

Finding intersection point of two lines

Eyayaw Beze

10/17/2020

Contents

1	Intersection of two (straight) lines	1
1.1	Existence of intersection?	1
1.2	Find point of intersection	1

1 Intersection of two (straight) lines

1.1 Existence of intersection?

- The lines should be non-parallel, non-vertical and non-coincidental.
- If they do intersect, it's at a unique point. At the point where the two lines intersect, both lines have the same y coordinates.

1.2 Find point of intersection

Given the equations of the two lines, the point of intersection ((x, y) coordinates) of two lines can be found by simultaneously solving the two line equations—using substitution method or linear algebra.

Suppose the equations of the two lines are

$$y = ax + c \tag{1}$$

$$y = bx + d \tag{2}$$

where **a** and **b** are the **slopes** and **c** and **d** are the **y-intercepts** of the two lines.

We can rewrite them as

$$y - ax = c \tag{3}$$

$$y - bx = d \tag{4}$$

In a matrix form, we can write the system as below.

$$\overbrace{\begin{bmatrix} 1 & -a \\ 1 & -b \end{bmatrix}}^A \underbrace{\begin{pmatrix} y \\ x \end{pmatrix}}_X = \underbrace{\begin{bmatrix} c \\ d \end{bmatrix}}_D$$

Then, we can solve for \mathbf{X} by pre-multiplying both sides by \mathbf{A}^{-1} :

$$\mathbf{X} = \begin{pmatrix} y^* \\ x^* \end{pmatrix} = \mathbf{A}^{-1} \mathbf{d}$$

given $\det \mathbf{A} \neq 0$, \mathbf{A} should be **non-singular**.

For a 2x2 matrix, the *determinant* is the difference between the product of the diagonal elements, i.e., product of the main diagonal elements subtracted off the product of the **secondary diagonal** elements. Thus, in our case $\det A = (1(-b) - (-a)1)$ which simplifies to $a - b$.

Next, given the non-singularity of A , its inverse is:

$$\mathbf{A}^{-1} = \frac{1}{\det A} \begin{bmatrix} -b & a \\ -1 & 1 \end{bmatrix}.$$

Then,

$$\begin{aligned} \mathbf{X} &= \frac{1}{a-b} \begin{bmatrix} -b & a \\ -1 & 1 \end{bmatrix} \begin{bmatrix} c \\ d \end{bmatrix} \\ &= \frac{1}{a-b} \begin{bmatrix} -bc + ad \\ -c + d \end{bmatrix} \\ &= \begin{bmatrix} \frac{ad-bc}{a-b} \\ \frac{d-c}{a-b} \end{bmatrix} \end{aligned} \tag{5}$$

Therefore, the point of intersection is $x = \frac{d-c}{a-b}$ and $y = \frac{ad-bc}{a-b}$ or $y = a \frac{d-c}{a-b} + c$ by (1).

```
# Given equations of the two lines as:
# y = ax+c
# y = bx+d

lines_intersect <- function(a, c, b, d) {
  # A %% X = D
  # where A is a 2x2 coefficient matrix
  # X = a 2x1 variable vector
  # a 2x1 constant vector
  if (a==b) {
    message("system is exactly singular: det A = 0")
    message("no intersection")
    return(NULL)
  }
  else {
    A <- matrix(c(1, -a, 1, -b), nrow = 2, ncol = 2, byrow = TRUE)
    D <- matrix(c(c, d), nrow = 2, ncol = 1, byrow = TRUE)
    sol <- solve(A, D) # (y, x)
    rev(sol) # solution as (x, y)
  }
}
```

Note if $a = b$ it means the two lines are parallel, hence no intersection. If $c \neq d$ as well, the lines are different and there is no intersection, otherwise the two lines are identical.

```
plot_intersection = function(a, c, b, d, xlim = c(-10, 10), gg) {

  f1 <- function(x) a * x + c
  f2 <- function(x) b * x + d
  xll <- xlim[[1]] # x-axis lower limit
  xul <- xlim[[2]] # x-axis upper limit
  yll <- min(f1(xll), f2(xll), f1(xul), f2(xul)) # y-axis lower limit
  yul <- max(f1(xll), f2(xll), f1(xul), f2(xul)) # y-axis upper limit

  # create equation labels
  equ_label <- function() {
    slope = c(a, b)
    paste0("y = ", ifelse(slope == 1, "", slope), "x + ", c(c, d))
  }
  labs = equ_label()
  p = lines_intersect(a, c, b, d) # point of intersection
  lab.p = paste0("(", paste(round(p, 2), collapse = ", "), ")")

  if (missing(gg)) gg = ggplot()

  gg +
    geom_function(fun = f1, col = "red") +
    geom_label(aes(x = xul/2, y = f1(xul/2), label = labs[1]), col = "red") +
    geom_function(fun = f2, col = "blue") +
    geom_label(aes(x = -xul/2, y = f2(-xul/2), label = labs[2]), col = "blue") +
    geom_point(aes(x = p[1], y = p[2])) +
    geom_label(aes(x = p[1], y = p[2]), label = lab.p) +
    #lims(x = c(xll, xul), y = c(yll, yul)) +
    labs(x = "x", y = "y")
}
```

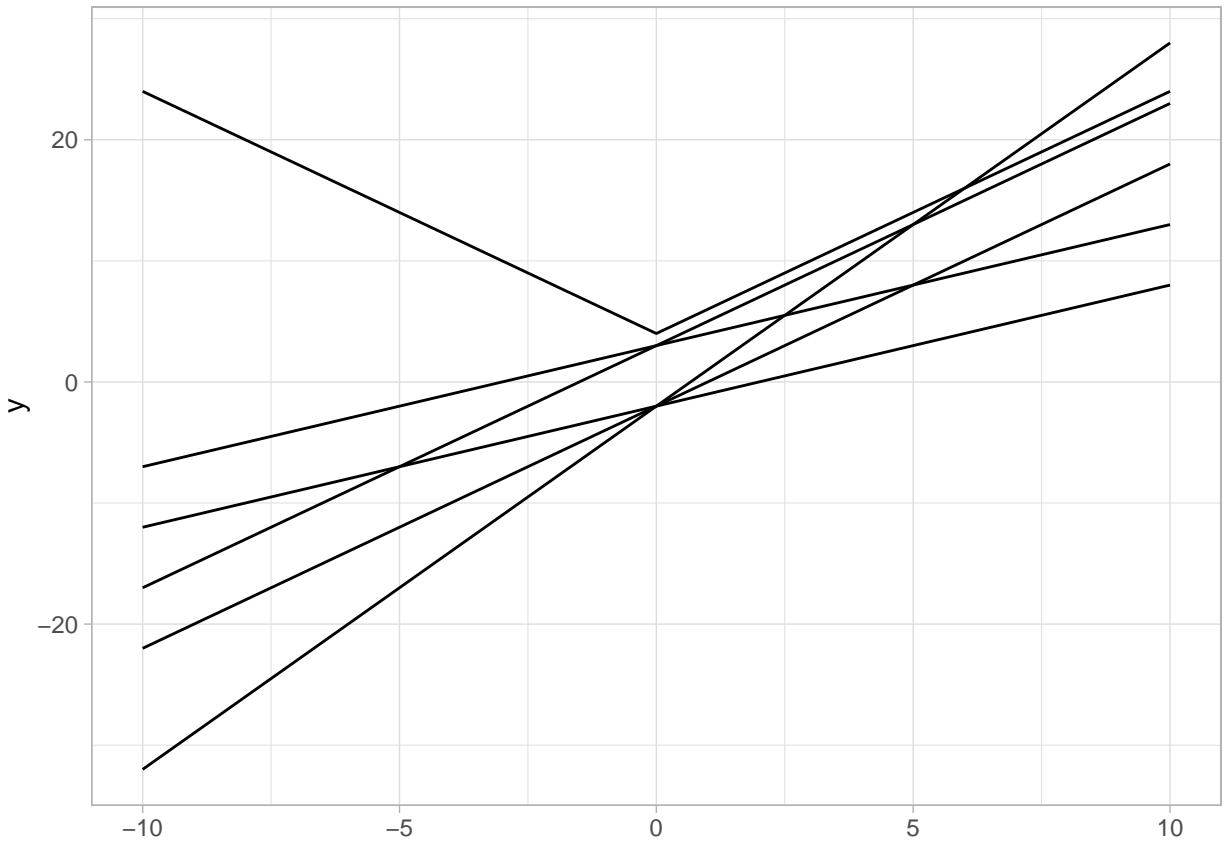
Let's try to answer a question asked on twitter

```
eqns <- list(
  main.fun = function(x) abs(2 * x) + 4,
  A = function(x) x - 2,
  B = function(x) x + 3,
  C = function(x) 2 * x - 2,
  D = function(x) 2 * x + 3,
  E = function(x) 3 * x - 2
)

params = data.frame(slope = c(1, 1, 2, 2, 3),
  yintercept = c(-2, 3, -2, 3, -2))
```

```
pp = ggplot() +
  geom_function(fun = eqns$main.fun) +
  lims(x = c(-10, 10))

pp +
  lapply(eqns[-1], function(f) geom_function(fun = f))
```



```
# solution
Map(lines_intersect,
  a = 2, c = 4,
  b = params$slope,
  d = params$yintercept) # case 1:  $y = 2x + 4$ , if  $x > 0$ 
```

```
## system is exactly singular: det A = 0
```

```
## no intersection
```

```
## system is exactly singular: det A = 0
```

```
## no intersection
```

```
## [[1]]
## [1] -6 -8
##
## [[2]]
## [1] -1 2
##
## [[3]]
## NULL
##
## [[4]]
```

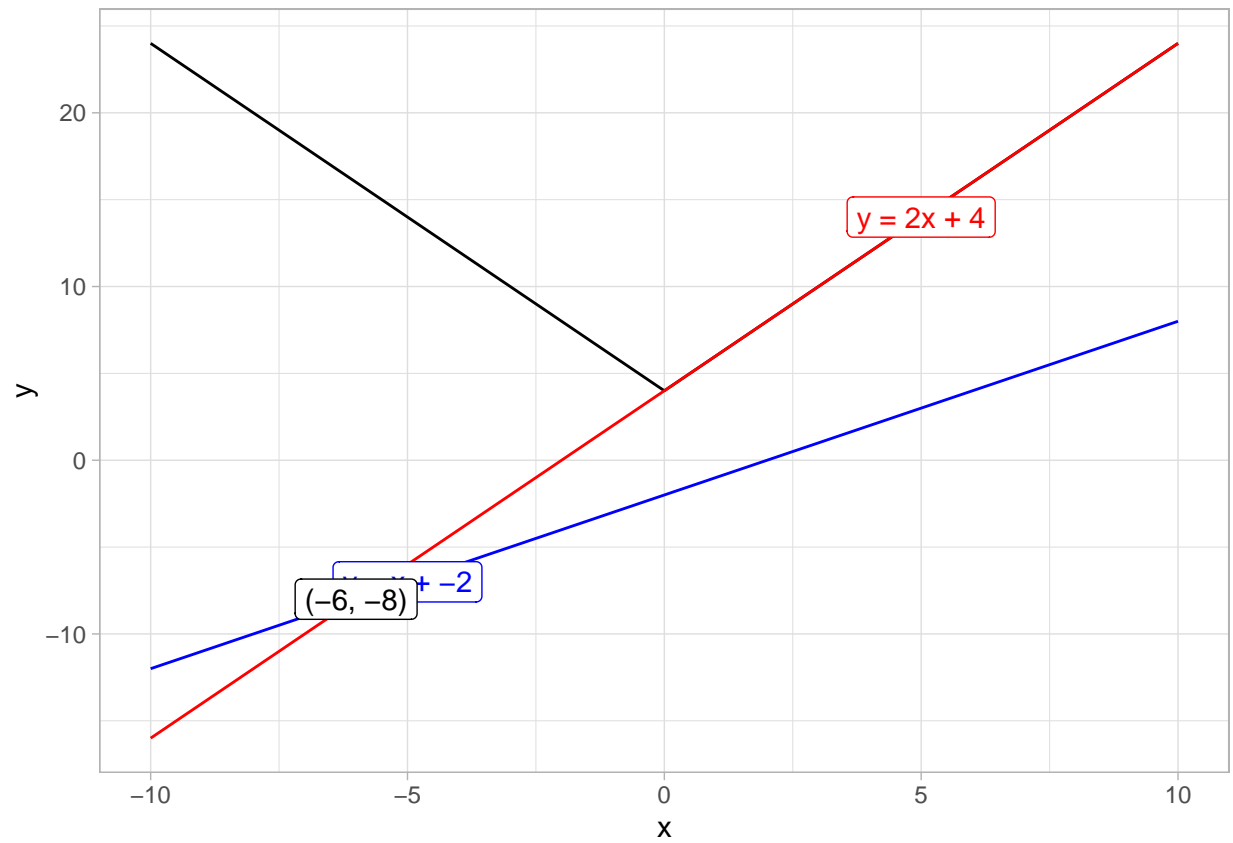
```
## NULL
##
## [[5]]
## [1] 6 16

Map(lines_intersect, a = -2, c = 4,
b = params$slope,
d = params$yintercept) # case 2:  $y = -2x+4$ , if  $x < 0$ 
```

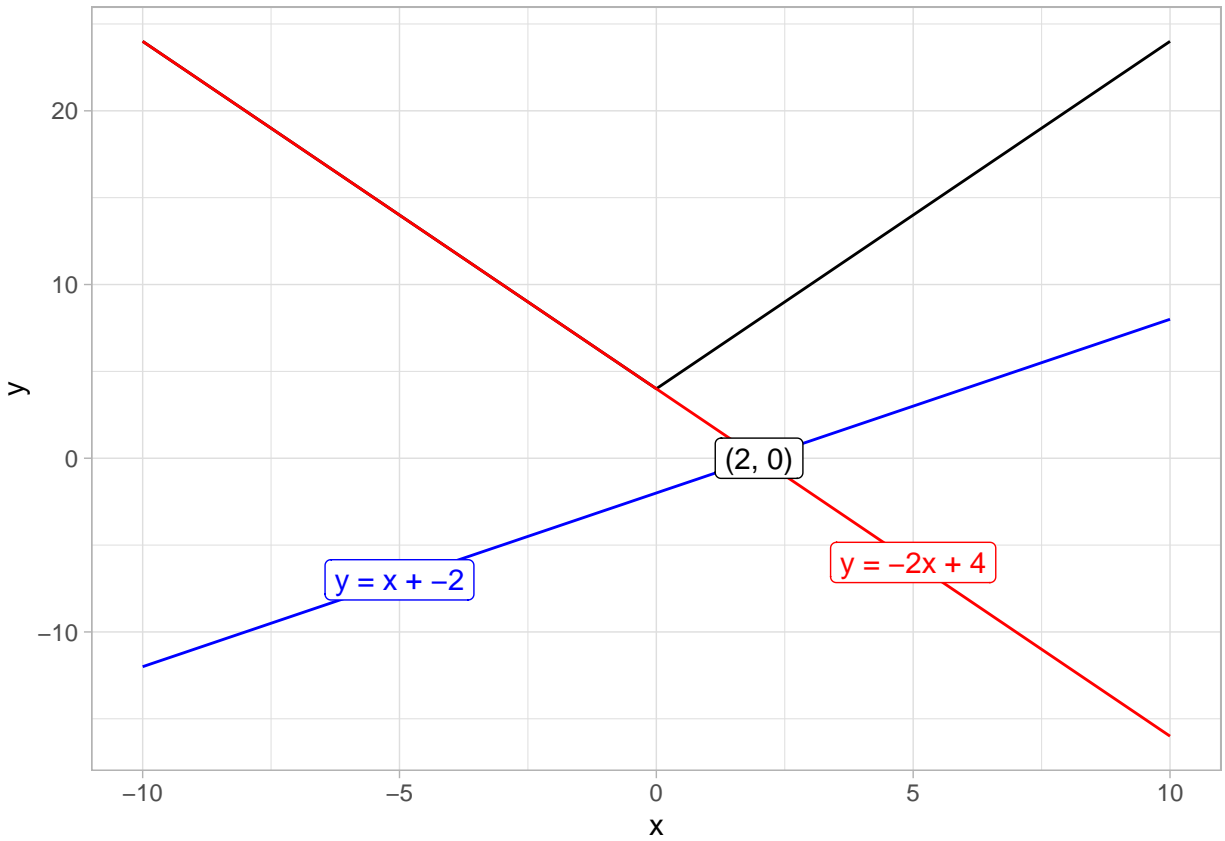
```
## [[1]]
## [1] 2 0
##
## [[2]]
## [1] 0.3333333 3.3333333
##
## [[3]]
## [1] 1.5 1.0
##
## [[4]]
## [1] 0.25 3.50
##
## [[5]]
## [1] 1.2 1.6
```

```
# main equation:  $y = \text{abs}(2*x) + 4$ 
## case 1
a = 2; c = 4
## case 2
a. = -2; c. = 4

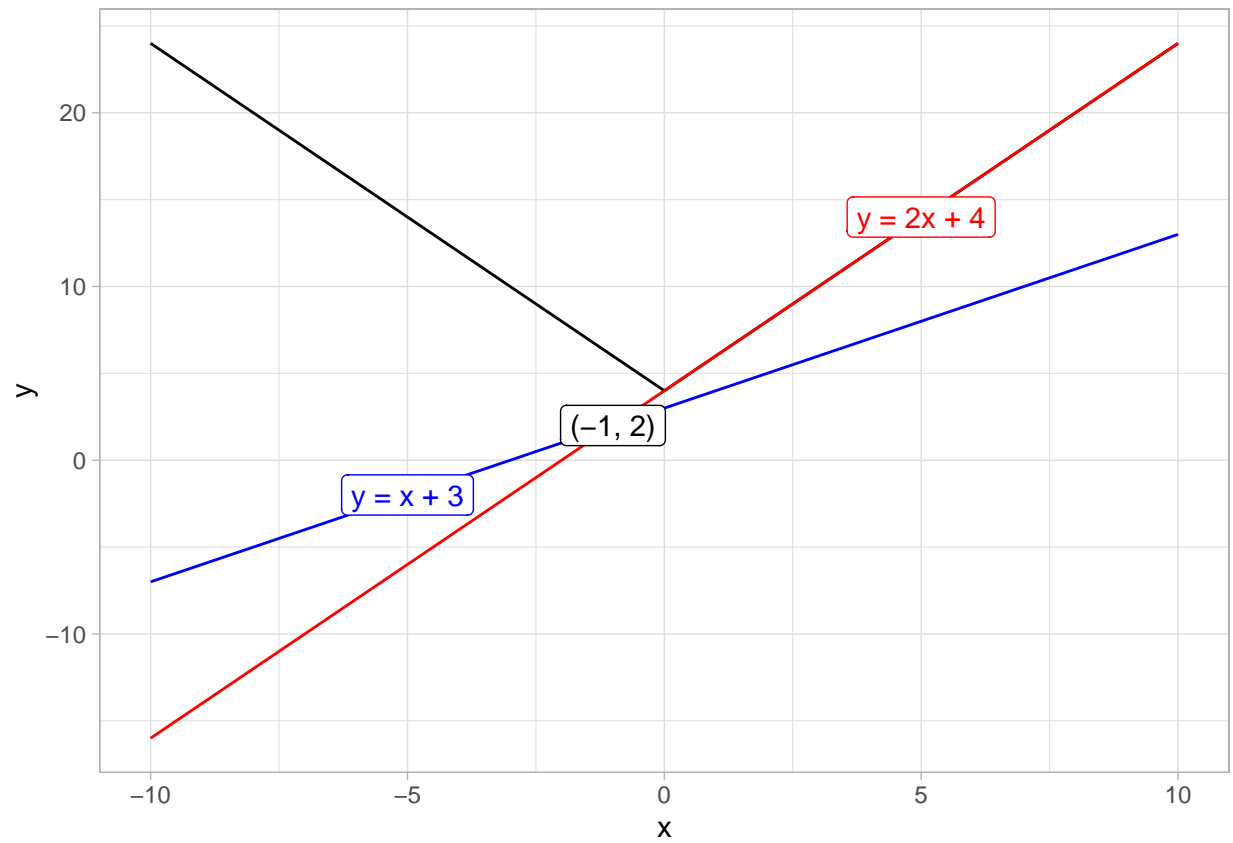
# choice A ->  $y = x-2$ 
b = 1; d = -2
plot_intersection(a, c, b, d, gg = pp)
```



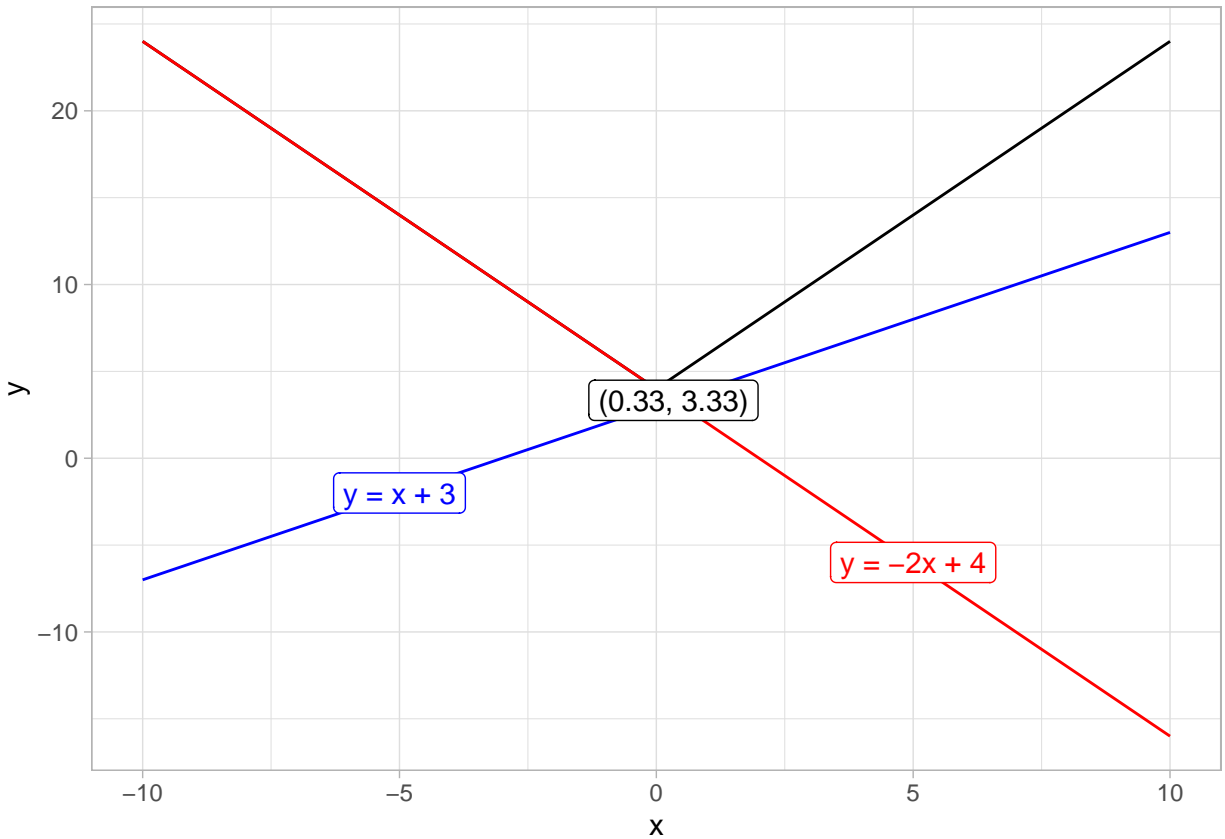
```
plot_intersection(a., c., b, d, gg = pp)
```



```
# choice B ->  $y = x + 3$   
b = 1; d = 3  
plot_intersection(a, c, b, d, gg = pp)
```



```
plot_intersection(a., c., b, d, gg = pp)
```

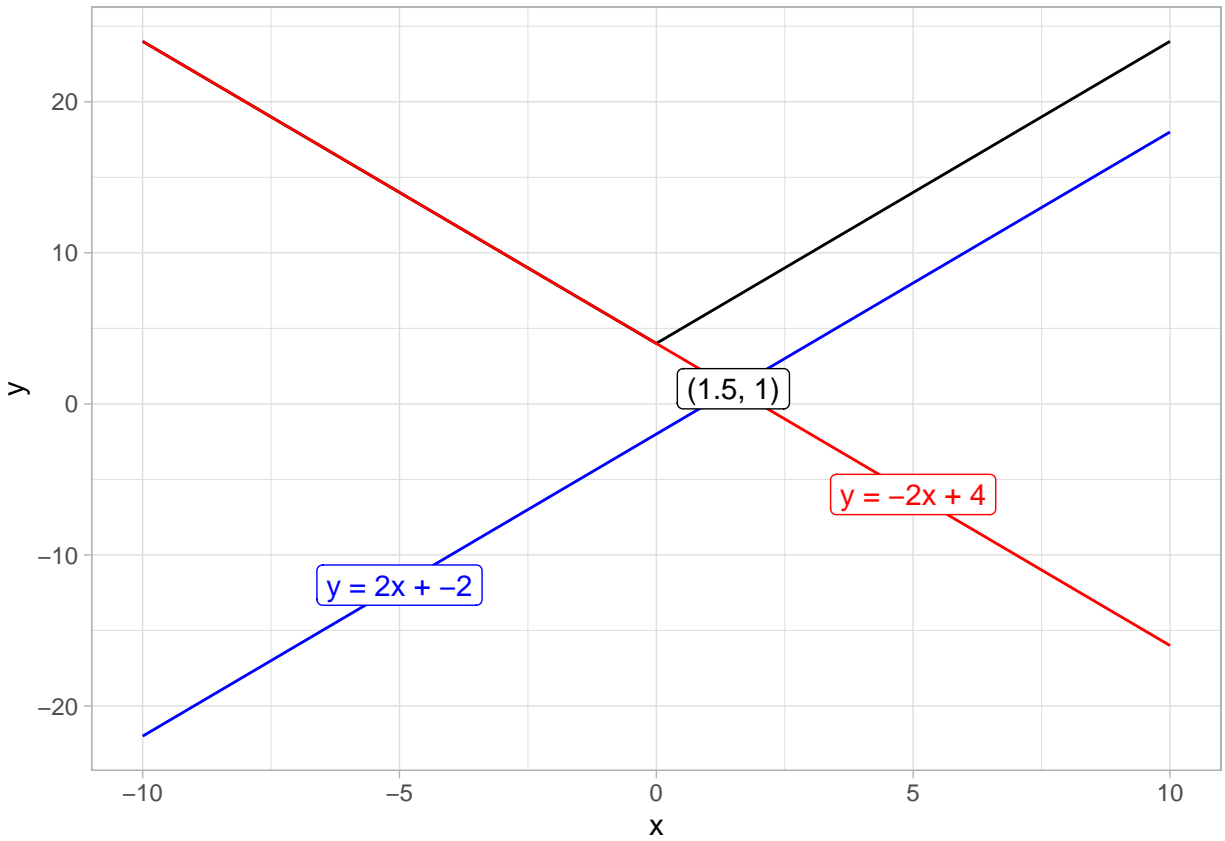



```
# C -> y = 2 * x - 2
b = 2; d = -2
plot_intersection(a, c, b, d, gg = pp)
```

```
## system is exactly singular: det A = 0
## no intersection
```

```
## Error in round(p, 2): non-numeric argument to mathematical function
```

```
plot_intersection(a., c., b, d, gg = pp)
```

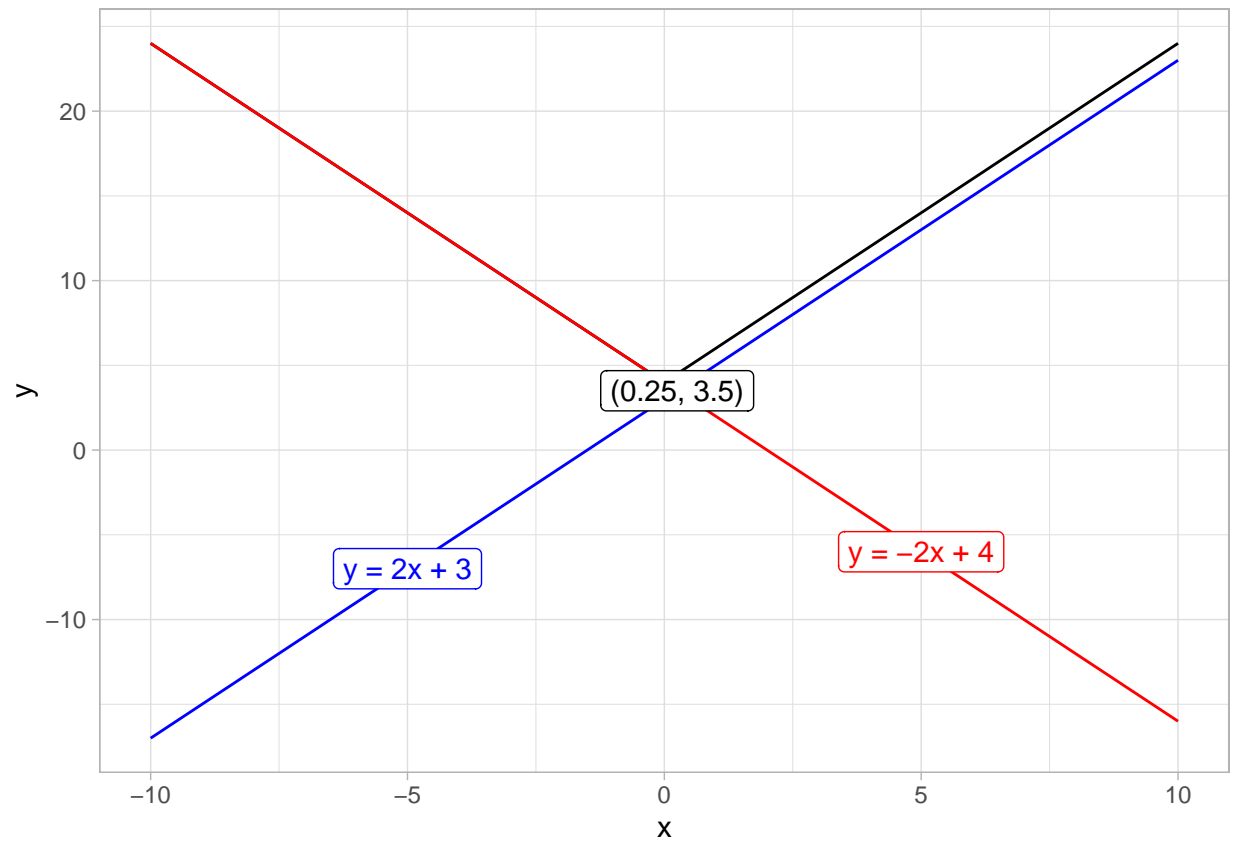


```
# D -> y = 2 * x + 3
b = 2; d = 3
plot_intersection(a, c, b, d, gg = pp)
```

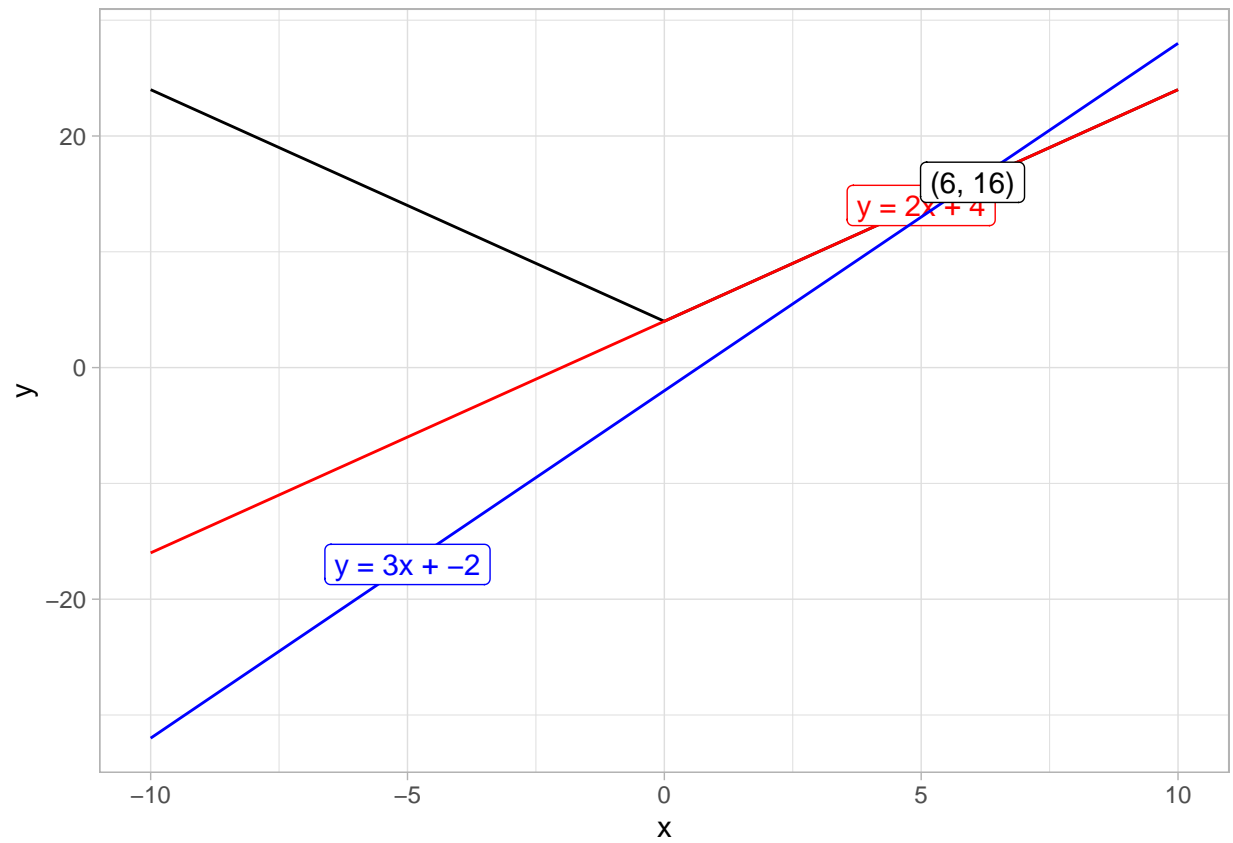
```
## system is exactly singular: det A = 0
## no intersection
```

```
## Error in round(p, 2): non-numeric argument to mathematical function
```

```
plot_intersection(a., c., b, d, gg = pp)
```



```
# E  $y = 3 * x - 2$   
b = 3; d = -2  
plot_intersection(a, c, b, d, gg = pp)
```



```
plot_intersection(a., c., b, d, gg = pp)
```

