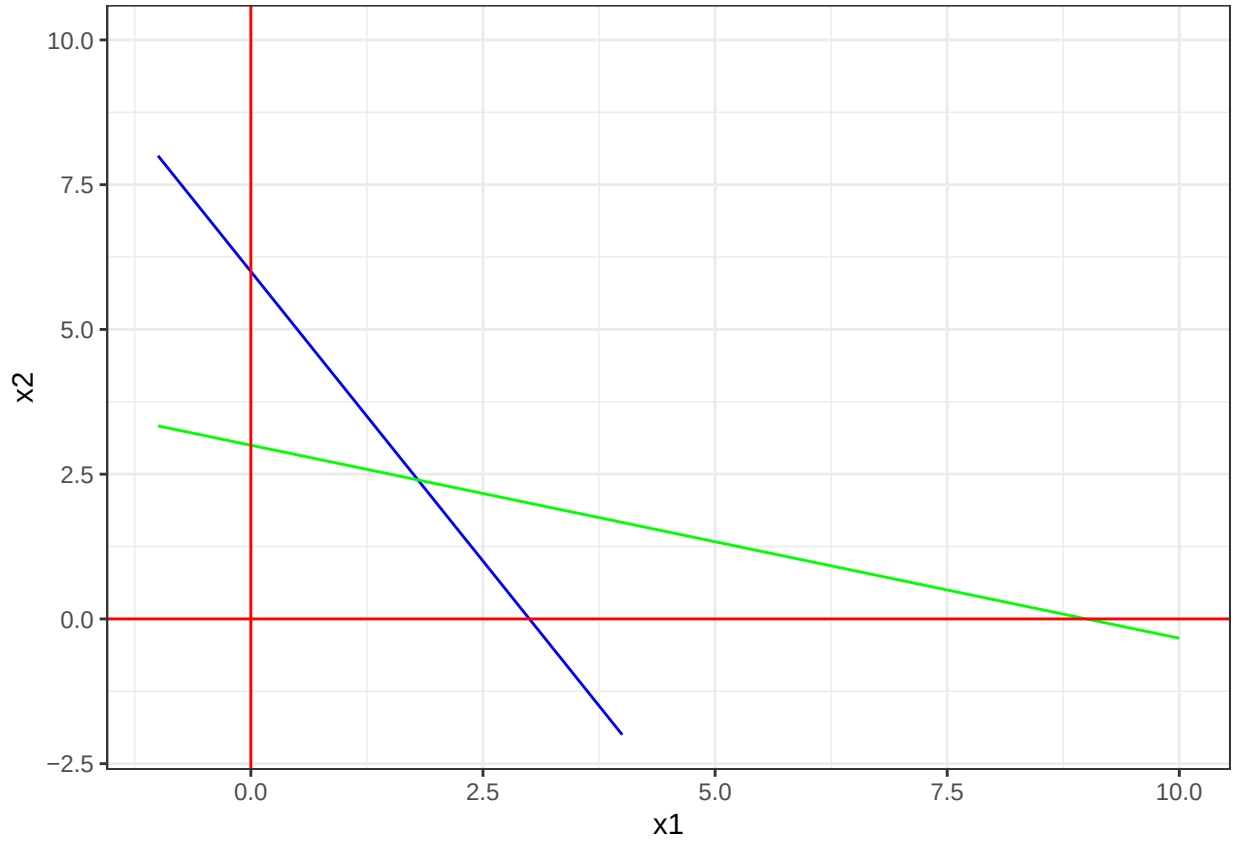
Untitled

2026-02-02

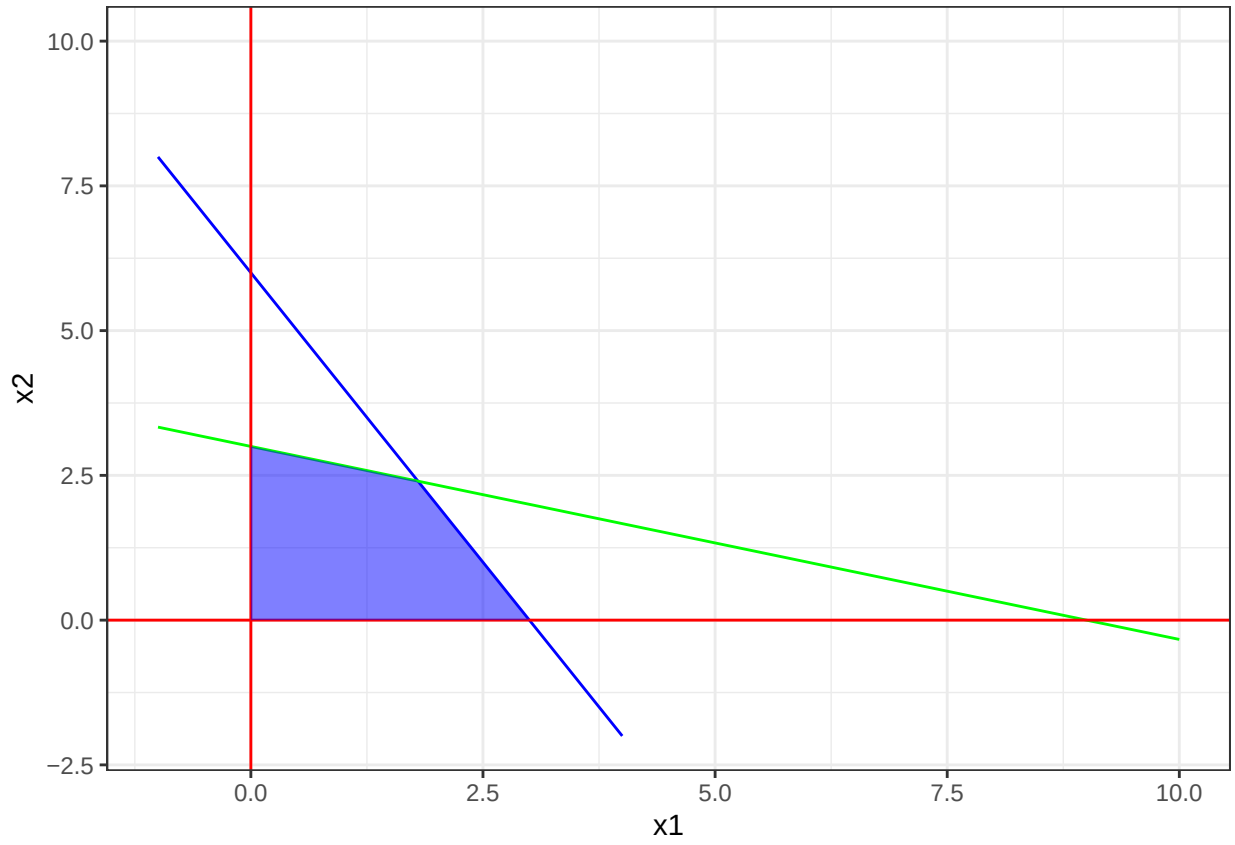
```
R1 <- function(x) 6-2*x
R2 <- function(x) 3 - x/3
x1 <- seq(-1,10, length.out=100)
datos <- data.frame(x1, x2=R1(x1), x2.2=R2(x1))
p <- ggplot(datos, aes(x = x1)) +
geom_line(aes(y = x2), colour = "blue") +
geom_line(aes(y = x2.2), colour = "green") +
geom_hline(yintercept = 0, colour = "red") +
geom_vline(xintercept = 0, colour="red") +
ylim(c(-2,10)) + theme_bw()
p
```

```
## Warning: Removed 54 rows containing missing values or values outside the scale range
## (`geom_line()`).
```




```
datos <- transform(datos, z = pmax(0,pmin(x2,x2.2)))
p + geom_ribbon(data=subset(datos, 0 <= x1),
aes(ymin=rep(0,length(z)),ymax=z), fill="blue", alpha=0.5)
```

```
## Warning: Removed 54 rows containing missing values or values outside the scale range
## (`geom_line()`).
```




```
A = rbind(c(2,1), #R1
          c(1,3), #R2
          c(0,1), #Eje
          c(1,0)) #Eje

b = c(6,9,0,0)

pto.R1R2 = solve(A[1:2,],b[1:2]); pto.R1R2
```

```
## [1] 1.8 2.4
```

```
pto.R1 = solve(A[c(1,3),], b[c(1,3)]); pto.R1
```

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

```
pto.R2 = solve(A[c(2,4),], b[c(2,4)]); pto.R2
```

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

```
# ¿Cómo graficar esto?
puntos = rbind(pto.R2, pto.R1, pto.R1R2)
puntos = as.data.frame(puntos)
puntos
```

```
##           V1  V2
```

```
## pto.R2   0.0 3.0
## pto.R1   3.0 0.0
## pto.R1R2 1.8 2.4
```

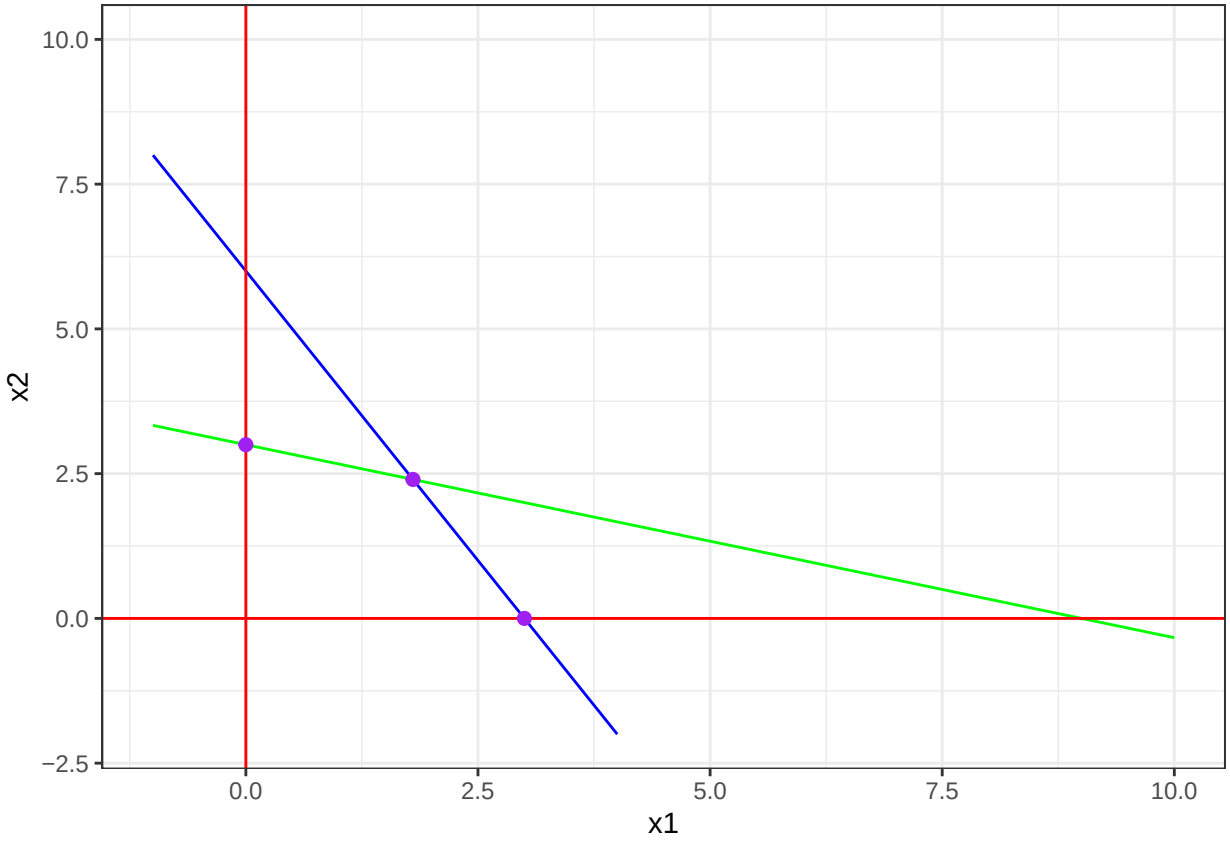
Los nombres anteriores no son muy descriptivos, es mejor cambiarlos con `names`:

```
names(puntos) = c("x1", "x2")
puntos
```

```
##           x1  x2
## pto.R2   0.0 3.0
## pto.R1   3.0 0.0
## pto.R1R2 1.8 2.4
```

```
p = p + geom_point(data = puntos, aes(x=x1,y=x2), col="purple", size=2)
p
```

```
## Warning: Removed 54 rows containing missing values or values outside the scale range
## (`geom_line()`).
```



Queda resolver la optimización como tal. Para ello, definiremos la función objetivo:

```
f.obj = function(x) 3*x[1] + x[2]

# Comprobamos que evalúe como es esperable
f.obj(pto.R1)
```

```
## [1] 9
```

```
f.obj(pto.R2)
```

```
## [1] 3
```

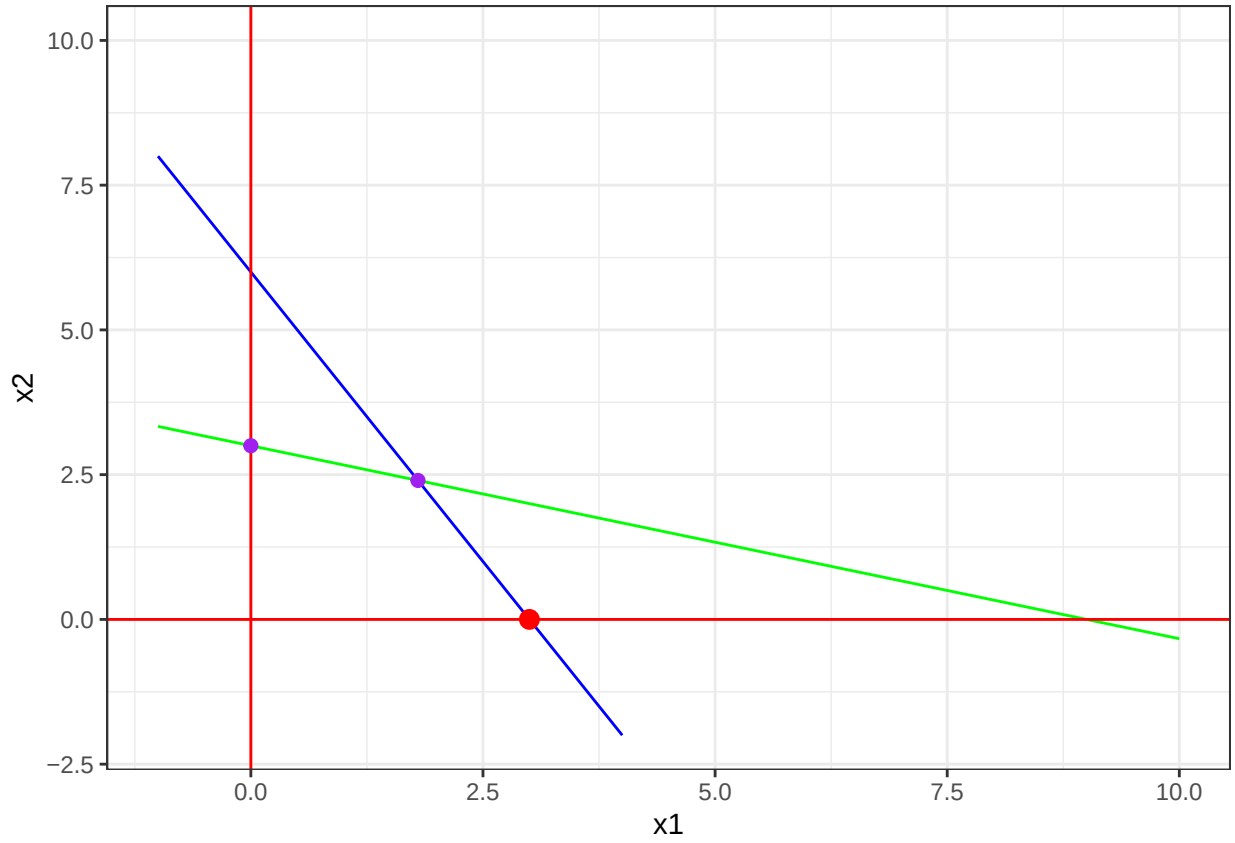
```
f.obj(pto.R1R2)
```

```
## [1] 7.8
```

La solución óptima se alcanza en `pto.R1`: $(3, 0)$. Veamos cómo representarlo.

```
p + geom_point(data=puntos["pto.R1",], aes(x=x1, y=x2), col="red", size = 3)
```

```
## Warning: Removed 54 rows containing missing values or values outside the scale range
## (`geom_line()`).
```



El orden de las capas sí importa. Es imperativo haber seguido los pasos en el orden anterior.