

# Introdução ao programa R

Ronald Targino, DEMA-UFC

Notas de aula

## 5. Gráficos - Parte 1

```
# A função f1 retorna o quadrado do seu argumento.
```

```
f1 = function(x) x^2
```

```
# Calcular f1 para vários valores de x
```

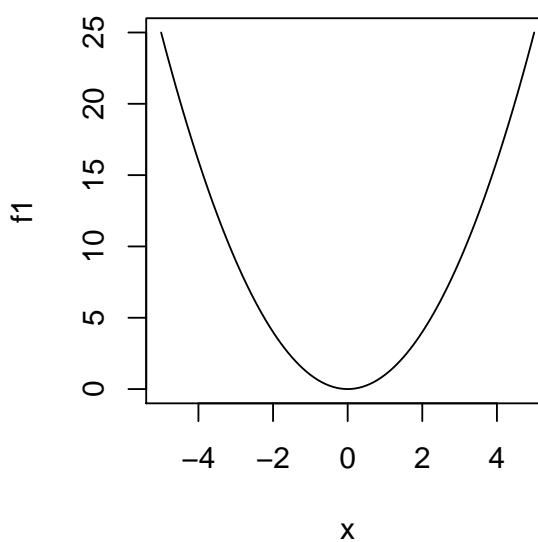
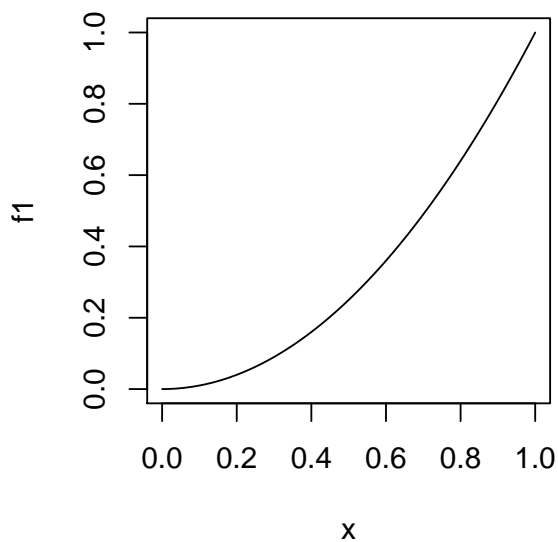
```
f1(-5:5)
```

```
## [1] 25 16 9 4 1 0 1 4 9 16 25
```

```
old.par = par(mfrow = c(1, 2)) # configurar a janela gráfica (1 linha e 2 colunas)
```

```
plot(f1) # padrão: eixo x no intervalo (0,1)
```

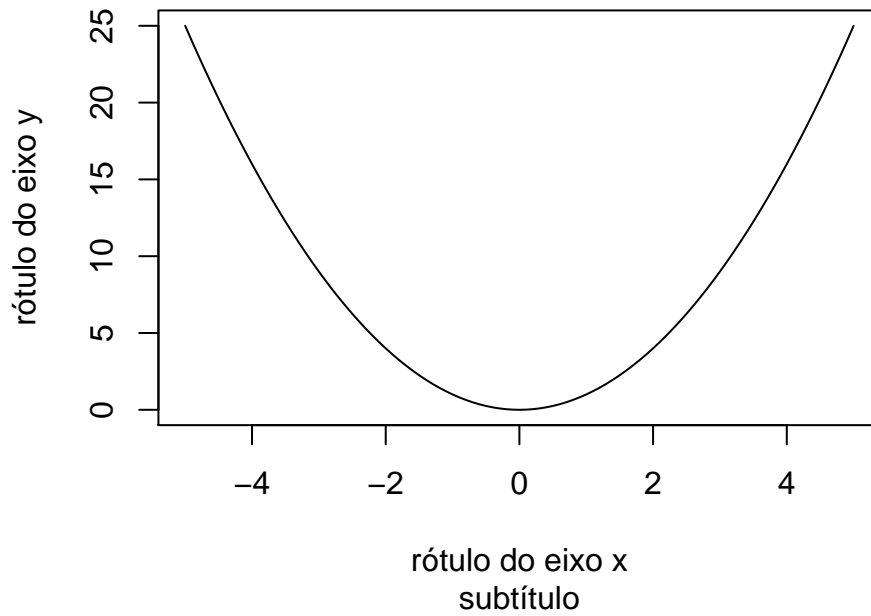
```
plot(f1, xlim = c(-5, 5)) # alterar o intervalo do eixo x
```



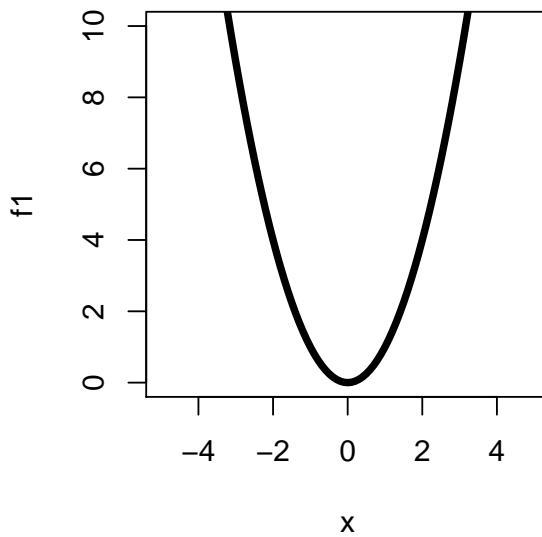
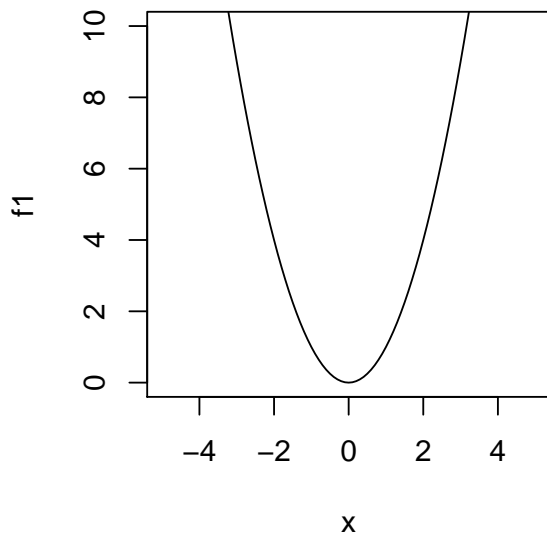
```
par(old.par) # retomar a configuração padrão da janela gráfica (1 linha e 1 coluna)
```

```
plot(f1, xlim = c(-5, 5), main = "Título", sub = "subtítulo", xlab = "rótulo do eixo x",  
     ylab = "rótulo do eixo y")
```

## Título



```
old.par = par(mfrow = c(1, 2)) # configurar a janela gráfica (1 linha e 2 colunas)
plot(f1, xlim = c(-5, 5), ylim = c(0, 10)) # indicar os limites dos eixos x e y
plot(f1, xlim = c(-5, 5), ylim = c(0, 10), lwd = 4) # alterar a largura da linha
```

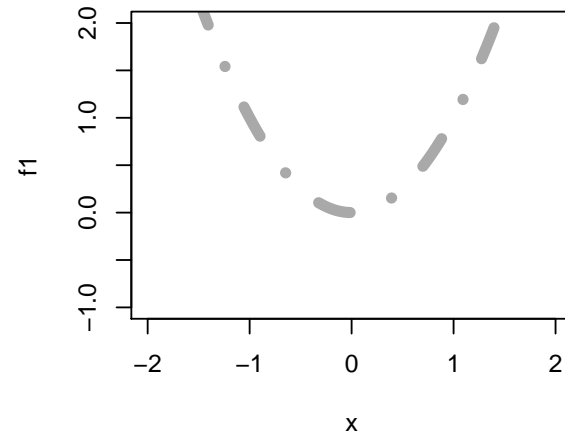
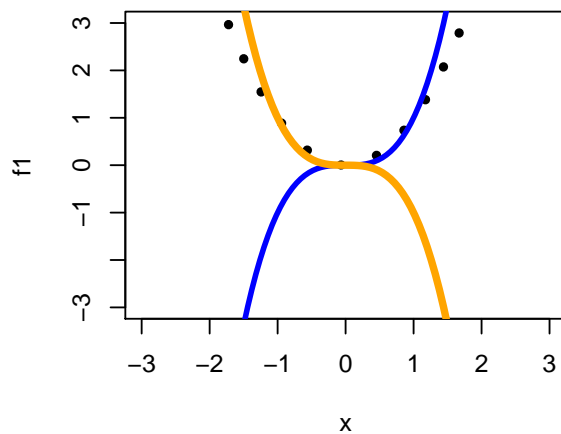
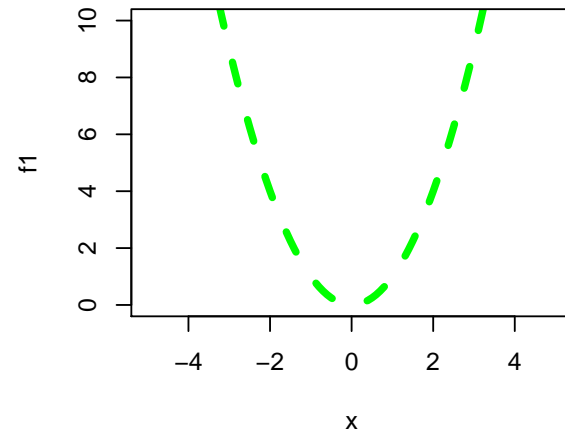
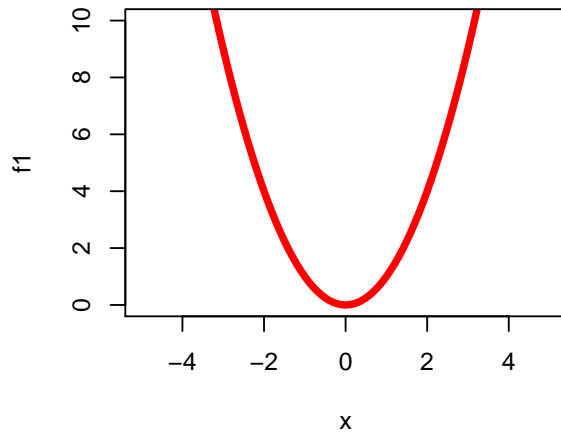


```
par(old.par) # retomar a configuração padrão da janela gráfica
```

```

old.par = par(mfrow = c(2, 2)) # configurar a janela gráfica (2 linhas e 2 colunas)
f1 = function(x) x^2
# primeiro gráfico: alterando o estilo (lty) e a cor (col) da linha
plot(f1, xlim = c(-5, 5), ylim = c(0, 10), lwd = 4, lty = 1, col = "red")
# segundo gráfico
plot(f1, xlim = c(-5, 5), ylim = c(0, 10), lwd = 4, lty = 2, col = "green")
# terceiro gráfico
plot(f1, xlim = c(-3, 3), ylim = c(-3, 3), lwd = 5, lty = 3, col = "black")
# as curvas para as funções f2 e f3 são adicionadas ao terceiro gráfico
f2 = function(x) x^3
f3 = function(x) (-x)^3
curve(f2, add = TRUE, lwd = 3, col = "blue")
curve(f3, add = TRUE, lwd = 4, col = "orange")
# quarto gráfico
plot(f1, xlim = c(-2, 2), ylim = c(-1, 2), lwd = 6, lty = 4, col = "darkgray")

```



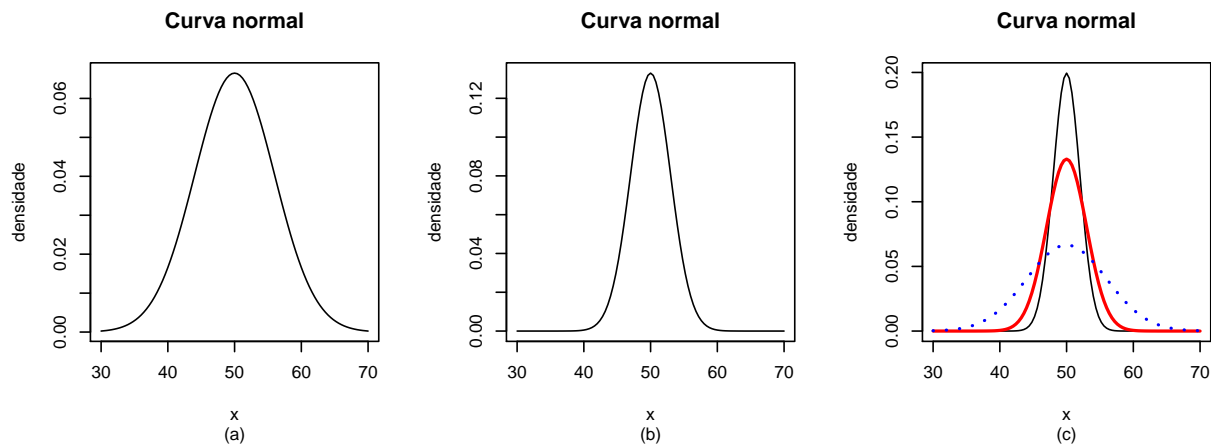
```
par(old.par) # retomar a configuração padrão da janela gráfica
```

A função definida abaixo é uma importante função na área de estatística. Essa função é denominada densidade de probabilidade da distribuição normal (ou Gaussiana) de média  $\mu$  e desvio padrão  $\sigma$ . Sua forma é de um sino, sendo simétrica em relação ao valor  $\mu$  e com abertura associada ao valor  $\sigma$ . No R, essa densidade de probabilidade é disponibilizada pela função `dnorm` com a seguinte sintaxe: `dnorm(x, média, desvio padrão)`.

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right), \quad x \in \mathbb{R}, \mu \in \mathbb{R}, \sigma > 0.$$

```
# Funções densidade da distribuição normal de média 50 e diferentes desvios
# padrões
f1 = function(x) dnorm(x, 50, 6)
f2 = function(x) dnorm(x, 50, 3)
f3 = function(x) dnorm(x, 50, 2)

old.par = par(mfrow = c(1, 3)) # configurar a janela gráfica (1 linha e 3 colunas)
curve(f1, xlim = c(30, 70), ylab = "densidade", main = "Curva normal", sub = "(a)")
curve(f2, xlim = c(30, 70), ylab = "densidade", main = "Curva normal", sub = "(b)")
curve(f3, xlim = c(30, 70), ylab = "densidade", main = "Curva normal", sub = "(c)")
curve(f2, add = TRUE, col = "red", lwd = 2, lty = 1)
curve(f1, add = TRUE, col = "blue", lwd = 2, lty = 3)
```



```
par(old.par) # retomar a configuração padrão da janela gráfica
```

```
# f4 é a função densidade normal padrão, isto é, de média 0 (zero) e desvio
# padrão 1 (um)
f4 = function(x) (1/sqrt(2 * pi)) * exp(-0.5 * x^2)
# Obtendo valores de f4
f4(0)
```

```
## [1] 0.3989423
```

```
f4(c(-2, -1, 0, 1, 2))
```

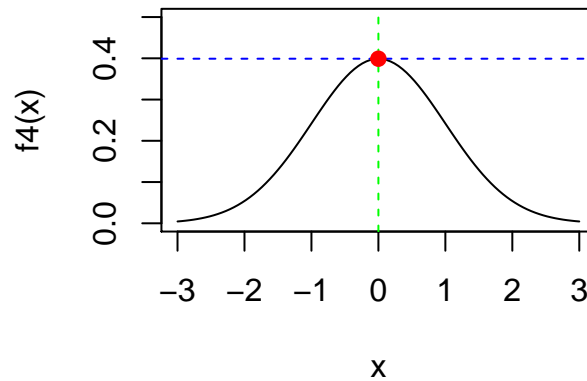
```
## [1] 0.05399097 0.24197072 0.39894228 0.24197072 0.05399097
```

```
# arredondando os valores
```

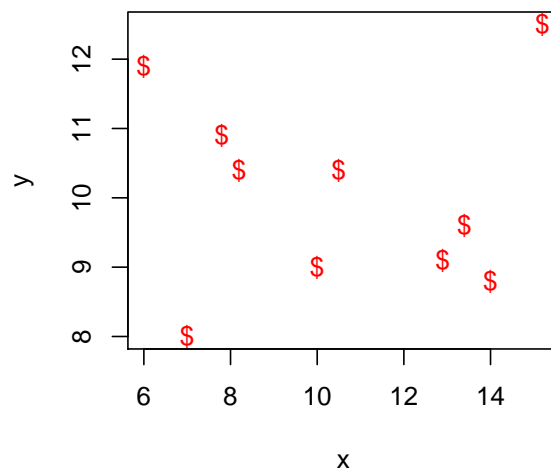
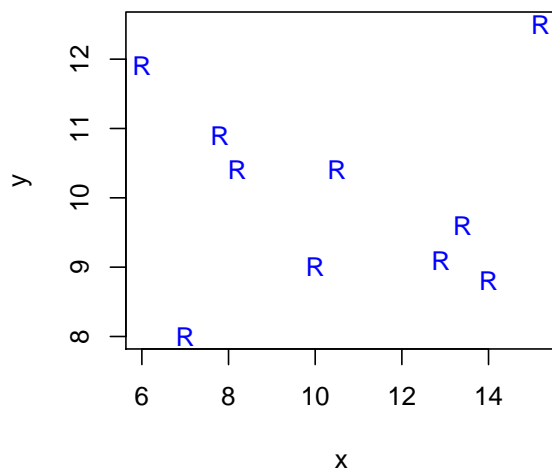
```
round(f4(c(-2, -1, 0, 1, 2)), 3)
```

```
## [1] 0.054 0.242 0.399 0.242 0.054
```

```
# Gráfico de f4
curve(f4, xlim = c(-3, 3), ylim = c(0, 0.5))
# adicionando linhas e ponto no gráfico
abline(v = 0, lty = 2, col = "green") # linha vertical no valor v=0
abline(h = 0.399, lty = 2, col = "blue") # linha horizontal no valor h=0.399
points(0, 0.399, pch = 19, col = "red") # ponto na coordenada(0, 0.399)
```

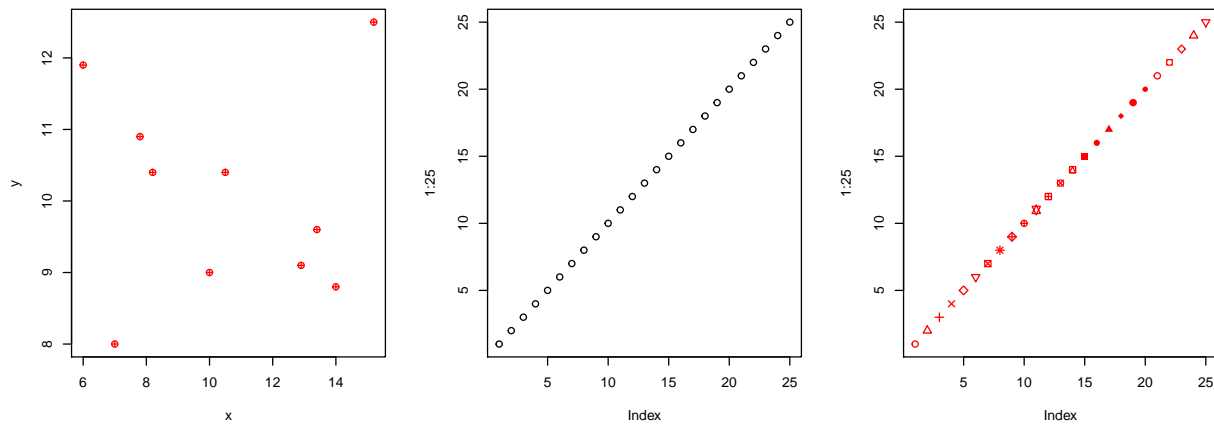


```
# Marcadores dos pares ordenados
old.par = par(mfrow = c(1, 2)) # configurar a janela gráfica (1 linha e 2 colunas)
x = c(7, 10, 15.2, 7.8, 14, 10.5, 12.9, 6, 8.2, 13.4)
y = c(8, 9, 12.5, 10.9, 8.8, 10.4, 9.1, 11.9, 10.4, 9.6)
plot(x, y, pch = "R", col = "blue") # pch: caractere usado como marcador
plot(x, y, pch = "$", col = "red")
```



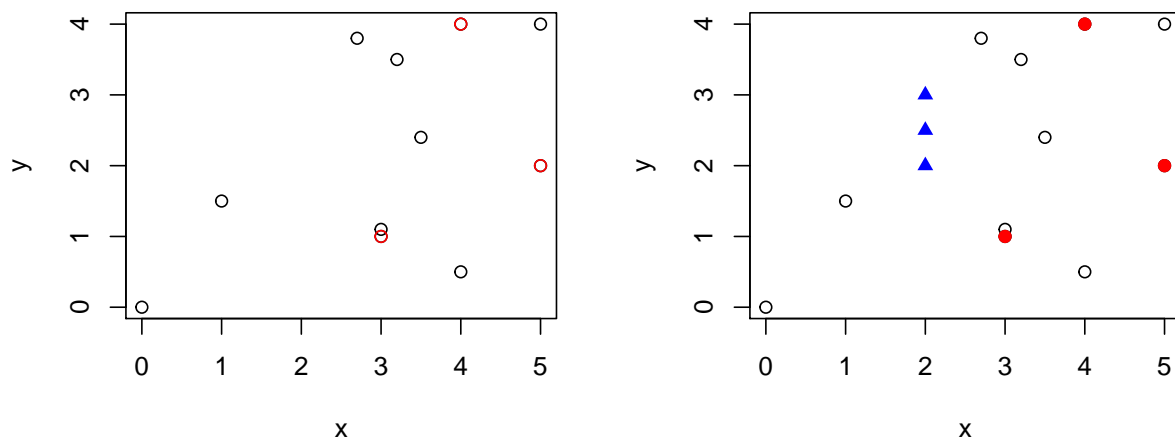
```
par(old.par)
```

```
# Marcadores dos pares ordenados
old.par = par(mfrow = c(1, 3)) # configurar a janela gráfica (1 linha e 2 colunas)
plot(x, y, pch = 10, col = "red") # cartere pode ser indicado por um número de 1 a 25
plot(1:25)
plot(1:25, pch = 1:25, col = "red")
```



```
par(old.par)
```

```
old.par = par(mfrow = c(1, 2)) # configurar a janela gráfica (1 linha e 2 colunas)
x = c(4, 3, 5, 3.2, 5, 4, 3.5, 2.7, 3, 0, 1)
y = c(4, 1, 2, 3.5, 4, 0.5, 2.4, 3.8, 1.1, 0, 1.5)
plot(x, y) # gráfico dos pares ordenados
points(c(4, 3, 5), c(4, 1, 2), col = "red") # colorir os pontos (4,4), (3,1) e (5,2)
plot(x, y)
points(c(4, 3, 5), c(4, 1, 2), pch = 19, col = "red") # trocar o marcador
points(c(2, 2, 2), c(2, 2.5, 3), pch = 17, col = "blue") # adicionar os pontos (2,2),
```



*# (2,2.5) e (2,3) usando outro marcador e cor.*

`par(old.par)`

*# Para desenhar polígonos podemos usar a função polygon*

`old.par = par(mfrow = c(2, 2))`

*# gráfico 1*

`plot(c(0, 0, 5, 5), c(1, 4, 1, 4), xlab = "gráfico 1", ylab = "")` *# gerar gráfico*

*# destacando os pares ordenados (0,1), (0,4), (5,1) e (5,4). gráfico 2*

`plot(c(0, 0, 5, 5), c(1, 4, 1, 4), type = "n", xlab = "gráfico 2", ylab = "")` *# mesmo*

*# gráfico acima sem os pontos*

`lines(c(0, 2), c(1, 3), col = "red")` *# adicionar linha entre os pontos (0,1) e (2,3)*

`lines(c(1, 5, 4), c(1.5, 2, 3.5), col = "blue", lwd = 3)` *# adicionar linha passando*

*# pelos pontos (1,1.5), (5,2) e (4,3.5) gráfico 3*

`plot(c(0, 3), c(1, 2), type = "n", xlab = "gráfico 3", ylab = "")` *# gerar gráfico*

*# apenas com os eixos*

`polygon(c(1, 1, 2), c(1, 2, 1), density = 5, angle = 45)` *# adicionar o polígono de*

*# vértices (1,1), (1,2) e (2,1) gráfico 4*

`plot(c(0, 3), c(1, 2), type = "n", xlab = "gráfico 4", ylab = "")`

`polygon(c(1, 2, 2, 1), c(2, 2, 1, 1), density = 15, angle = -45, col = "red")` *# adicionar*

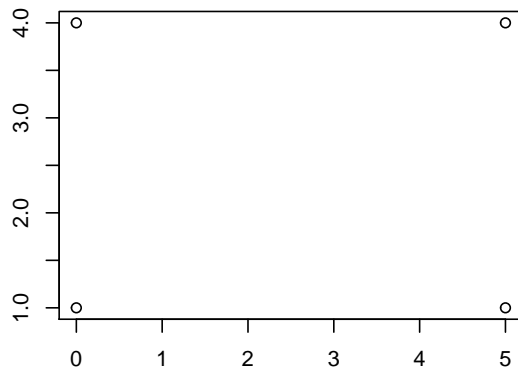


gráfico 1

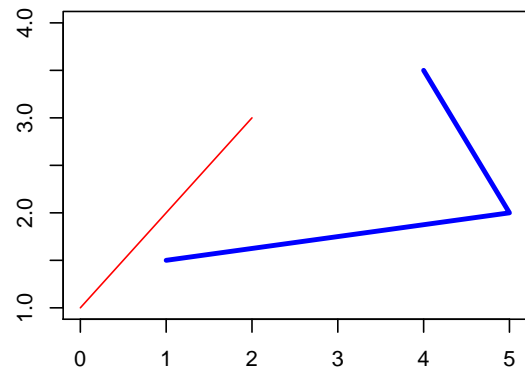


gráfico 2

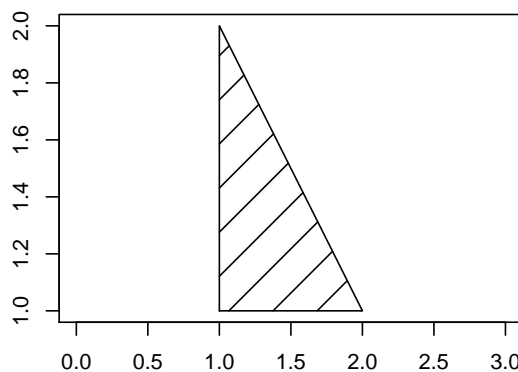


gráfico 3

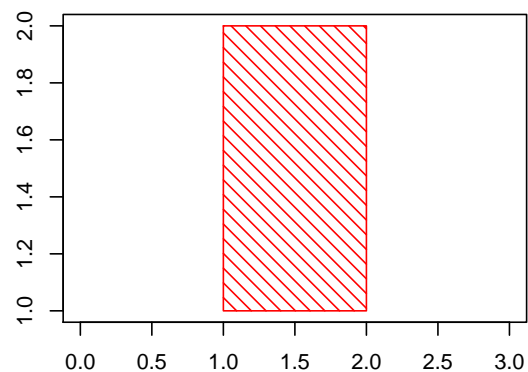


gráfico 4

```

# o polígono de vértices (1,2), (2,2), (2,1) e (1,1)
par(old.par)

old.par = par(mfrow = c(1, 2))
# gráfico 5
plot(c(0, 3), c(1, 2), type = "n", xlab = "gráfico 5", ylab = "")
polygon(c(1, 2, 2), c(2, 2, 1), density = 15, angle = 90, col = "blue") # adicionar o
# polígono de vértices (1,2), (2,2) e (2,1) gráfico 6
plot(c(0, 9), c(0, 2), type = "n", xlab = "gráfico 6", ylab = "")
polygon(1:9, c(2, 1, 2, 1, NA, 2, 1, 2, 1), density = c(10, 20), angle = c(-45,
45), col = c("red", "blue"))

```

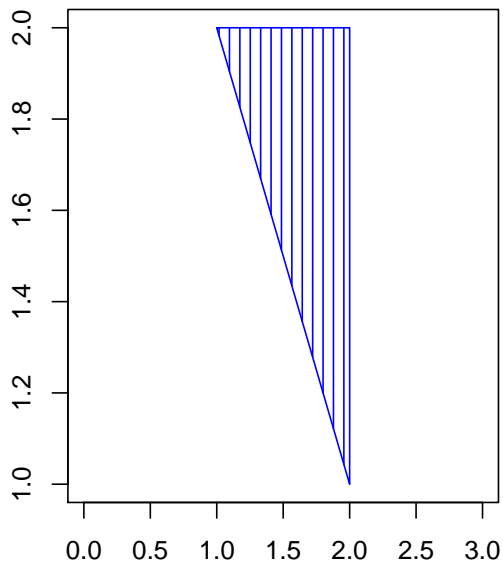


gráfico 5

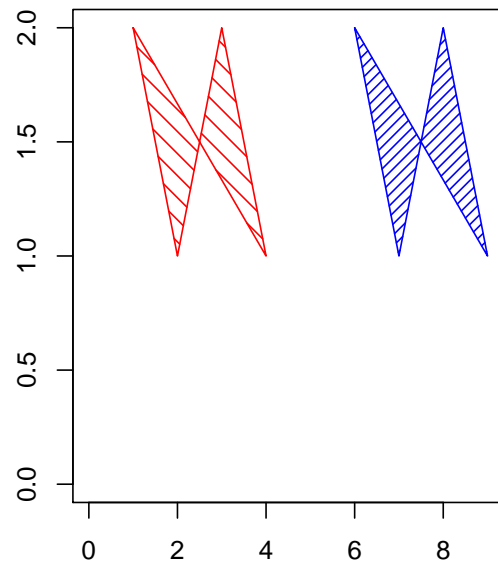


gráfico 6

```

par(old.par)

```