

# Falsa Posición

Uriel Paluch

6/9/2021

## Método de la falsa posición

No es un método que el libro recomiende.

```
FalsaPosicion <- function(p0, p1, tol, n = 100){  
  
  q0 <- f(p0)  
  q1 <- f(p1)  
  
  for (i in 2:n){  
    p = p1 - q1*(p1-p0)/(q1-q0)  
  
    if (abs(p-p1) < tol){  
      return(p)  
    }  
  
    q = f(p)  
  
    if (q*q1 < 0){  
      p0 <- p1  
      q0 <- q1  
    }  
  
    p1 <- p  
    q1 <- q  
  }  
  
  return(paste("El método falló luego de ", n, " iteraciones"))  
}
```

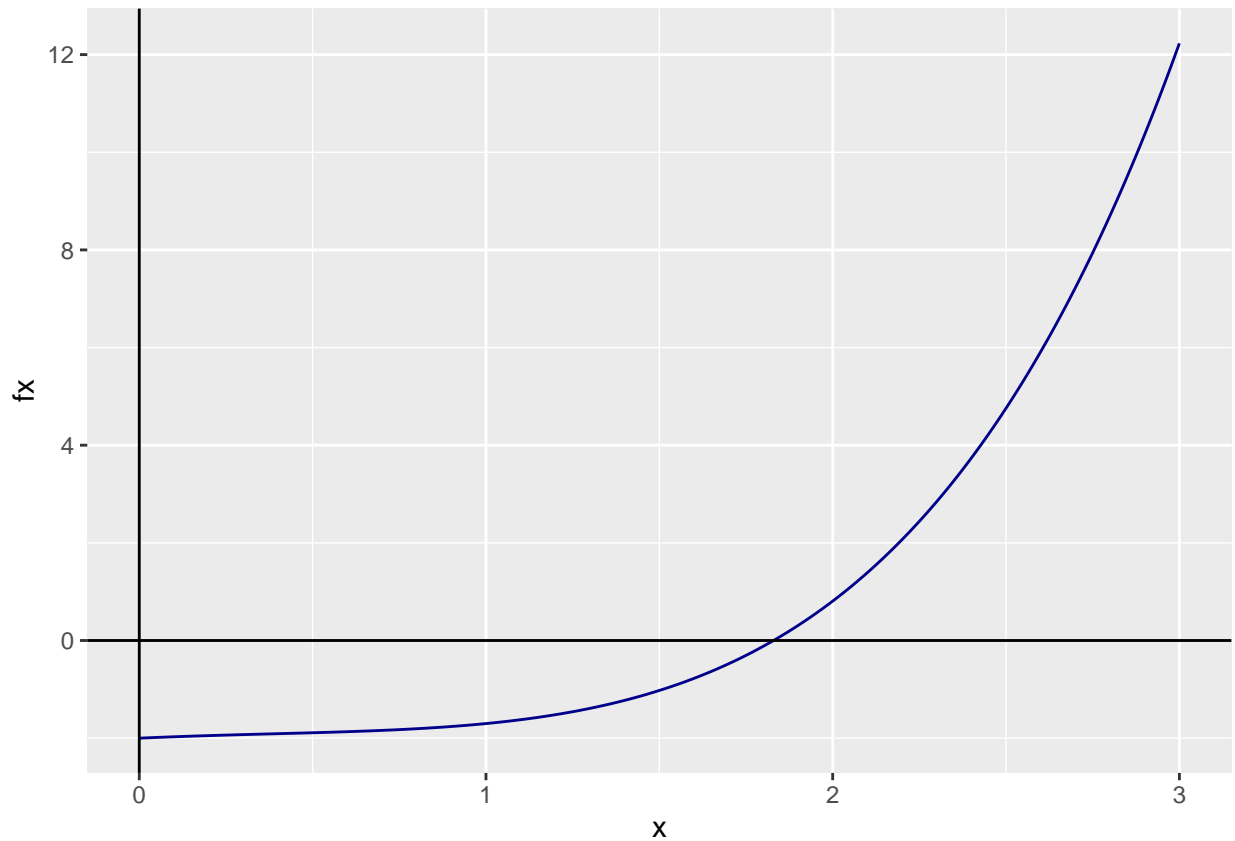
## Ejercicios:

- Hallar las soluciones de (si es posible):
  - $e^x + 2^{-x} + 2\cos(x) - 6 = 0 \quad 1 \leq x \leq 2$
  - $\ln(x-1) + \cos(x-1) = 0 \quad 1.3 \leq x \leq 2$
  - $2x * \cos(2x) - (x-2)^2 = 0 \quad 2 \leq x \leq 3 \text{ and } 3 \leq x \leq 4$
  - $(x-2)^2 - \ln(x) = 0 \quad 1 \leq x \leq 2 \text{ and } e \leq x \leq 4$
  - $e^x - 3x^2 = 0 \quad 0 \leq x \leq 1 \text{ and } 3 \leq x \leq 5$
  - $\sin(x) - e^{-x} = 0 \quad 0 \leq x \leq 1 \text{ and } 3 \leq x \leq 4 \text{ and } 6 \leq x \leq 7$
  - $\cos(x) = \sqrt{x}$
  - $2 + \cos(e^x - 2) = e^x$
  - $x^3 - 7x^2 + 14x - 6 = 0$
  - $-x^3 - \cos(x) = 0$

## Solución:

Ejercicio 1:

```
f <- function(x){  
  return(exp(x) + 2^(-x) + 2*cos(x)-6)  
}  
  
#Instancio un vector que me va a indicar los puntos en la función  
x <- seq(0, 3, by = 0.01)  
  
#Genero los puntos  
fx <- f(x)  
  
#Creo un data frame con los x e y  
df <- data.frame(x, fx)  
  
#Instancio los datos  
gg_fx <- ggplot(data = df)  
  
#Agrego la capa con los datos  
gg_fx <- gg_fx + aes(x = x, y = fx)  
  
#Est grafica una linea  
gg_fx <- gg_fx + geom_line(linetype = 1, colour = "darkblue")  
  
#Agrego el eje X  
gg_fx <- gg_fx + geom_vline(xintercept = 0, linetype = 1)  
  
#Agrego el eje Y  
gg_fx <- gg_fx + geom_hline(yintercept = 0, linetype = 1)  
  
#Grafico  
gg_fx
```



```
FalsaPosicion(p0 = 1.5, p1 = 2, tol = 0.01)
```

```
## [1] 1.828488
```

Ejercicio 2:

```
f <- function(x){
  return( log(x-1) + cos(x-1))
}

#Instancio un vector que me va a indicar los puntos en la función
x <- seq(1.3, 2, by = 0.01)

#Genero los puntos
fx <- f(x)

#Creo un data frame con los x e y
df <- data.frame(x, fx)

#Instancio los datos
gg_fx <- ggplot(data = df)

#Agrego la capa con los datos
gg_fx <- gg_fx + aes(x = x, y = fx)

#Est grafica una linea
gg_fx <- gg_fx + geom_line(linetype = 1, colour = "darkblue")
```

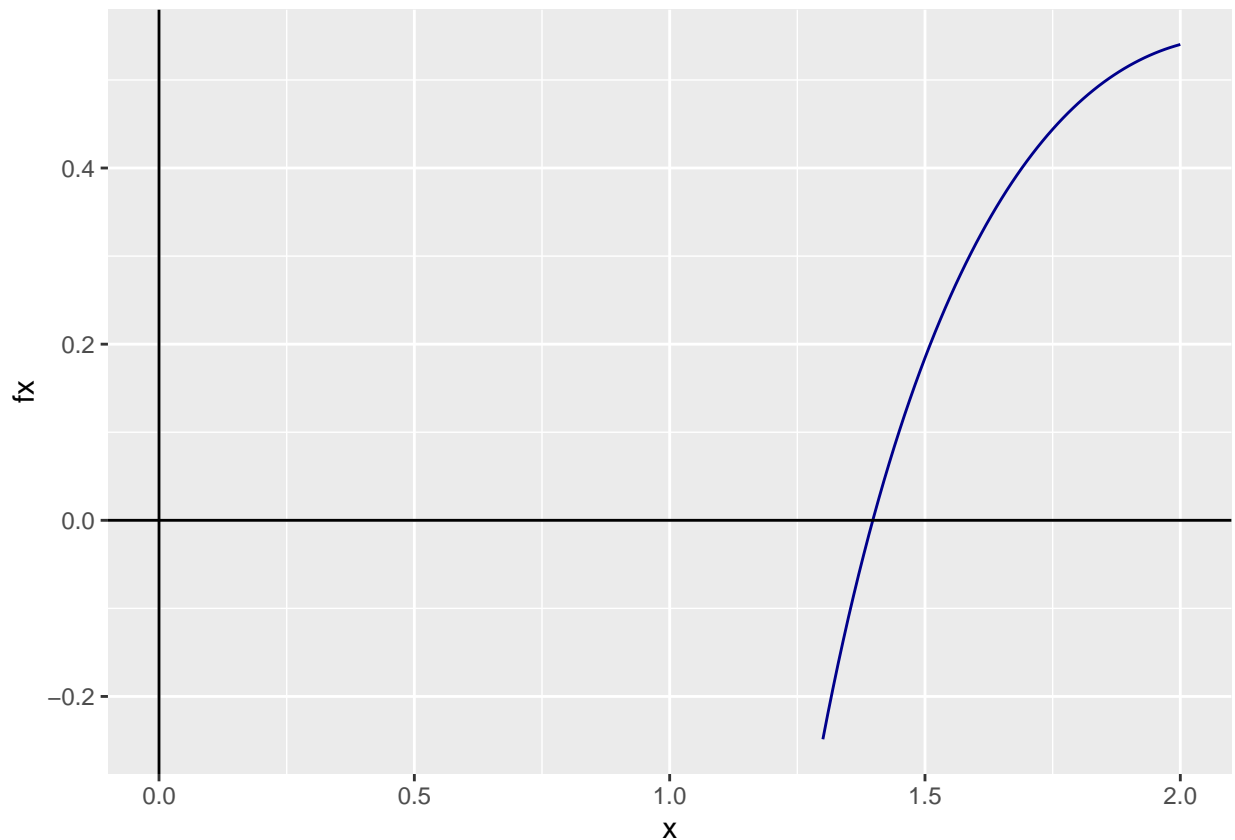
```

#Agrego el eje X
gg_fx <- gg_fx + geom_vline(xintercept = 0, linetype = 1)

#Agrego el eje Y
gg_fx <- gg_fx + geom_hline(yintercept = 0, linetype = 1)

#Grafico
gg_fx

```



```

FalsaPosicion(p0 = 1.25, p1 = 1.5, tol = 0.01)

```

```
## [1] 1.399319
```

Ejercicio 3:

```

f <- function(x){
  return( 2*x * cos(2*x) - (x-2)^2)
}

#Instancio un vector que me va a indicar los puntos en la función
x <- seq(2, 4, by = 0.01)

#Genero los puntos
fx <- f(x)

#Creo un data frame con los x e y
df <- data.frame(x, fx)

```

```

#Instancio los datos
gg_fx <- ggplot(data = df)

#Agrego la capa con los datos
gg_fx <- gg_fx + aes(x = x, y = fx)

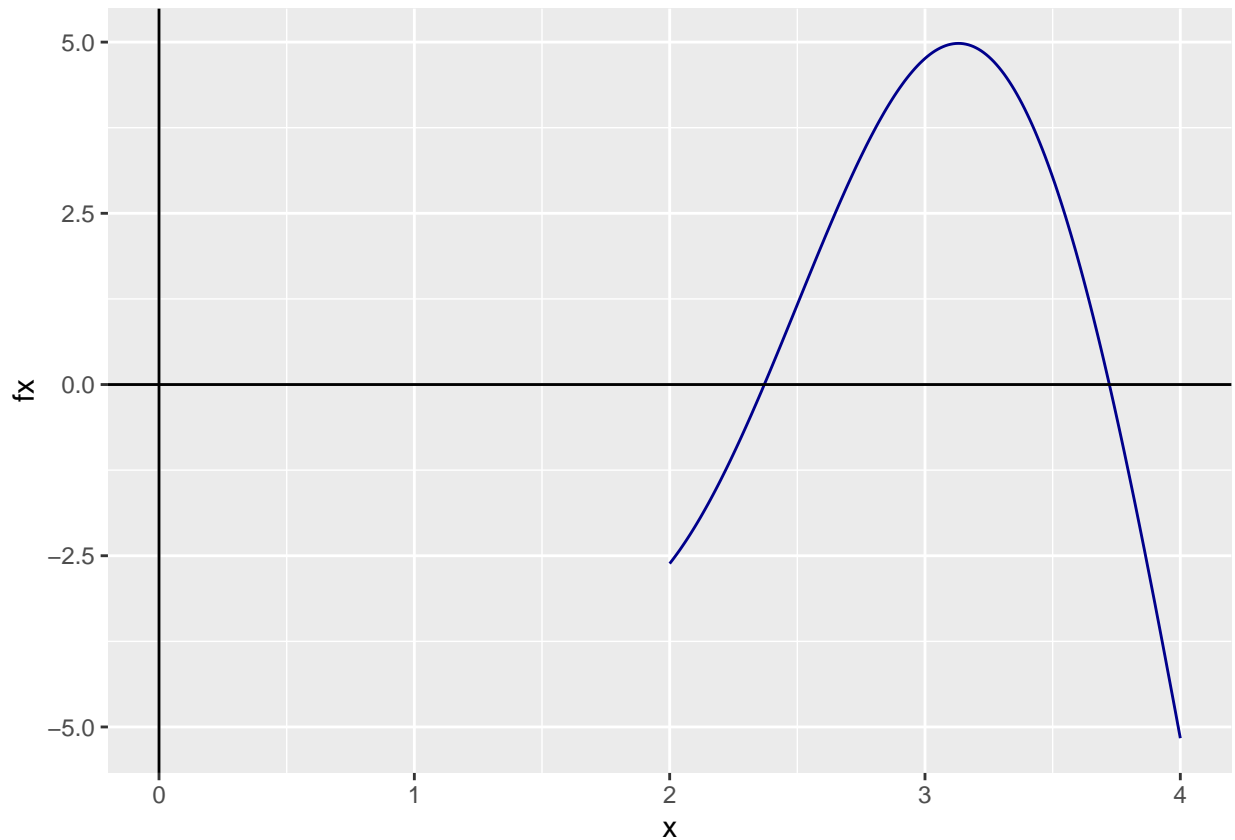
#Est grafica una linea
gg_fx <- gg_fx + geom_line(linetype = 1, colour = "darkblue")

#Agrego el eje X
gg_fx <- gg_fx + geom_vline(xintercept = 0, linetype = 1)

#Agrego el eje Y
gg_fx <- gg_fx + geom_hline(yintercept = 0, linetype = 1)

#Grafico
gg_fx

```



```
FalsaPosicion(p0 = 2, p1 = 2.5, tol = 0.01)
```

```
## [1] 2.370668
```

```
FalsaPosicion(p0 = 3.5, p1 = 4, tol = 0.01)
```

```
## [1] 3.721515
```

Ejercicio 4:

```

f <- function(x){
  return((x-2)^2 - log(x))
}

#Instancio un vector que me va a indicar los puntos en la función
x <- seq(1, 4, by = 0.01)

#Genero los puntos
fx <- f(x)

#Creo un data frame con los x e y
df <- data.frame(x, fx)

#Instancio los datos
gg_fx <- ggplot(data = df)

#Agrego la capa con los datos
gg_fx <- gg_fx + aes(x = x, y = fx)

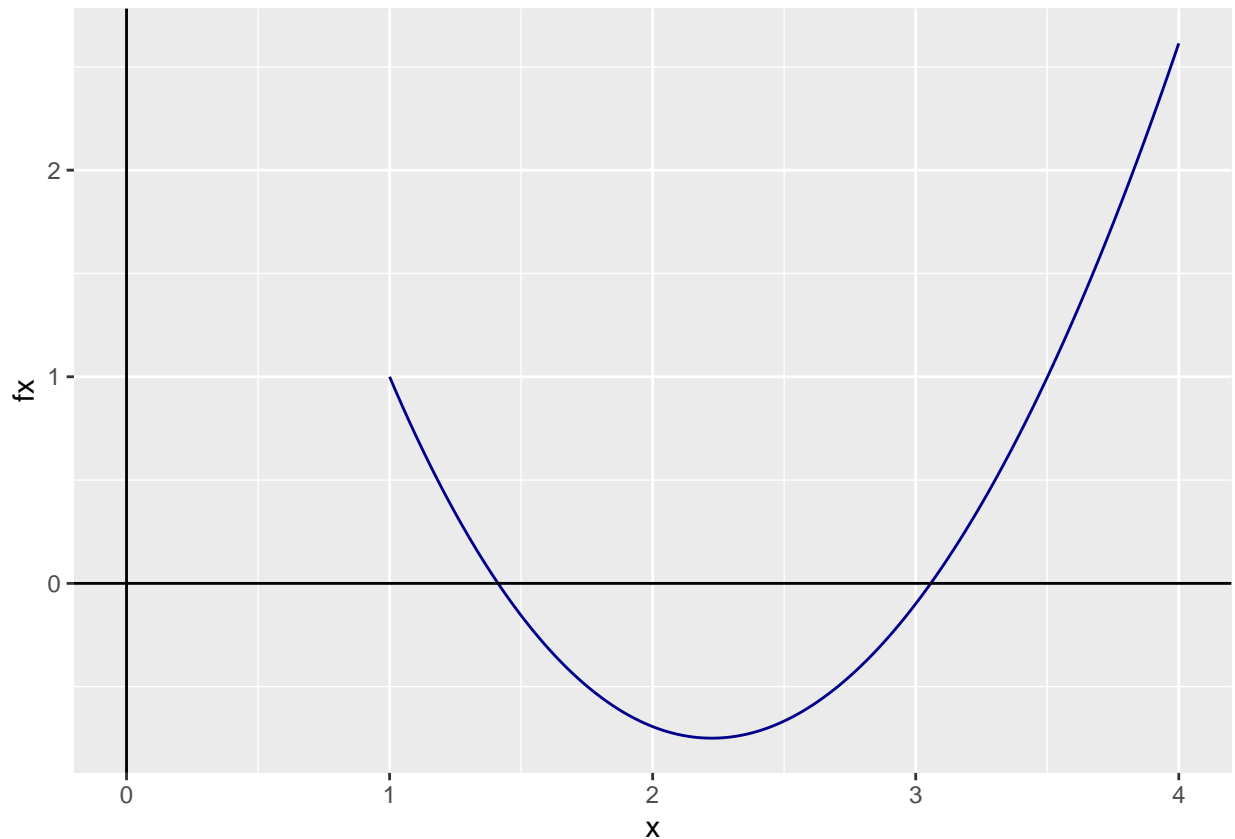
#Est grafica una linea
gg_fx <- gg_fx + geom_line(linetype = 1, colour = "darkblue")

#Agrego el eje X
gg_fx <- gg_fx + geom_vline(xintercept = 0, linetype = 1)

#Agrego el eje Y
gg_fx <- gg_fx + geom_hline(yintercept = 0, linetype = 1)

#Grafico
gg_fx

```



```
FalsaPosicion(p0 = 1, p1 = 1.5, tol = 0.01)
```

```
## [1] 1.413417
```

```
FalsaPosicion(p0 = 3, p1 = 3.5, tol = 0.01)
```

```
## [1] 3.054591
```

Ejercicio 5:

```
f <- function(x){
  return(exp(x) - 3*x^2)
}
```

```
#Instancio un vector que me va a indicar los puntos en la función
x <- seq(0, 5, by = 0.1)
```

```
#Genero los puntos
fx <- f(x)
```

```
#Creo un data frame con los x e y
df <- data.frame(x, fx)
```

```
#Instancio los datos
gg_fx <- ggplot(data = df)
```

```
#Agrego la capa con los datos
gg_fx <- gg_fx + aes(x = x, y = fx)
```

```

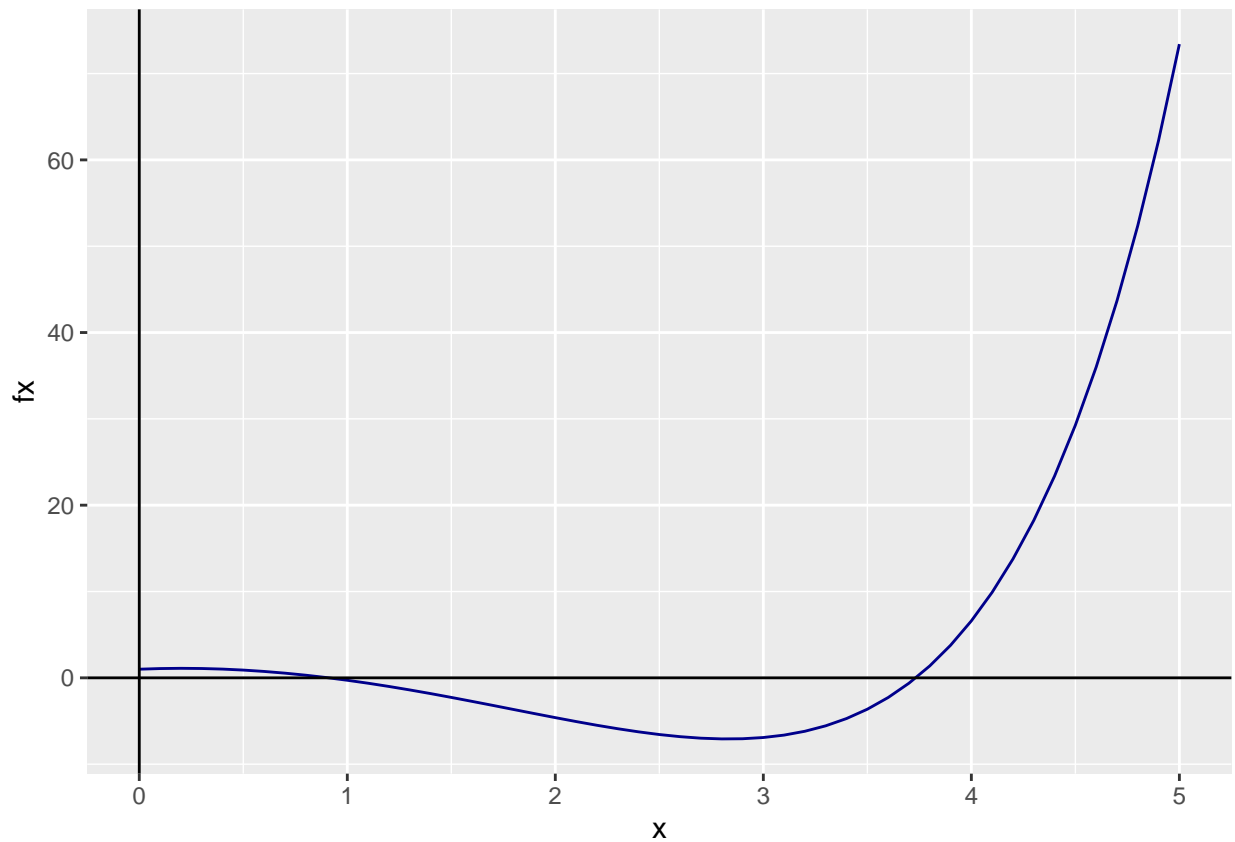
#Est grafica una linea
gg_fx <- gg_fx + geom_line(linetype = 1, colour = "darkblue")

#Agrego el eje X
gg_fx <- gg_fx + geom_vline(xintercept = 0, linetype = 1)

#Agrego el eje Y
gg_fx <- gg_fx + geom_hline(yintercept = 0, linetype = 1)

#Grafico
gg_fx

```



```
FalsaPosicion(p0 = 0.5, p1 = 1, tol = 0.01)
```

```
## [1] 0.9099341
```

```
FalsaPosicion(p0 = 3.5, p1 = 4, tol = 0.01)
```

```
## [1] 3.730442
```

Ejercicio 6:

```

f <- function(x){
  return(sin(x) - exp(-x))
}

```

```
#Instancio un vector que me va a indicar los puntos en la función
```



```

x <- seq(0, 7, by = 0.1)

#Genero los puntos
fx <- f(x)

#Creo un data frame con los x e y
df <- data.frame(x, fx)

#Instancio los datos
gg_fx <- ggplot(data = df)

#Agrego la capa con los datos
gg_fx <- gg_fx + aes(x = x, y = fx)

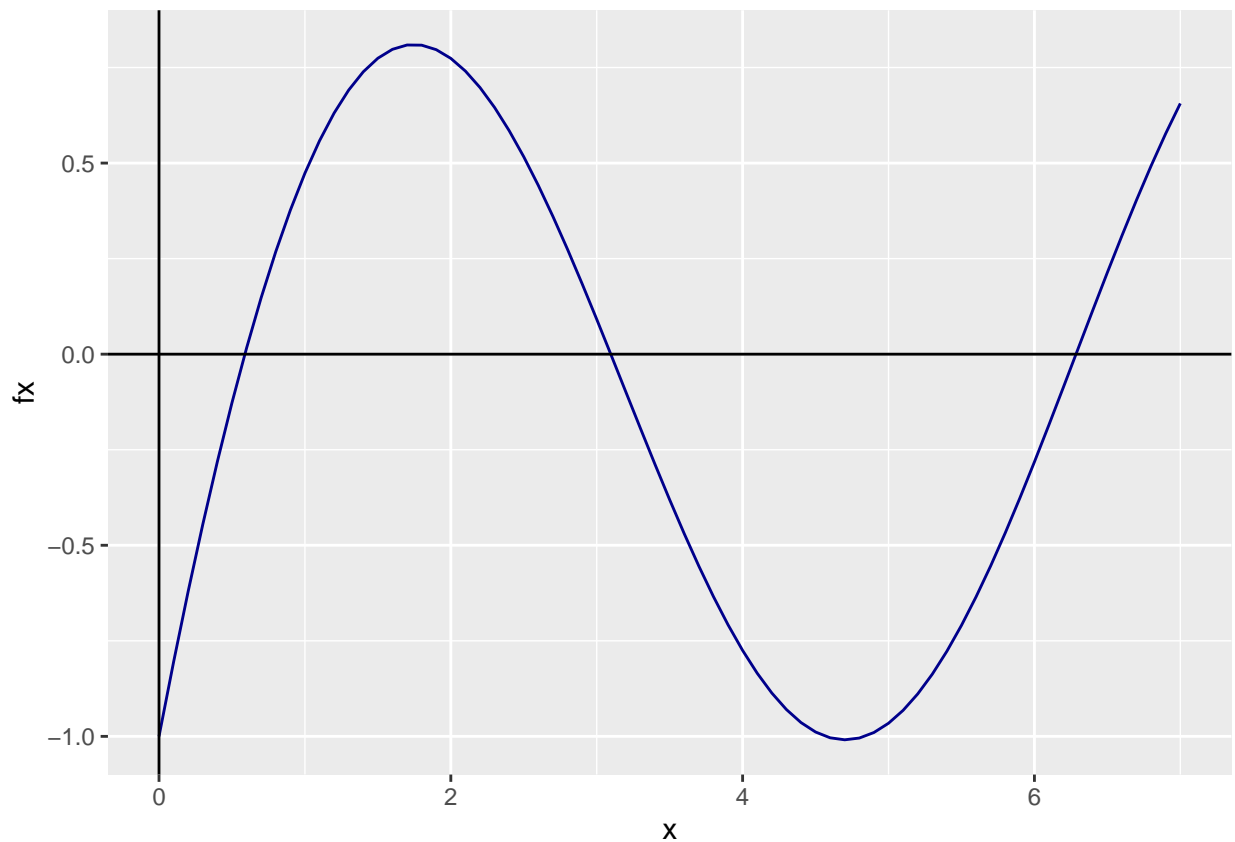
#Est grafica una linea
gg_fx <- gg_fx + geom_line(linetype = 1, colour = "darkblue")

#Agrego el eje X
gg_fx <- gg_fx + geom_vline(xintercept = 0, linetype = 1)

#Agrego el eje Y
gg_fx <- gg_fx + geom_hline(yintercept = 0, linetype = 1)

#Grafico
gg_fx

```



```
FalsaPosicion(p0 = 0, p1 = 1, tol = 0.01)
```

```
## [1] 0.5891168
```

```
FalsaPosicion(p0 = 3, p1 = 4, tol = 0.01)
```

```
## [1] 3.096308
```

```
FalsaPosicion(p0 = 6, p1 = 7, tol = 0.01)
```

```
## [1] 6.285049
```

Ejercicio 7:

```
f <- function(x){  
  return(cos(x) - sqrt(x))  
}
```

```
#Instancio un vector que me va a indicar los puntos en la función  
x <- seq(0, 7, by = 0.1)
```

```
#Genero los puntos  
fx <- f(x)
```

```
#Creo un data frame con los x e y  
df <- data.frame(x, fx)
```

```
#Instancio los datos  
gg_fx <- ggplot(data = df)
```

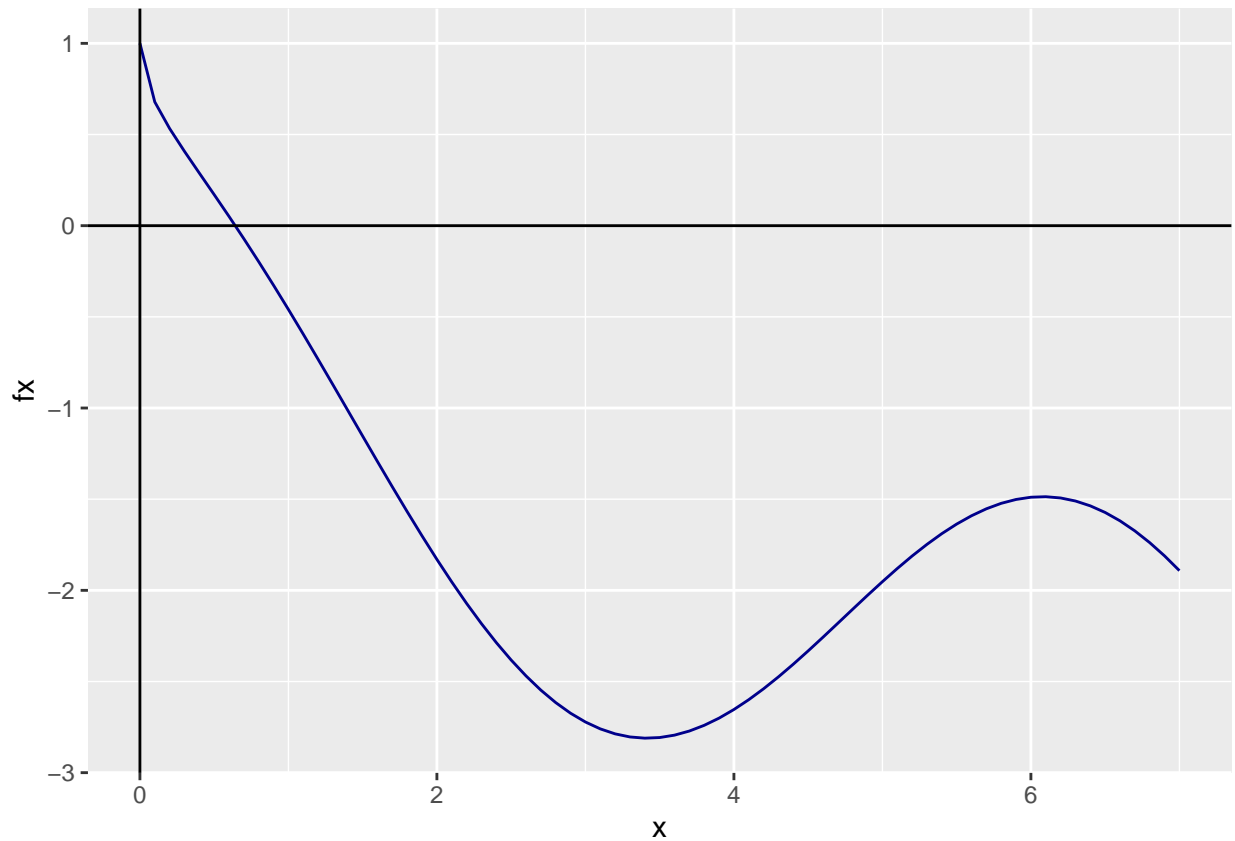
```
#Agrego la capa con los datos  
gg_fx <- gg_fx + aes(x = x, y = fx)
```

```
#Est grafica una linea  
gg_fx <- gg_fx + geom_line(linetype = 1, colour = "darkblue")
```

```
#Agrego el eje X  
gg_fx <- gg_fx + geom_vline(xintercept = 0, linetype = 1)
```

```
#Agrego el eje Y  
gg_fx <- gg_fx + geom_hline(yintercept = 0, linetype = 1)
```

```
#Grafico  
gg_fx
```



```
FalsaPosicion(p0 = 0.5, p1 = 1, tol = 0.01)
```

```
## [1] 0.6414111
```

Ejercicio 8:

```
f <- function(x){
  return(2 + cos(exp(x) - 2) - exp(x))
}

#Instancio un vector que me va a indicar los puntos en la función
x <- seq(0, 7, by = 0.1)

#Genero los puntos
fx <- f(x)

#Creo un data frame con los x e y
df <- data.frame(x, fx)

#Instancio los datos
gg_fx <- ggplot(data = df)

#Agrego la capa con los datos
gg_fx <- gg_fx + aes(x = x, y = fx)

#Est grafica una linea
gg_fx <- gg_fx + geom_line(linetype = 1, colour = "darkblue")
```

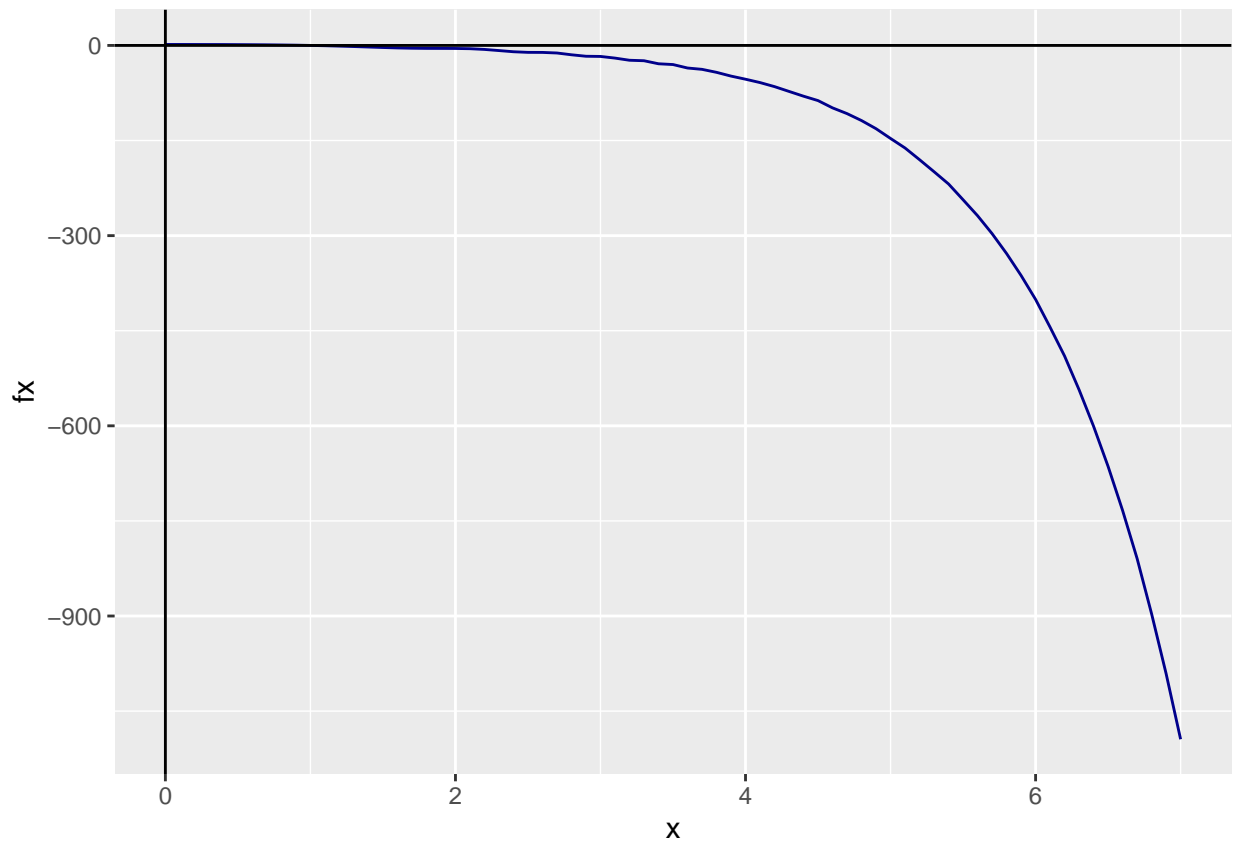
```

#Agrego el eje X
gg_fx <- gg_fx + geom_vline(xintercept = 0, linetype = 1)

#Agrego el eje Y
gg_fx <- gg_fx + geom_hline(yintercept = 0, linetype = 1)

#Grafico
gg_fx

```



```

FalsaPosicion(p0 = 0, p1 = 1, tol = 0.01)

```

```
## [1] 1.007622
```

Ejercicio 9:

```

f <- function(x){
  return(x^3 - 7*x^2 + 14*x - 6)
}

#Instancio un vector que me va a indicar los puntos en la función
x <- seq(0, 7, by = 0.1)

#Genero los puntos
fx <- f(x)

#Creo un data frame con los x e y
df <- data.frame(x, fx)

```

```

#Instancio los datos
gg_fx <- ggplot(data = df)

#Agrego la capa con los datos
gg_fx <- gg_fx + aes(x = x, y = fx)

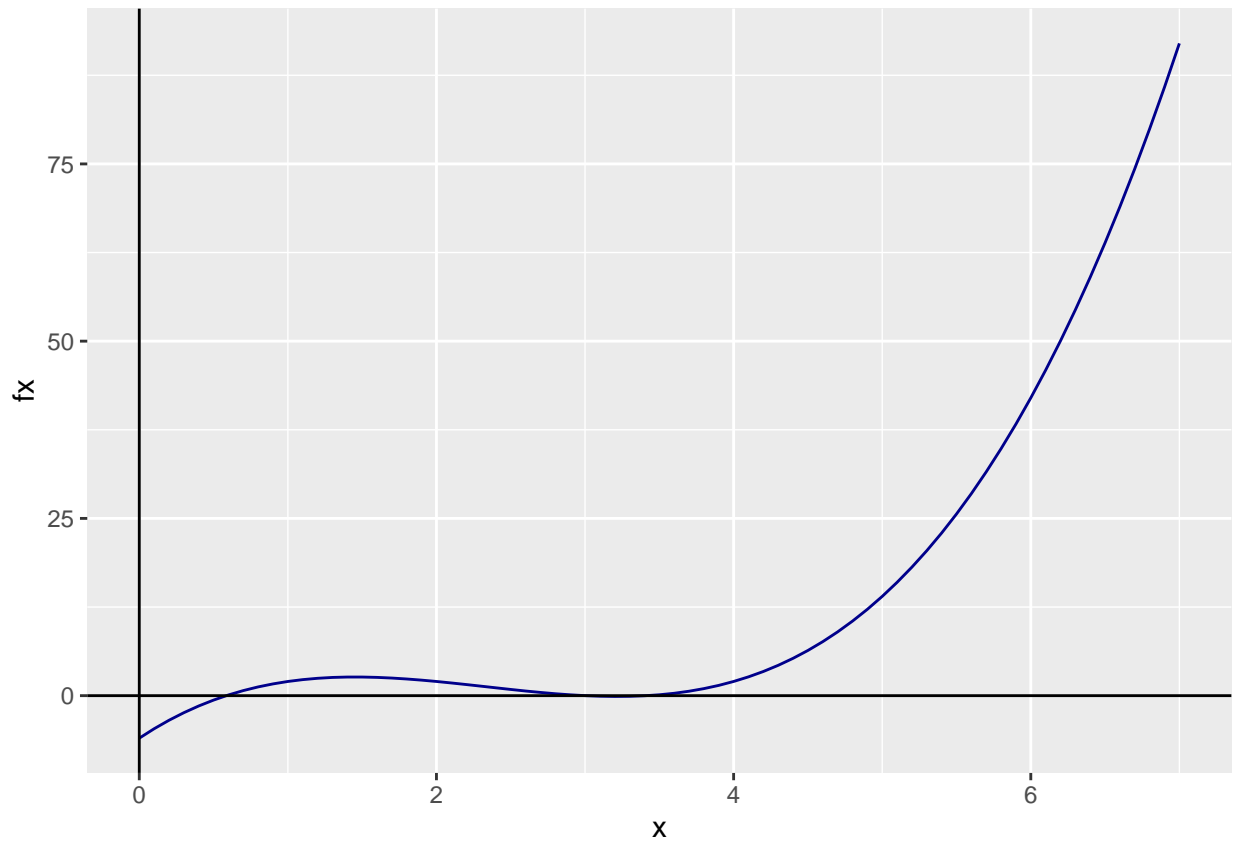
#Est grafica una linea
gg_fx <- gg_fx + geom_line(linetype = 1, colour = "darkblue")

#Agrego el eje X
gg_fx <- gg_fx + geom_vline(xintercept = 0, linetype = 1)

#Agrego el eje Y
gg_fx <- gg_fx + geom_hline(yintercept = 0, linetype = 1)

#Grafico
gg_fx

```



```

FalsaPosicion(p0 = 0.1, p1 = 1, tol = 0.01)

## [1] 0.5896053

FalsaPosicion(p0 = 2.9, p1 = 3.1, tol = 0.01)

## [1] 3.000542

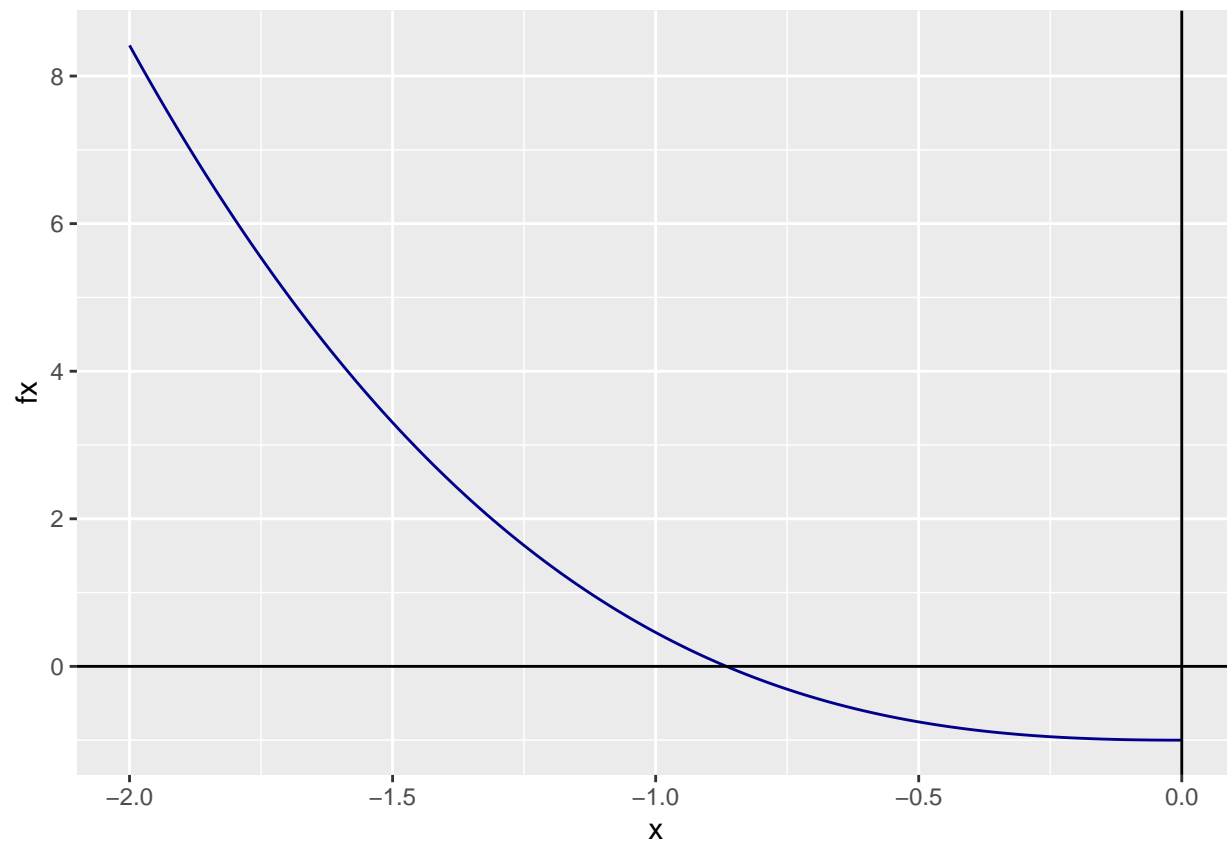
```

```
FalsaPosicion(p0 = 3.5, p1 = 4, tol = 0.01)
```

```
## [1] 3.416004
```

Ejercicio 10:

```
f <- function(x){  
  return(-x^3 - cos(x))  
}  
  
#Instancio un vector que me va a indicar los puntos en la función  
x <- seq(-2, 0, by = 0.01)  
  
#Genero los puntos  
fx <- f(x)  
  
#Creo un data frame con los x e y  
df <- data.frame(x, fx)  
  
#Instancio los datos  
gg_fx <- ggplot(data = df)  
  
#Agrego la capa con los datos  
gg_fx <- gg_fx + aes(x = x, y = fx)  
  
#Est grafica una linea  
gg_fx <- gg_fx + geom_line(linetype = 1, colour = "darkblue")  
  
#Agrego el eje X  
gg_fx <- gg_fx + geom_vline(xintercept = 0, linetype = 1)  
  
#Agrego el eje Y  
gg_fx <- gg_fx + geom_hline(yintercept = 0, linetype = 1)  
  
#Grafico  
gg_fx
```



```
FalsaPosicion(p0 = -1, p1 = -0.75, tol = 0.01)
```

```
## [1] -0.86526
```