

## 2. Programación Lineal

Javier Ortín

2026-02-12

### Ejercicios propuestos

#### Solución gráfica

##### Ejercicio 2.1.3 (a)

Buscamos minimizar la función  $-x_1 + 3x_2$  con las siguientes restricciones:

$$\begin{cases} x_1 - x_2 \leq 4 \\ x_1 + 2x_2 \geq 4 \\ x_1, x_2 \geq 0 \end{cases}$$

Comenzamos definiendo las restricciones:

```
R1 = function(x1) x1 - 4
R2 = function(x1) 2 - x1/2
```

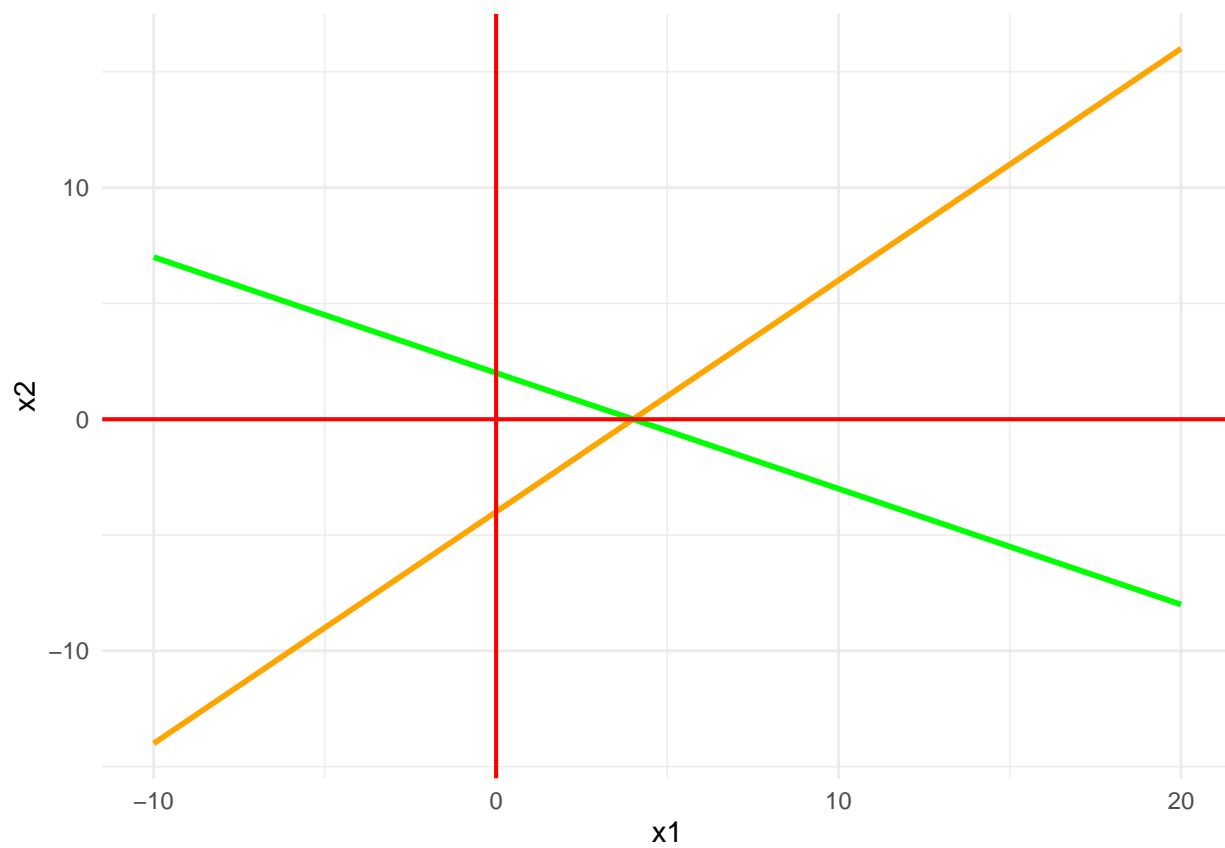
Despejando, solola variable  $x_1$  solo toma valores en  $[0, 4]$ . Este será nuestro rango de representación para graficar la región factible.

```
x1 = seq(from=-10, to=20, length.out=1000) # Valores de x1 en el rango deseado
datos = data.frame(x1=x1,
                   x2=R1(x1),
                   x2.2=R2(x1))

p = ggplot(data=datos, aes(x=x1)) +
  geom_line(aes(y=x2), col="orange", size=1) +
  geom_line(aes(y=x2.2), col="green", size=1) +
  geom_vline(xintercept = 0, col="red", size=0.72) +
  geom_hline(yintercept = 0, col="red", size=0.72) +
  theme_minimal()
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

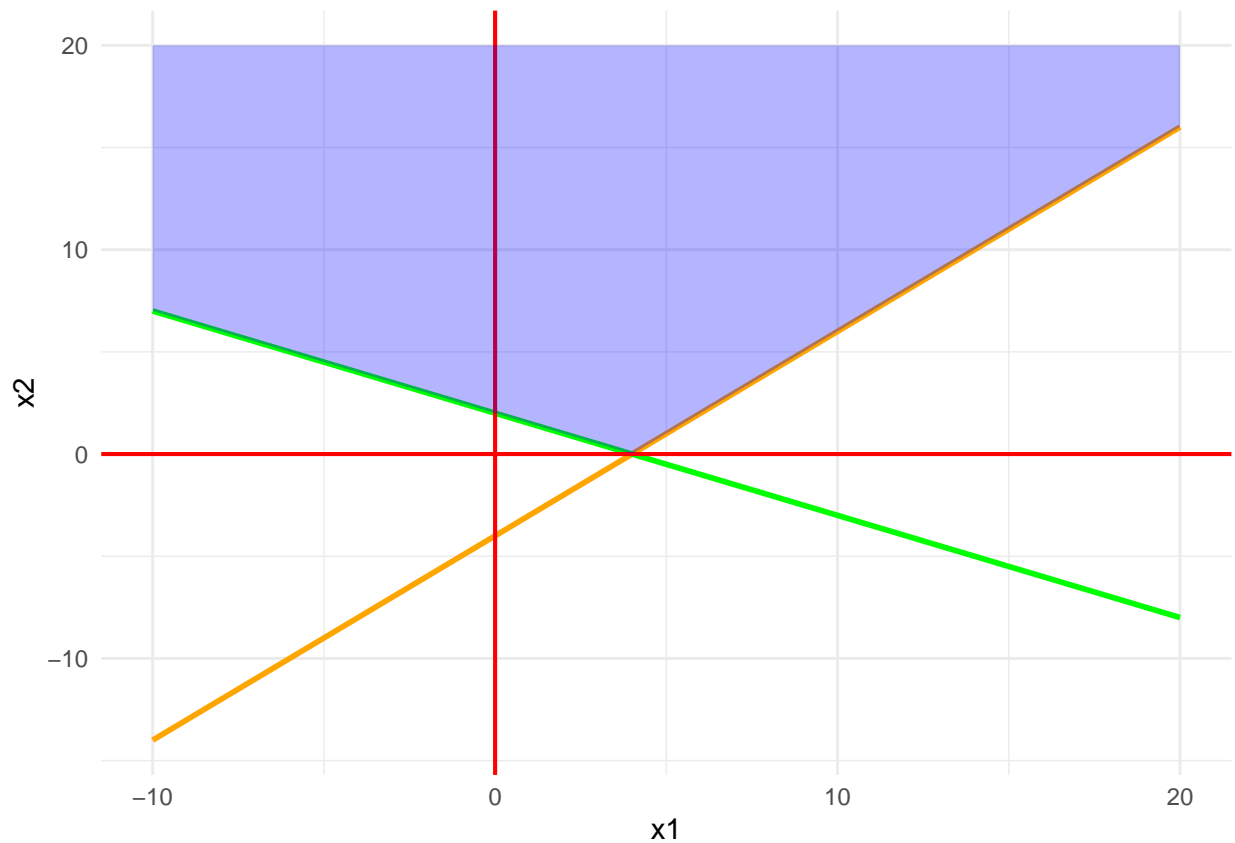
p



Ya hemos visto las restricciones, queda ver el área plausible:

```
p = p + geom_ribbon(data = transform(datos,  
                                     maxY= 20,  
                                     minY= pmax(datos$x2, datos$x2.2)),  
                   aes(ymin=minY, ymax=maxY), fill="blue", alpha=0.3)
```

p



Ahora queda hallar los puntos extremos y evaluar la función en ellos:

```
A = rbind(c(1,-1), # R1
          c(1,2),  # R2
          c(1,0),  # x1 = 0
          c(0,1))  # x2 = 0

# Vector del lado derecho
b = c(4,4,0,0)

# En este caso, la región crítica tiene un solo vértice
punto = solve(A[c(1,2),], b[1:2])

puntos = rbind(punto)
puntos = data.frame(puntos)

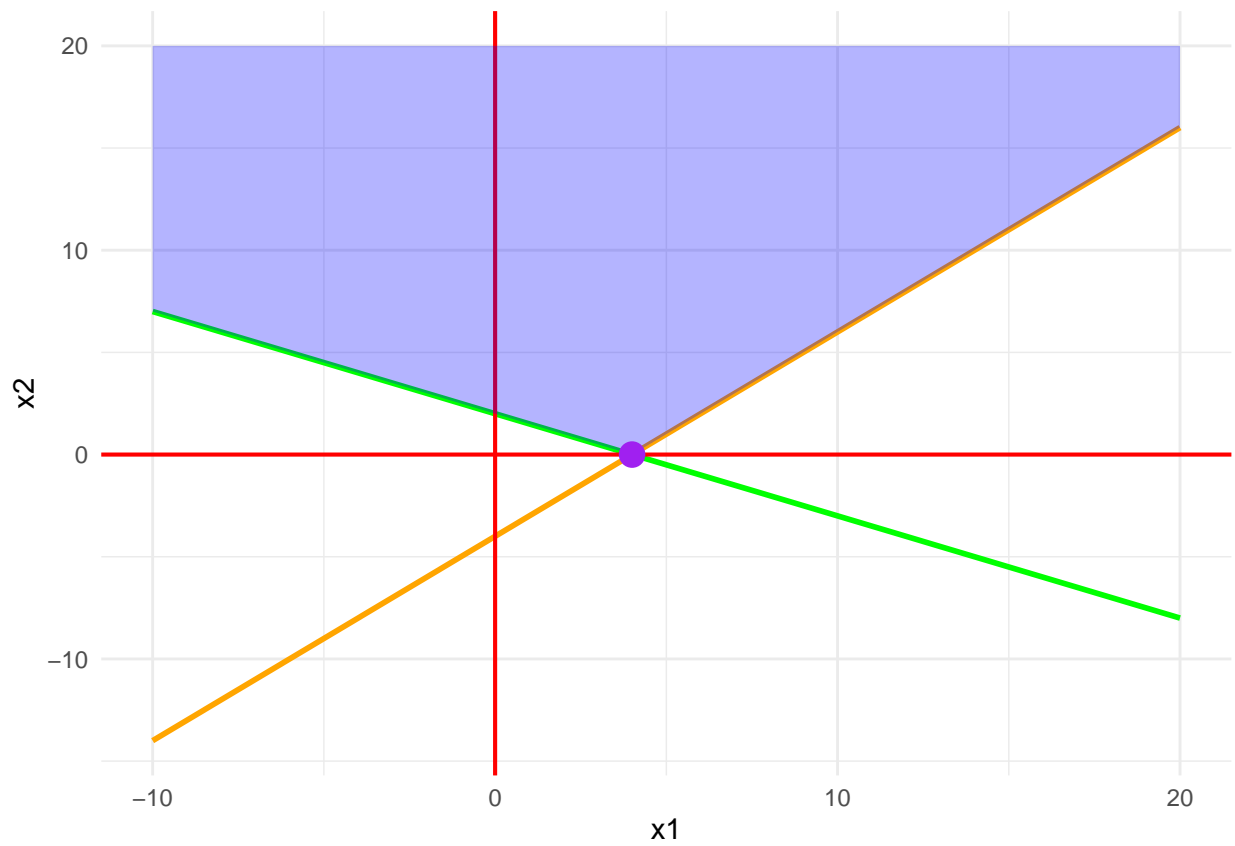
f_obj = function(x) - x$X1 + 3*x$X2
puntos[which.min(f_obj(puntos)),]

##      X1 X2
## punto 4  0
f_obj(puntos)

## [1] -4
```

Grafiquemos el resultado:

```
p + geom_point(data=puntos, aes(x=X1, y=X2), col="purple", size=4)
```



La región factible es no acotada, y tiene un único punto extremo.

### Ejercicio 2.1.3(b)

Maximizar  $4x_1 + x_2$  sujeto a:

$$\begin{cases} 8x_1 + 2x_2 \leq 16 \\ 5x_1 + 2x_2 \leq 12 \\ x_1, x_2 \geq 0 \end{cases}$$

Al igual que en el caso anterior, comenzamos definiendo las instrucciones:

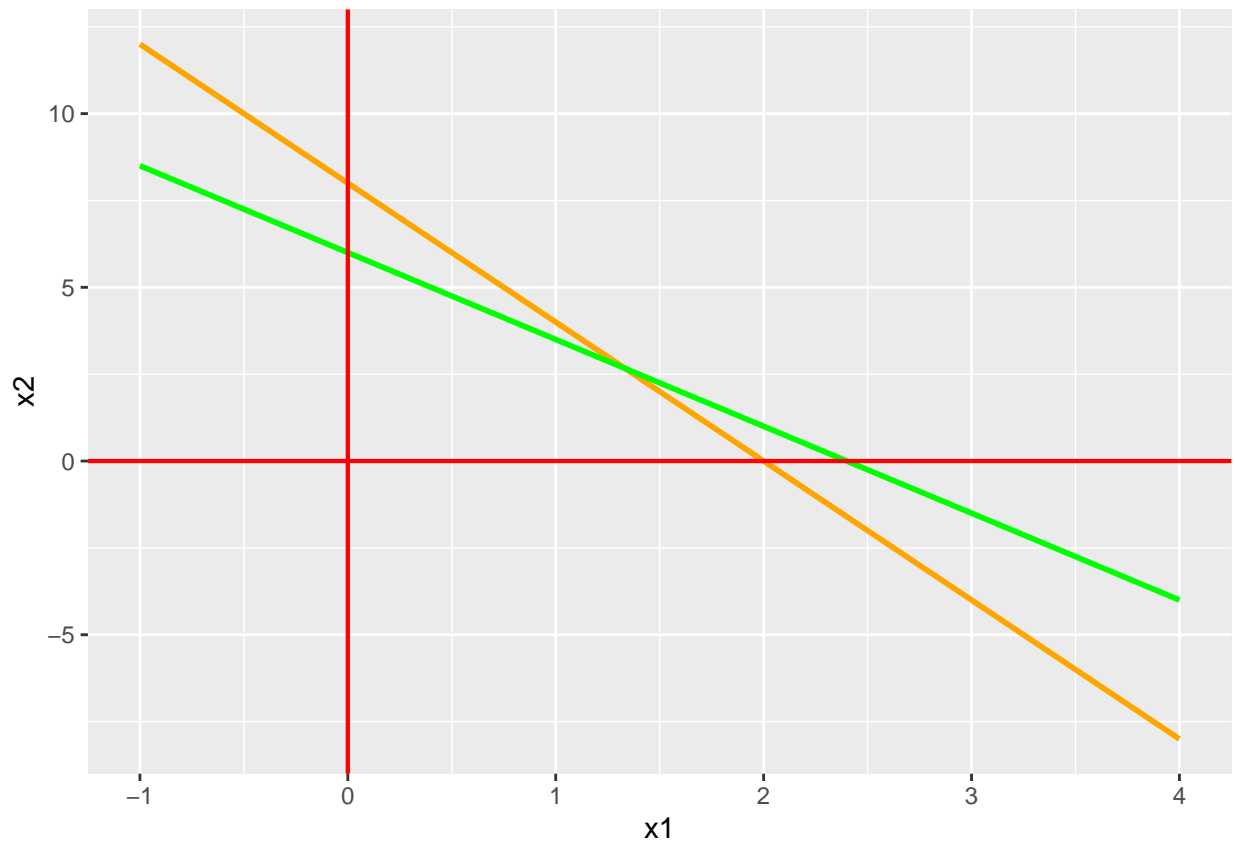
```
R1 = function(x1) 8 - 4*x1
R2 = function(x1) 6 - 5*x1/2
x1 = seq(-1, 4, length.out=100)

datos = data.frame(x1=x1,
                   x2=R1(x1),
                   x2.2=R2(x1))
```

Grafiquemos las restricciones:

```
p = ggplot(data=datos, aes(x=x1)) +
  geom_line(aes(y=x2), col="orange", size=1) +
  geom_line(aes(y=x2.2), col="green", size=1) +
  geom_hline(yintercept=0, col="red", size=0.75) +
  geom_vline(xintercept=0, col="red", size=0.75)
```

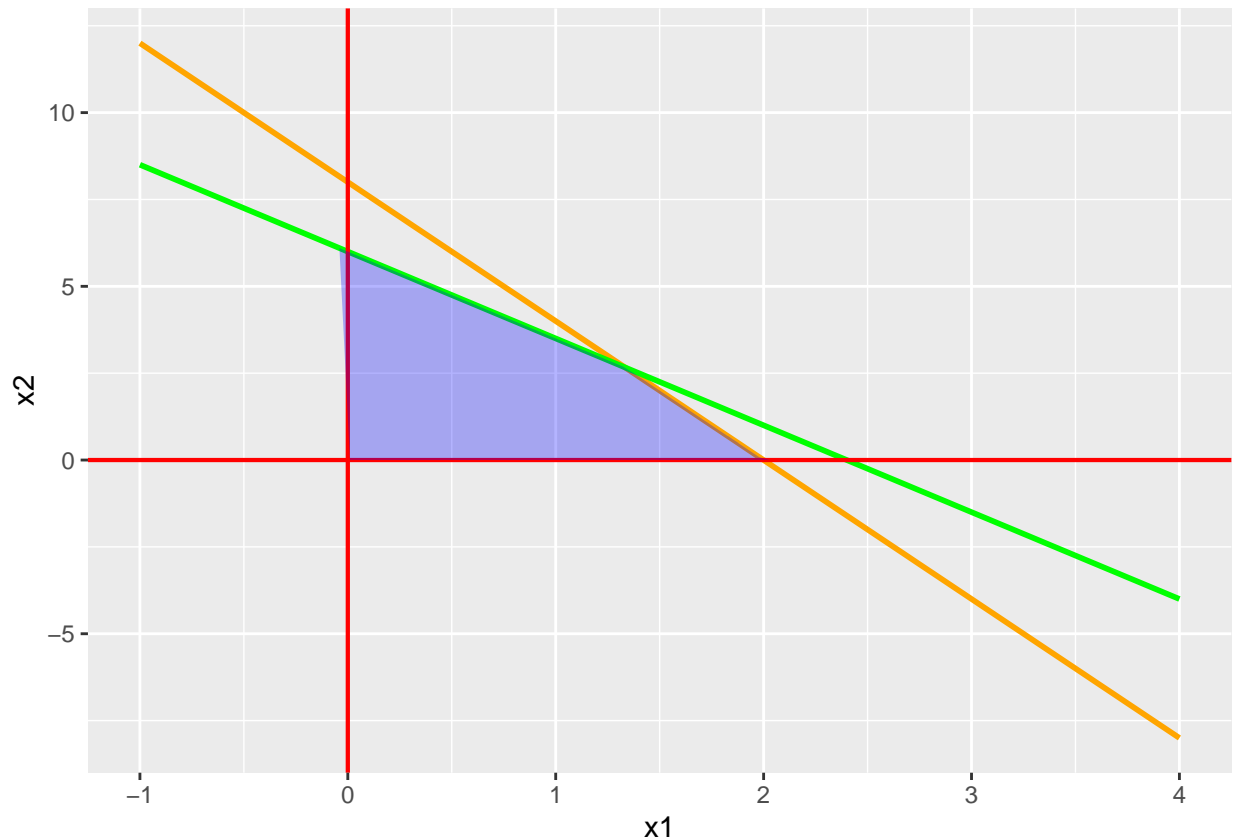
p



Pasando ahora a la región factible:

```
p = p + geom_ribbon(data=transform(datos,
                                yMin=ifelse(datos$x1>=0,0,datos$x2.2),
                                yMax=ifelse(pmin(datos$x2, datos$x2.2) > 0, pmin(datos$x2, datos$x2.2), (
                                aes(ymin = yMin, ymax=yMax),
                                fill="blue", alpha=0.3)
```

p



Veamos ahora qué puntos extremos tiene el conjunto y cuál es su máximo:

```
A = rbind(c(8,2), # R1
          c(5,2), # R2
          c(1,0), # x1 >= 0
          c(0,1)) # x2 >= 0

b = c(16,12,0,0)

pto.R1R2 = solve(A[c(1,2),], b[c(1,2)])
pto.R10 = solve(A[c(1,4),], b[c(1,4)])
pto.R20 = solve(A[c(2,3),], b[c(2,3)])
pto.00 = solve(A[c(3,4),], b[c(3,4)])

puntos = rbind(pto.R1R2, pto.R10, pto.R20, pto.00)
puntos = as.data.frame(puntos)
puntos
```

```
##          V1          V2
```

```
## pto.R1R2  1.333333e+00 2.666667
## pto.R10   2.000000e+00 0.000000
## pto.R20  -3.552714e-16 6.000000
## pto.00    0.000000e+00 0.000000

f_obj = function(x) 4*x$V1 + x$V2
maximo = puntos[which.max(f_obj(puntos)),]
maximo
```

```
##           V1      V2
## pto.R1R2 1.333333 2.666667
```

Grafiquemos el resultado:

```
p + geom_point(data=puntos, aes(x=V1, y=V2), col="purple", size=2.5) +
  geom_point(data=maximo, aes(x=V1, y=V2), col="purple", size=3.5)
```

