R Notebook

Alonso Melgar López

Ecuación de la recta

$$y = mx + b$$

Cargamos librerias

```
#install.packages("tidyverse")
library(tidyverse)
```

```
## -- Attaching packages ---------------------------------------------------------------------------
```

```
## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.3     v dplyr   1.0.2
## v tidyr   1.1.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0
```

```
## Warning: package 'ggplot2' was built under R version 4.0.2
```

```
## Warning: package 'tibble' was built under R version 4.0.2
```

```
## Warning: package 'dplyr' was built under R version 4.0.2
```

```
## -- Conflicts ------------------------------------------------------------------------------------
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
#library(dplyr)
```

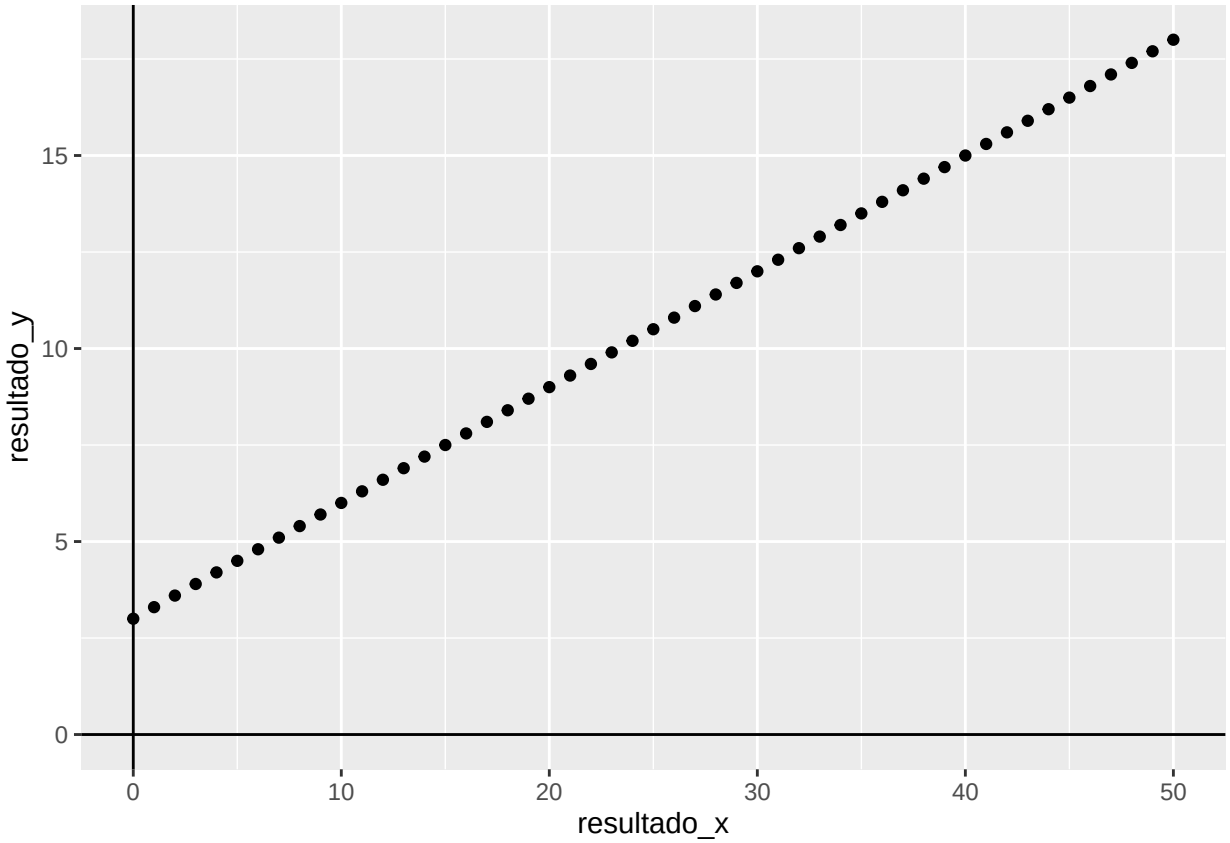
```
# creamos una función que calcula los valores de y según ciertos valores
ecu_recta <- function(m, b, x) {

  y <- m * x + b

  return(data.frame( resultado_x = x, resultado_y = y ))

}
```

```r
x <- 0:50

recta <- ecu_recta(m=.3, b=3, x=x)
recta
```

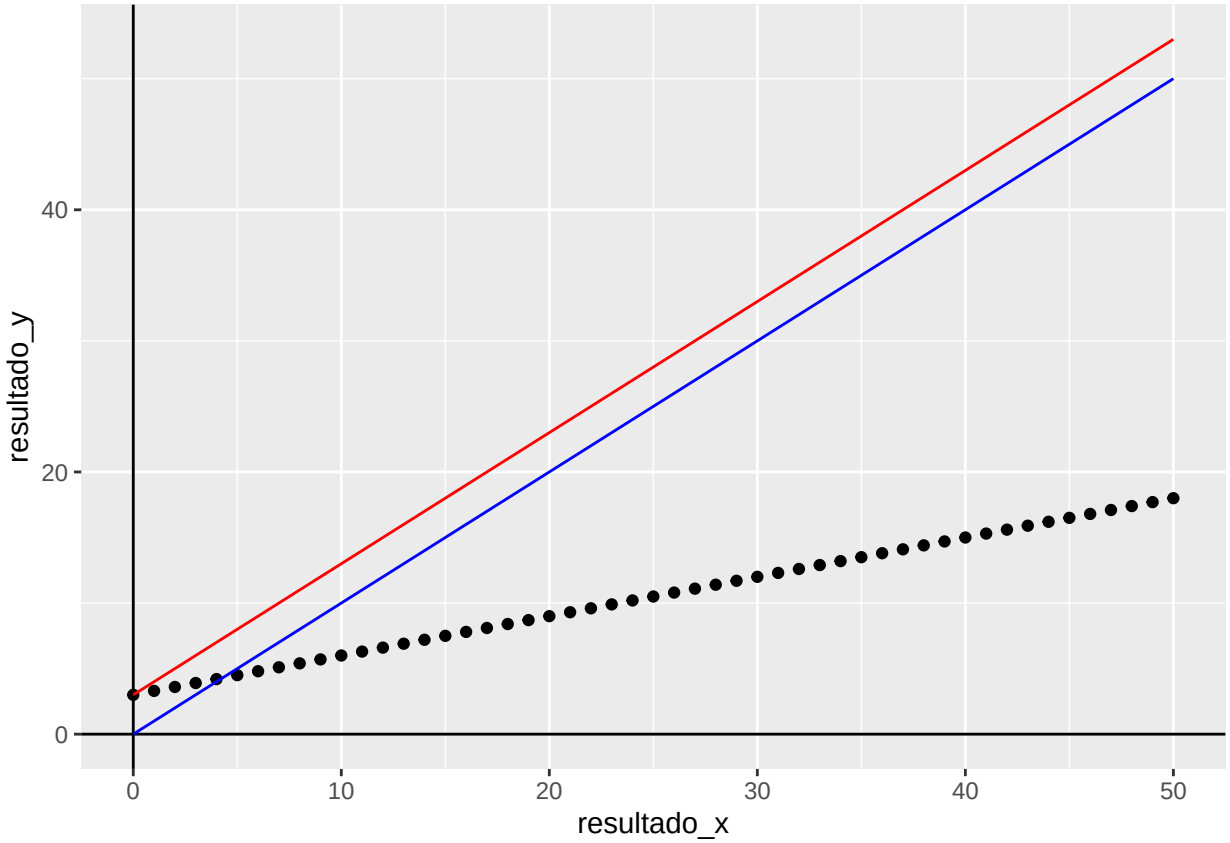
```
##    resultado_x resultado_y
## 1            0         3.0
## 2            1         3.3
## 3            2         3.6
## 4            3         3.9
## 5            4         4.2
## 6            5         4.5
## 7            6         4.8
## 8            7         5.1
## 9            8         5.4
## 10           9         5.7
## 11          10         6.0
## 12          11         6.3
## 13          12         6.6
## 14          13         6.9
## 15          14         7.2
## 16          15         7.5
## 17          16         7.8
## 18          17         8.1
## 19          18         8.4
## 20          19         8.7
## 21          20         9.0
## 22          21         9.3
## 23          22         9.6
## 24          23         9.9
## 25          24        10.2
## 26          25        10.5
## 27          26        10.8
## 28          27        11.1
## 29          28        11.4
## 30          29        11.7
## 31          30        12.0
## 32          31        12.3
## 33          32        12.6
## 34          33        12.9
## 35          34        13.2
## 36          35        13.5
## 37          36        13.8
## 38          37        14.1
## 39          38        14.4
## 40          39        14.7
## 41          40        15.0
## 42          41        15.3
## 43          42        15.6
## 44          43        15.9
## 45          44        16.2
## 46          45        16.5
## 47          46        16.8
## 48          47        17.1
```

```
## 49          48        17.4
## 50          49        17.7
## 51          50        18.0
```

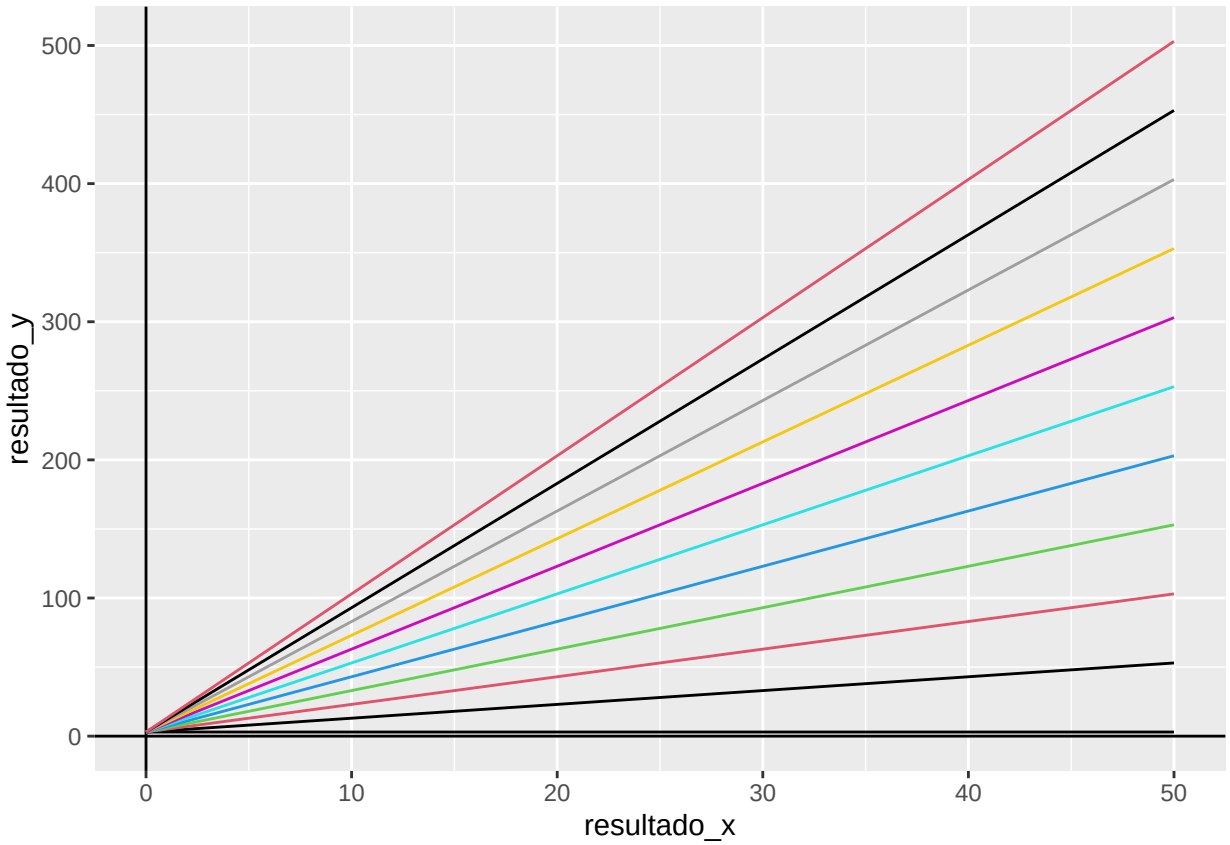
```
#graficamos
p <- ggplot() +
  geom_point(
    data = recta,
    aes(x=resultado_x, y=resultado_y)
             ) +
  geom_vline(xintercept = 0) +
  geom_hline(yintercept = 0)

p
```



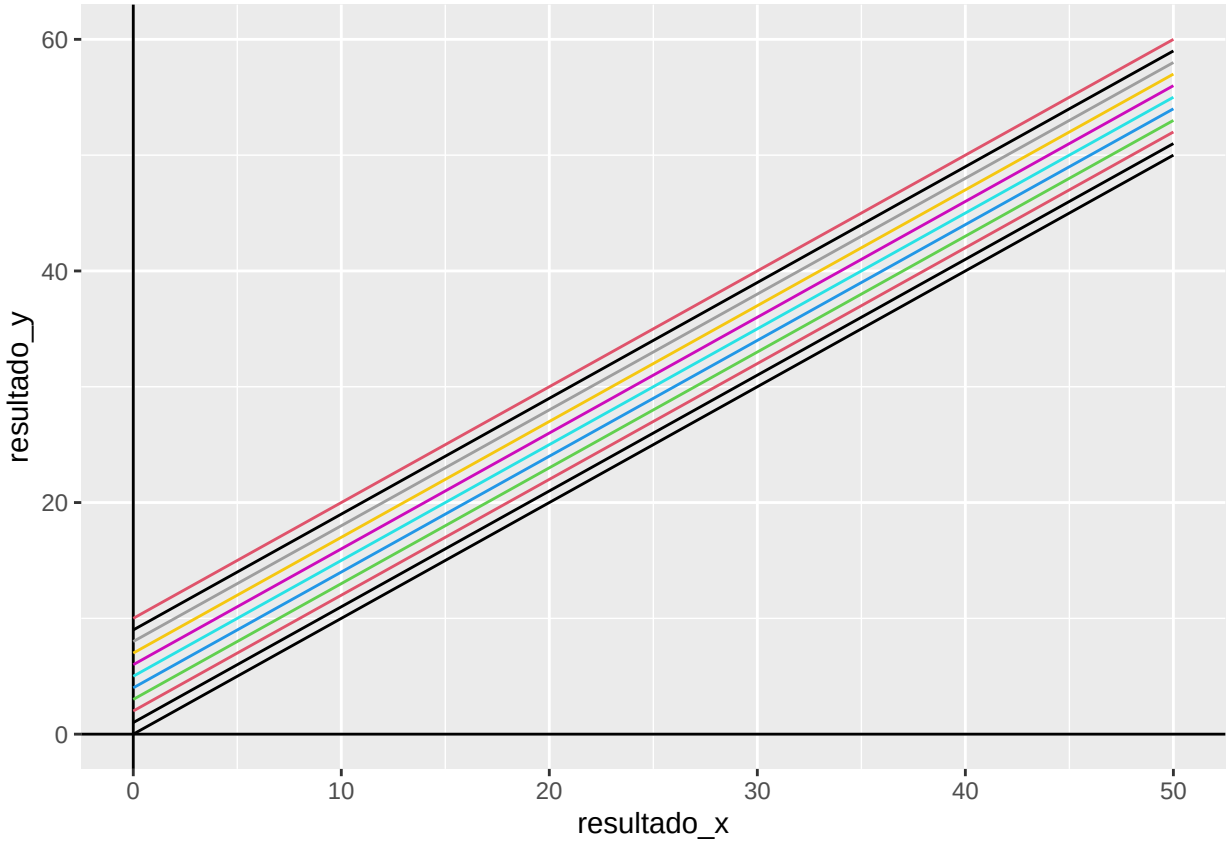

```
# Agregamos distintas rectas
recta_2 <- ecu_recta(m=1, x=x, b=3)
recta_3 <- ecu_recta(m=2, x=x, b=3)
p +
  geom_line(data = recta_2, aes(x=resultado_x,y=resultado_y), color = "red") +
  geom_line(data = recta_3, aes(x=resultado_x,y=resultado_x), color = "blue" )
```




```
x <- 0:50
recta <- ecu_recta(m=0, x=x, b=3)

p <- ggplot() +
  geom_line(data = recta, aes(x=resultado_x, y = resultado_y)) +
  geom_vline(xintercept = 0) +
  geom_hline(yintercept = 0)

# Agregamos multiples rectas
for (i in 1:10) {
  new_recta <- ecu_recta(m=i, x=x, b=3)

  p <- p +
    geom_line(data = new_recta,
              aes(x=resultado_x, y=resultado_y),
              color = i)
}
p
```




```
x <- 0:50
recta <- ecu_recta(m=1, x=x, b=0)

p <- ggplot() +
  geom_line(data = recta, aes(x=resultado_x, y = resultado_y)) +
  geom_vline(xintercept = 0) +
  geom_hline(yintercept = 0)

for (i in 1:10) {
  new_recta <- ecu_recta(m=1, x=x, b=i)

  p <- p +
    geom_line(data = new_recta,
              aes(x=resultado_x, y=resultado_y),
              color = i)
}
p
```



Segundo grado

$$ax^2 + bx + c = 0$$

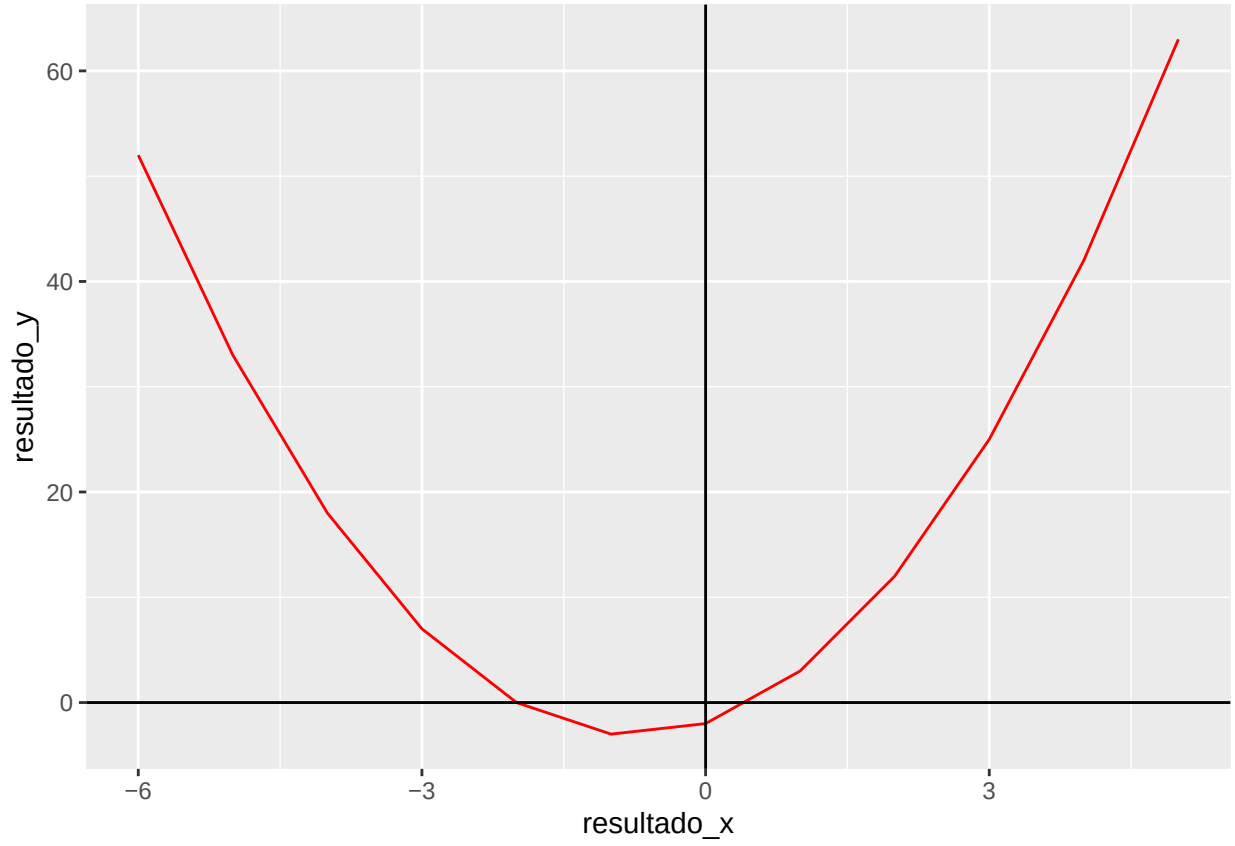
```
calcula_segundo_grado <- function(x, a, b, c) {

  y <- a * x^2 + b * x + c
  curva <- data.frame(resultado_x=x, resultado_y=y)

  return(curva)
}
```

```
x <- -6:5

curva_resultado <- calcula_segundo_grado(x=x, a=2, b=3, c=-2)
curva_resultado
```

```
##    resultado_x resultado_y
## 1           -6          52
## 2           -5          33
## 3           -4          18
## 4           -3           7
## 5           -2           0
```

```
## 6           -1          -3
## 7            0          -2
## 8            1           3
## 9            2          12
## 10           3          25
## 11           4          42
## 12           5          63
```

```
ggplot() +
  geom_line(data = curva_resultado,
            aes(x=resultado_x, y=resultado_y),
            color = "red"
            ) +
  geom_vline(xintercept = 0) +
  geom_hline(yintercept = 0)
```



Raices para ecuaciones de segundo grado

$$\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

```
raices <- function(b, a, c) {
  raiz <- sqrt(b^2 - (4 * a * c))
```

```
  primer_x <- (-b + raiz)/(2 * a)
  segunda_x <- (-b - raiz)/(2 * a)

  return(c(primer_x, segunda_x))
}
```

```
raices(a=2, b=3, c=-2)
```

```
## [1]  0.5 -2.0
```

```
# Unimos lo anterior en una sola función
calcula_segundo_grado <- function(x, a, b, c, raiz = FALSE) {

  if(raiz) {
    raices_resultado <- raices(b, a, c)
    return(raices_resultado)

  } else {

    y <- a * x^2 + b * x + c
    curva <- data.frame(resultado_x=x, resultado_y=y)

    return(curva)
  }

}
```

```
calcula_segundo_grado(x=x,a=1,b=1,c=-3, raiz = TRUE)
```

```
## [1]  1.302776 -2.302776
```