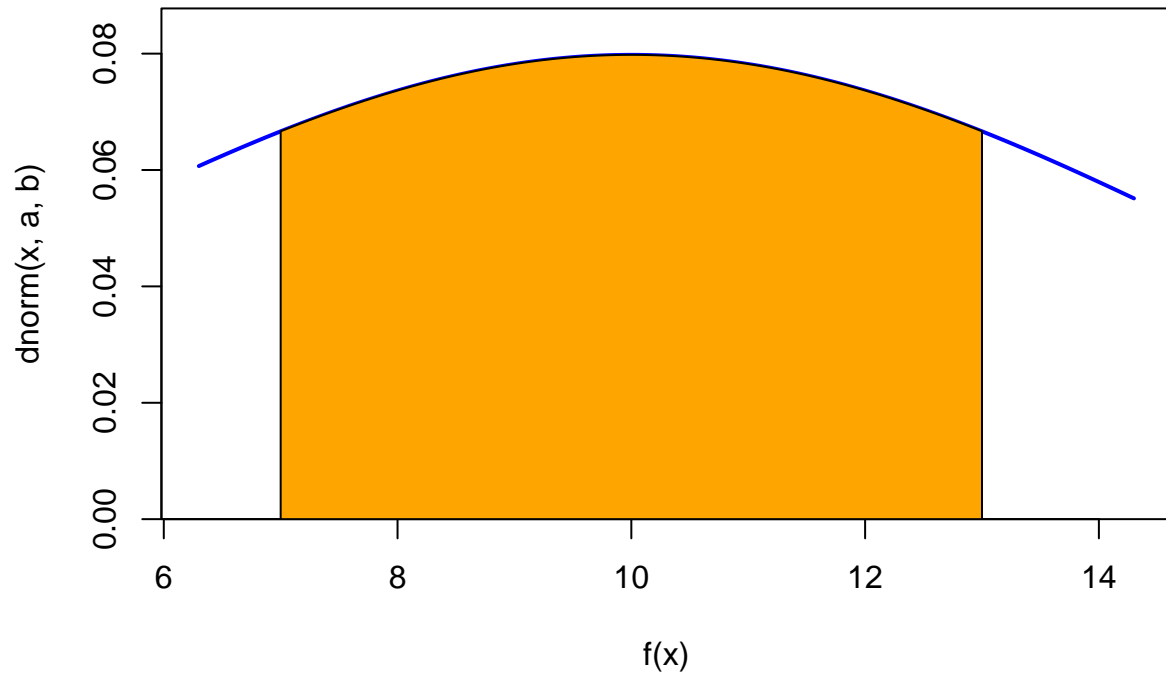
### Funcion de distribucion N( 10 , 5 )



```
areadeN_entre(mu4, sigma4, 7, 13)
```

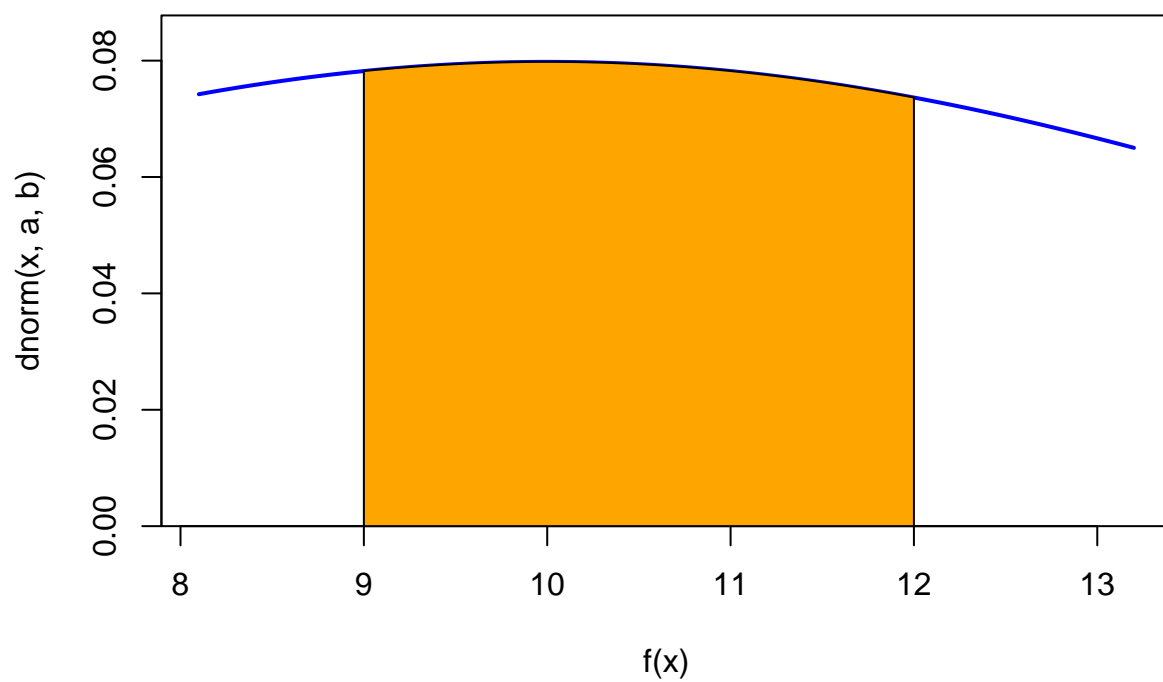


### Area bajo $N(10, 5)$



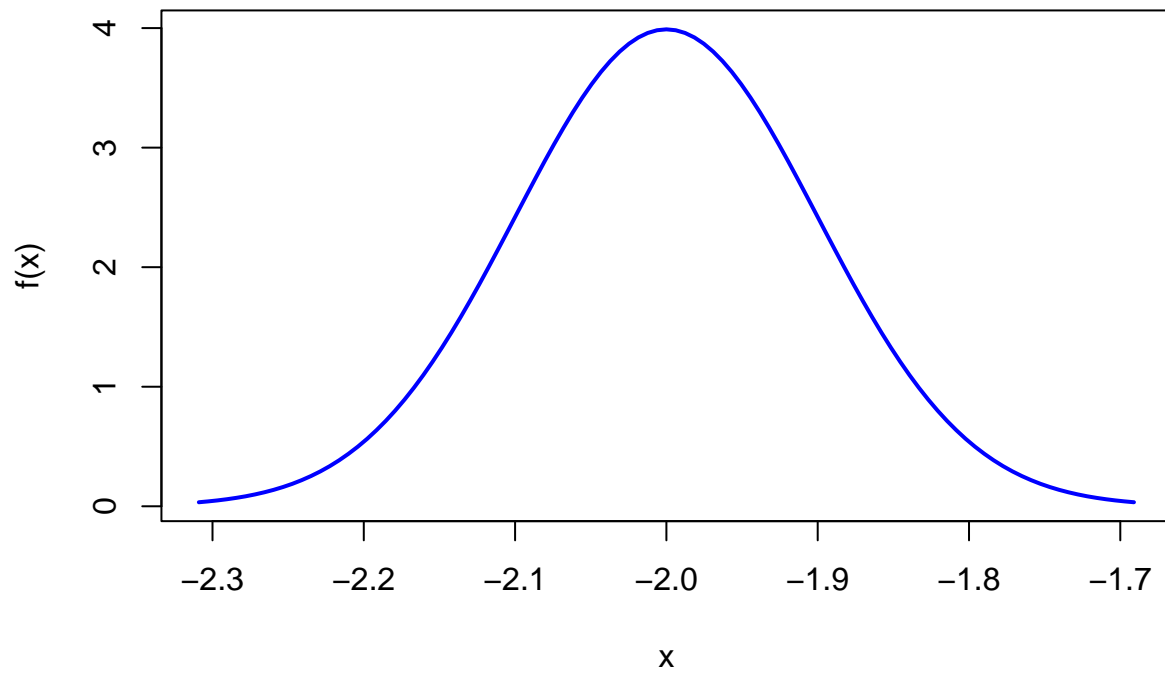
```
areadeN_entre(mu4, sigma4, 9, 12)
```

### Area bajo N( 10 , 5 )



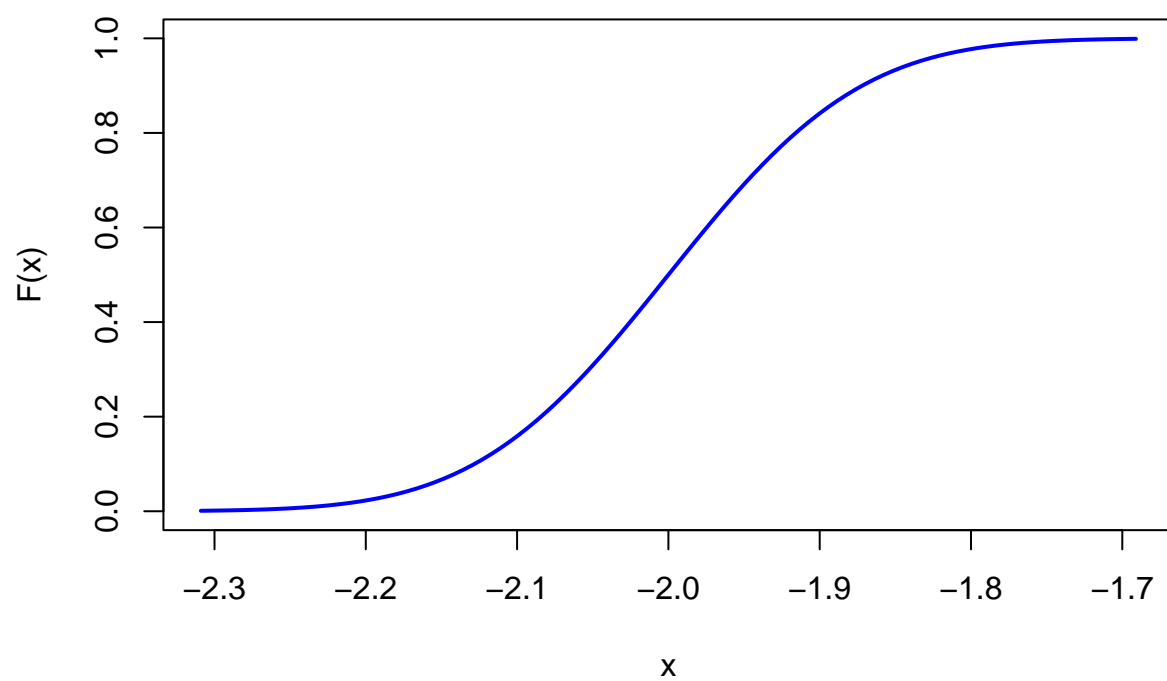
```
# Función normal 5  
mu5 <- -2  
sigma5 <- 0.1  
graf_densidadN(mu5, sigma5)
```

### Funcion de densidad N( -2 , 0.1 )



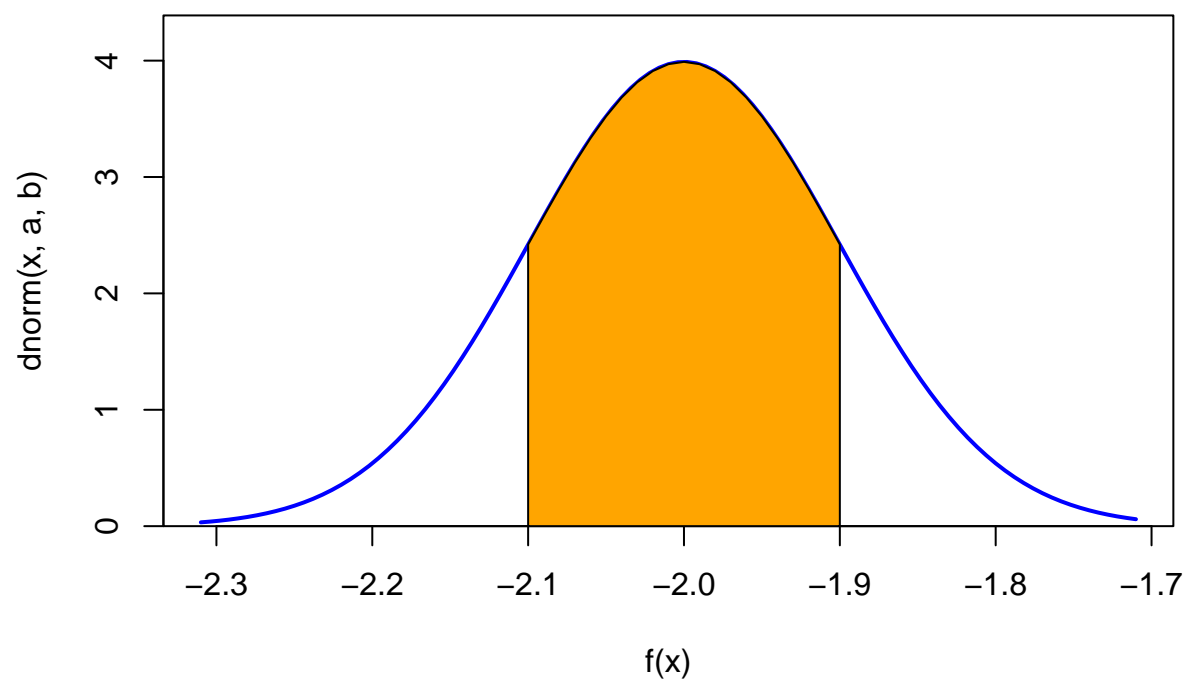
```
graf_distribucionN(mu5, sigma5)
```

### Funcion de distribucion N( -2 , 0.1 )



```
areadeN_entre(mu5, sigma5, -2.1, -1.9)
```

### Area bajo $N(-2, 0.1)$



```
areadeN_entre(mu5, sigma5, -2.2, -1.9)
```

Area bajo  $N(-2, 0.1)$

