

# Introdução ao Programa R: Elaboração e Customização de Gráficos

**Gilberto Rodrigues Liska; Josiane Rodrigues; Juliano Bortolini**  
gilbertoliska@ufscar.br; josirodrigues@ufscar.br;  
julianobortolini@ufmt.br

*Material de Apoio*



# Sumário

1 Gráficos

2 Gráficos de funções

# Introdução

- Agora que já temos em mãos as ferramentas para importar, arrumar e transformar os nossos dados, é hora de começarmos a extrair informações deles.
- O próximo passo é a construção de visualizações.
- A visualização dos dados é uma etapa importantíssima em muitas análises (estatística, química, física, biológica etc.).
- No caso da estatística, é a partir dela que criamos a intuição necessária para escolher o teste ou modelo mais adequado para o nosso problema.
- Visualizações podem ser uma simples medida resumo (frequência, média, variância, mínimo, máximo, ...), um conjunto dessas medidas organizadas em uma tabela ou a representação (de uma parte) dos dados em um gráfico.



Fonte: <https://pixabay.com/images/id-294525/>

# Gráficos

## Mas, o que é um gráfico?

- Segundo o estatístico norte-americano Leland Wilkinson, “um gráfico é o mapeamento dos dados a partir de atributos estéticos (posição, cor, forma, tamanho) de objetos geométricos (pontos, linhas, barras, caixas)” (WILKINSON, 2005).
- Em 2005, Leland publicou o livro *The Grammar of graphics*, uma fonte de princípios fundamentais para a construção de gráficos estatísticos. O link para ver o livro é <https://www.springer.com/gp/book/9780387245447>
- A seguir apresentaremos vários exemplos de elaboração e customização de gráficos no R. Crie um novo script no seu computador, copie e cole os exemplos nos seu script e veja o que acontece.



Fonte: <https://pixabay.com/images/id-48100/>

# Gráficos: função plot

## Comando PLOT

- O comando mais simples pra fazer um gráfico no R com informações de dois objetos do tipo numérico é o *plot*.
- Esse gráfico também é chamado de gráfico de dispersão.

## Exemplo 1 no R:

```
# Gráficos
x <- c(1, 5, 10, 15, 20, 25, 30, 35)
y <- c(8, 16, 14, 21, 38, 25, 43, 39)

# gráficos de dispersão
?plot #ajuda da funcao
plot(y ~ x)
# ou
plot(x, y) # forma equivalente
```

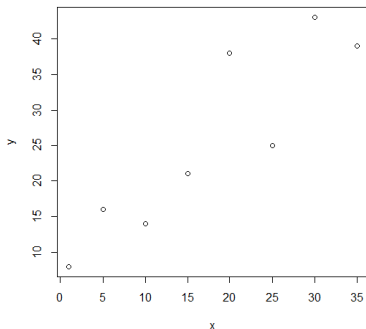


Figura 1: Gráfico do Exemplo 1.

# Gráficos: explorando a função plot

## Alguns argumentos da função plot

- col: cor ("nome", número ou rgb())
- pch: tipo de caracter
- type: tipo de gráfico
- cex: tamanho do caracter
- lty: tipo de linha
- xlim: limites do eixo x (horizontal)
- ylim: limites do eixo y (vertical)
- xlab: título do eixo x
- ylab: título do eixo y
- main: título do gráfico

## Exemplo 2 no R:

```
plot(x,y,  
     col="blue", pch=1, type="p",  
     cex=1.0,lty=1,xlim=c(0,45),  
     ylim=c(0,50),xlab="variável x",  
     ylab="variável y",main = "Título")
```

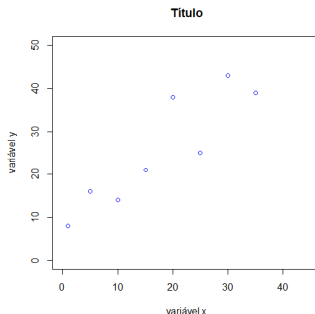


Figura 2: Gráfico do Exemplo 2.

# Gráficos: explorando a função plot

## Tipo de caracter

- É controlado pelo argumento *pch*.
- Varia de 0 a 25, com cada número representando um caracter.

0	1	2	3	4	5	6	7	8	9	10	11	12	13
□	○	△	+	×	◇	▽	⊠	*	◆	⊕	⊗	⊞	⊟
14	15	16	17	18	19	20	21	22	23	24	25		
⊠	■	●	▲	◆	●	●	●	●	■	◇	△	▽	

## Exemplo 3 no R:

```
# plotando 4 gráficos
par(mfrow=c(2,2))
plot(x,y, type="p", pch=1)
plot(x,y, type="p", pch=3)
plot(x,y, type="p", pch=5)
plot(x,y, type="p", pch=25)
# plotando apenas um gráfico
par(mfrow=c(1,1))
```

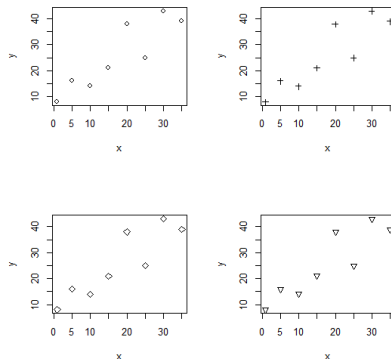


Figura 3: Gráfico do Exemplo 3.

# Gráficos: explorando a função plot

## Tipo do gráfico

É controlado pelo argumento *type*.

- “p”: para pontos
- “l”: para linhas
- “b”: para ambos
- “c”: para linhas sem os pontos
- “o”: igual ao “b”, porém com linhas e pontos sobrepostos
- “h”: para linhas verticais
- “s”: para degraus
- “S”: para degraus (outro tipo)
- “n”: não plota os pontos

## Exemplo 4 no R:

```
par(mfrow=c(2,2))
plot(x,y, type="p")
plot(x,y, type="l")
plot(x,y, type="b")
plot(x,y, type="o")
par(mfrow=c(1,1))
```

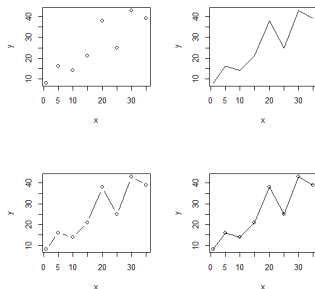


Figura 4: Gráfico do Exemplo 4.



# Gráficos: explorando a função plot

## Tamanho do caracter

- É controlado pelo argumento `cex`.
- Varia numa escala contínua de 0 a um número grande (aproximadamente 100).
- Não é indicado colocar números maiores do que 3.

## Exemplo 5 no R:

```
# tamanho do caracter
par(mfrow=c(2,2))
plot(x,y, type="p", cex=1)
plot(x,y, type="p", cex=3)
plot(x,y, type="p", cex=5)
plot(x,y, type="p", cex=10)
par(mfrow=c(1,1))
```

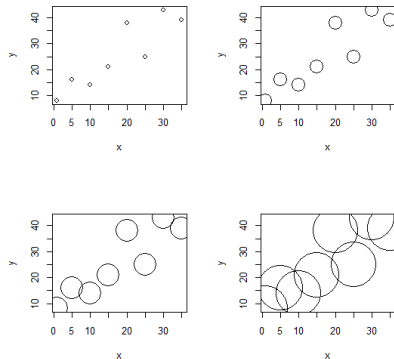


Figura 5: Gráfico do Exemplo 5.

# Gráficos: explorando a função plot

## Tipo da linha

- É controlado pelo argumento *lty*.
- Varia numa escala discreta de 1 a um número grande (aproximadamente 30).
- Quanto maior o número, mais tracejada fica a linha.

## Exemplo 6 no R:

```
# tipo de linha
par(mfrow=c(2,2))
plot(x,y, type="l", lty=0)
plot(x,y, type="l", lty=2)
plot(x,y, type="l", lty=3)
plot(x,y, type="l", lty=10)
par(mfrow=c(1,1))
```

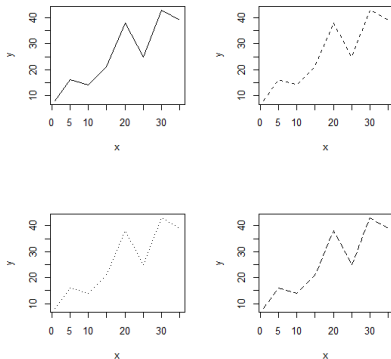


Figura 6: Gráfico do Exemplo 6.

# Gráficos: explorando a função plot

## Espessura da linha

- É controlado pelo argumento *lwd*.
- Varia numa escala discreta de 1 a um número grande (aproximadamente 30).
- Quanto maior o número, mais espessa fica a linha.

## Exemplo 7 no R:

```
# espessura de linha
par(mfrow=c(2,2))
plot(x,y, type="l", lwd=1)
plot(x,y, type="l", lwd=2)
plot(x,y, type="l", lwd=3)
plot(x,y, type="l", lwd=5)
par(mfrow=c(1,1))
```

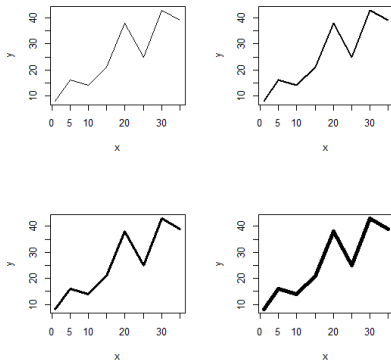


Figura 7: Gráfico do Exemplo 7.

# Gráficos: explorando a função plot

## Cor

- É controlado pelo argumento *col*.
- Podemos colocar a cor pelo seu nome.
- Existe uma lista com 657 nomes de cores no R. Para vê-las, acesse o link <https://www.datanovia.com/en/blog/awesome-list-of-657-r-color-names/>.

## Exemplo 8 no R:

```
par(mfrow=c(2,2))
plot(x,y, type="p", col="black")
plot(x,y, type="p", col="blue")
plot(x,y, type="p", col="yellow")
plot(x,y, type="p", col="red")
par(mfrow=c(1,1))
# para ver as primeiras 20 cores
r_color <- colors()
head(r_color, 20)
```

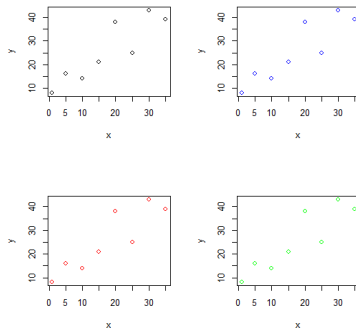


Figura 8: Gráfico do Exemplo 8.

# Gráficos: explorando a função plot

## Cor (continuação)

- Alternativamente, a cor pode ser selecionada por um número discreto de 1 a 8.
- Números além de 8 retornam aos anteriores, por exemplo, colocando 10 corresponde à cor de número 2.
- Outra forma é colocar a cor em termos dos componentes RGB formado pela sequência "#RRGGBB".

## Exemplo 9 no R:

```
par(mfrow=c(4,4))
for (i in c(1:16))
  plot(x, y, ylab="", xlab="",
       type="p", lwd=3,
       main=paste("col=",as.character(i)),
       col=i)
```

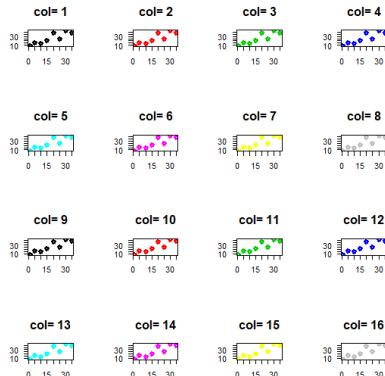


Figura 9: Gráfico do Exemplo 9.

# Gráficos: explorando a função plot

## Legenda

A função *legend* tem vários argumentos:

- x,y: a posição ("bottomright", "bottom", "bottomleft", "left", "topleft", "top", "topright", "right", "center")
- legend: texto da legenda
- col: cor dos pontos ou linhas
- pch: tipo do ponto na legenda
- lty, lwd: tipo e espessura das linhas que aparecerão na legenda
- bty: tipo da caixa em torno da legenda. "o": aparece caixa, "n": não aparece caixa.
- ncol: nº de colunas na legenda
- title: título da legenda
- horiz: se TRUE a legenda aparecerá horizontalmente

## Exemplo 10 no R:

```
x # do conjunto de dados anterior
y1=log(x)
y2=sqrt(x)
plot(x,y1, ylim=c(0,8), ylab="y")
points(x,y2, col="red")
legend("top",
      legend = c("logarítimo","raiz quadrada"),
      pch = c(1,1),col = c(1,2), lty = 0,
      bty="n")
```

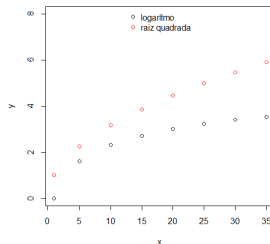


Figura 10: Gráfico do Exemplo 10.

# Gráficos: explorando a função plot

## Tipo de letra

- *font*: é um número inteiro que especifica o tipo de fonte que será usado no texto.
  - 1 o padrão
  - 2 negrito
  - 3 itálico
  - 4 negrito e itálico
- *font.axis*: tipo de fonte no texto dos eixos
- *font.lab*: tipo de fonte nos nomes dos eixos
- *font.main*: tipo de fonte no texto do título

## Exemplo 11 no R:

```
plot(x,y, main = "Título",  
     xlab = "eixo x", ylab = "eixo y",  
     font.axis = 2,  
     font.lab = 3,  
     font.main = 4)
```

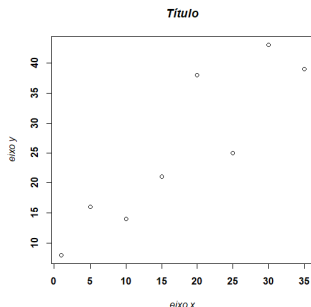


Figura 11: Gráfico do Exemplo 11.

# Gráficos: explorando a função plot

## Cor do texto no gráfico

- *col*: como explicado anteriormente
- *col.axis*: cor da fonte no texto dos eixos
- *col.lab*: cor da fonte nos nomes dos eixos
- *col.main*: cor da fonte no texto do título
- *col.sub*: cor do texto do sub-título (se houver)
- *fg*: cor das linhas dos eixos no gráfico

**OBS.:** As cores podem ser selecionadas por números também!

## Exemplo 12 no R:

```
plot(y ~ x, main = "Título",  
      xlab = "eixo x", ylab = "eixo y",  
      col = "green",  
      col.axis = "blue",  
      col.lab = "red",  
      col.main = "yellow",  
      fg = "gray")
```

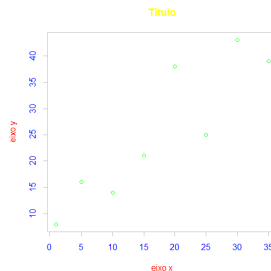


Figura 12: Gráfico do Exemplo 12.



# Gráficos: explorando a função plot

## Estilo dos textos dos eixos

- Controlado pelo argumento *las*.
- É um argumento numérico com os valores:
  - 0: texto paralelo aos eixos (padrão)
  - 1: texto horizontal
  - 2: texto perpendicular aos eixos
  - 3: texto vertical

## Exemplo 13 no R:

```
# estilo dos rótulos dos eixos
par(mfrow=c(2,2))
plot(x,y, las = 0)
plot(x,y, las = 1)
plot(x,y, las = 2)
plot(x,y, las = 3)
par(mfrow=c(1,1))
```

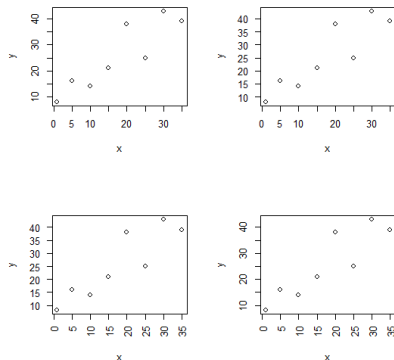


Figura 13: Gráfico do Exemplo 13.

# Gráficos: explorando a função plot

## Alterando os limites dos eixos

- Controlado pelos argumentos *xlim* (eixo horizontal) e *ylim* (eixo vertical)
- Se especificado, deve-se informar o início e o fim do intervalo desejado.
- Se não for especificado, o R tem por padrão plotar o gráfico considerando a amplitude dos dados em cada eixo.

### Exemplo 14 no R:

```
par(mfrow=c(2,1))
plot(x,y) #original
plot(x,y, xlim = c(10, 35),
     ylim = c(0, 40))
par(mfrow=c(1,1))
```

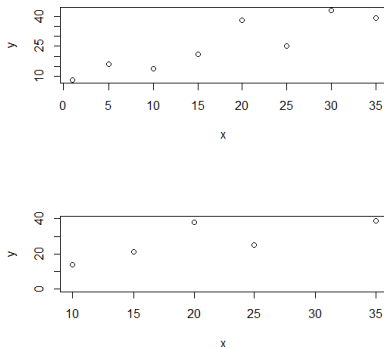


Figura 14: Gráfico do Exemplo 14.

# Gráficos: explorando a função plot

## Estilo da caixa do gráfico

- Controlado pelos argumentos *bty*.
- As opções são "o" (o padrão), "l", "7", "c", "u", "]", ou "n".

## Exemplo 15 no R:

```
par(mfrow=c(4,2))
plot(x,y, pch = 15, cex = 2.1,)
plot(x,y, pch = 15, cex = 2.1, bty = "o")
plot(x,y, pch = 15, cex = 2.1, bty = "l")
plot(x,y, pch = 15, cex = 2.1, bty = "7")
plot(x,y, pch = 15, cex = 2.1, bty = "c")
plot(x,y, pch = 15, cex = 2.1, bty = "u")
plot(x,y, pch = 15, cex = 2.1, bty = "]")
plot(x,y, pch = 15, cex = 2.1, bty = "n")
par(mfrow=c(1,1))
```

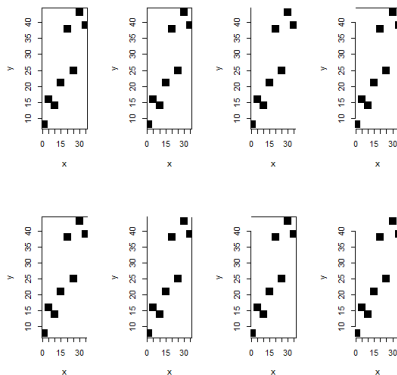


Figura 15: Gráfico do Exemplo 15.

# Gráficos: explorando a função plot

## colocando grid

- O grid é uma grade regular que fica no fundo do gráfico.
- Auxilia na visualização das coordenadas de um ponto particular.
- É adicionado ao gráfico pelo argumento *panel.first*, que tem como argumentos

*nx, ny*: número de células no grid

*col*: cor das linhas do grid

*lty*: tipo da linha do grid

*lwd*: espessura da linha do grid

## Exemplo 16 no R:

```
plot(x,y,panel.first=grid())
plot(x,y,panel.first=grid(lwd=2, lty=3))
plot(x,y,panel.first=grid(ny=5,col="red"))
plot(x,y,panel.first=grid(nx=5,col="red"))
```

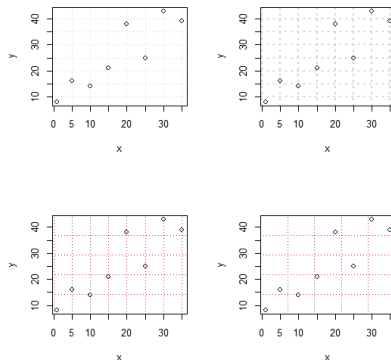


Figura 16: Gráfico do Exemplo 16.

# Gráficos: exportando figuras

## Exportando figuras

No RStudio é possível exportar figuras de duas formas:

- (1): Pelo menu "Export" (tela 4).
- (2): Por linha de comando. Nesse caso, os tipos permitidos são: **bmp()**; **jpeg()**; **png()**; **tiff()**; **pdf()**; **postscript()**; etc

Esses comandos têm ainda os seguintes argumentos:

**filename:** nome do arquivo seguido de sua extensão

**width:** comprimento da figura

**height:** altura da figura

**units:** as unidades do width e height (pode ser "px"(pixels, o padrão), "in"(polegadas), "cm"ou "mm").

**res:** resolução da figura (válido para jpeg e não válido para pdf)

**IMPORTANTE:** Esses comandos devem ser acompanhados do **dev.off()**, que fechará a janela criada pelos comandos acima.

### Exemplo 17 no R:

```
### Salvando graficos
# 1º certifique-se do diretorio
setwd("Particao/Pasta/Nome do arquivo")

pdf(file = "gráfico.pdf")
plot(x,y)
dev.off()

jpeg(file = "gráfico.jpeg", width = 5,
      height = 5, units = "in", res = 300)
plot(x,y)
dev.off()

png(file = "gráfico.png", width = 6,
     height = 6, units = 'in', res = 300)
plot(x,y)
dev.off()
```

# Sumário

1 Gráficos

2 Gráficos de funções

# Gráficos de funções

## Gráfico de função de duas variáveis

- O R fornece recursos fáceis para construção de gráficos de funções.  
**OBS.:** Agora estamos falando de funções matemáticas e não as funções criadas anteriormente dentro do R, como feito na aula anterior.
- A ideia básica é criar um vetor com valores das abscissas (valores de  $x$ ) e calcular o valor da função (valores de  $f(x)$ ) para cada elemento da função e depois fazer o gráfico unindo os pares de pontos.

### Exemplo 18

Para ilustrar como podemos fazer gráficos de funções vamos considerar cada uma das funções a seguir:

(a)  $f(x) = 5 \ln(x); 0 \leq x \leq 100$

(b)  $f(x) = \frac{1}{\sqrt{50\pi}} e^{-\frac{1}{50}(x-100)^2}; 80 \leq x \leq 120$

(c)  $f(x) = 1 - \frac{1}{x} \text{seno}(x); 0 \leq x \leq 50$

# Gráficos de funções

## Exemplo 18 (a)

O gráfico da função

$$f(x) = 5 \ln(x)$$

no intervalo  $0 \leq x \leq 100$  pode ser construído da seguinte forma no R:

## Exemplo 18 (a) no R:

```
# definindo os valores de x
x1 <- seq(0,100, by=0.1)
y1 <- 5*log(x1)
plot(x1, y1, type="l", col=2)

## alternativamente
y=function(x) 5*log(x)
y1=y(x1)
plot(x1, y1, type="l", col=2,lwd=2,
     panel.first = grid())
## alternativamente
plot(function(x) 5*log(x), 0, 10)
```

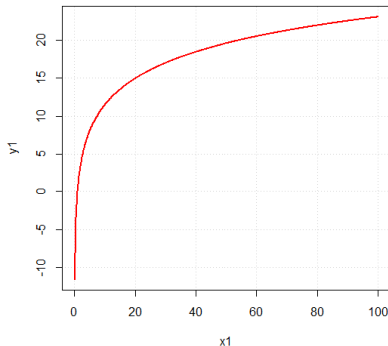


Figura 17: Gráfico do Exemplo 18 (a).



# Gráficos de funções

## Exemplo 18 (b)

O gráfico da função

$$f(x) = \frac{1}{\sqrt{50\pi}} e^{-\frac{1}{50}(x-100)^2}$$

no intervalo  $80 \leq x \leq 120$  pode ser construído da seguinte forma no R:

### Exemplo 18 (b) no R:

```
# definindo os valores de x
x2 <- seq(80, 120, l=101)
y2 <- (1/sqrt(50*pi))*exp(-0.02*(x2-100)^2)
plot(x2, y2, type="l")
```

```
## alternativamente
y=function(x) (1/sqrt(50*pi))*
              exp(-0.02*(x-100)^2)
y2=y(x2)
plot(x2, y2, type="l", col=2,lwd=2,
      panel.first = grid())
```

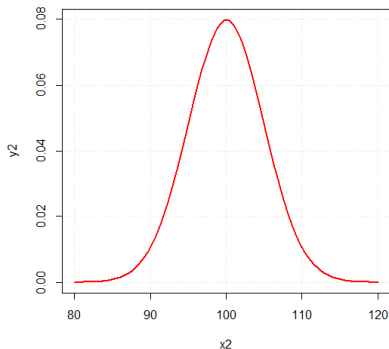


Figura 18: Gráfico do Exemplo 18 (b).

# Gráficos de funções

## Gráfico de função de três variáveis

- Similarmente ao feito para duas variáveis, é possível construir gráfico quando temos três variáveis, resultando num gráfico tridimensional.
- A ideia básica é criar dois vetores com valores das abscissas (valores de  $x$  e  $y$ ) e calcular o valor da função (valores de  $f(x, y)$ ) para cada elemento, formado pelo par  $(x, y)$ , da função e depois fazer o gráfico unindo os pares de pontos.
- No R, além de criar os vetores de valores, devemos criar um grid com todas as combinações dos valores de  $x$  e  $y$ .
- Por exemplo, se quisermos montar o gráfico da função

$$f(x, y) = \sqrt{x^2 + y^2}; 1 \leq x, y \leq 5$$

com valores de  $x$  e  $y$  sendo 1, 2, 3, 4 e 5, temos que criar as combinações desses dois vetores, resultando em 25 valores.

# Gráficos de funções

## expand.grid

Cria uma tabela com todas as combinações dos vetores ou fatores. Seus argumentos são:

- x: primeiro vetor
- y: segundo vetor
- stringsAsFactors: argumento lógico. Se TRUE, os vetores são convertidos para fatores. Default é FALSE.

## Exemplo 19

Obtenha todas combinações de valores de  $1 \leq x, y \leq 4$ , com valores de x e y sendo 1, 2, 3, e 4.

## Exemplo 19 no R:

```
x<-seq(1,4,by=1); x
y<-seq(1,4,by=1); y
expand.grid(x=x,y=y)
```

	x	y
1	1	1
2	2	1
3	3	1
4	4	1
5	1	2
6	2	2
7	3	2
8	4	2
9	1	3
10	2	3
11	3	3
12	4	3
13	1	4
14	2	4
15	3	4
16	4	4

# Gráficos de funções

## outer

Função que faz o produto de dois vetores de comprimento  $m$  e  $n$ , criando uma matriz de dimensão  $m \times n$  e calcula o valor de uma função a cada elemento da matriz.

## Exemplo 20

Calcule os valores da função

$f(x, y) = x^2 + y^2$ ;  $1 \leq x, y \leq 5$   
com valores de  $x$  e  $y$  sendo 1, 2, 3, 4 e 5.

## Exemplo 20 no R:

```
x<-seq(1,5,by=1); x
[1] 1 2 3 4 5
y<-seq(1,5,by=1); y
[1] 1 2 3 4 5
```

```
f1 <- function(x, y) x^2+y^2
```

```
z <- outer(x, y, f1)
z
```

	[,1]	[,2]	[,3]	[,4]	[,5]
[1,]	2	5	10	17	26
[2,]	5	8	13	20	29
[3,]	10	13	18	25	34
[4,]	17	20	25	32	41
[5,]	26	29	34	41	50

# Gráficos de funções

## persp

Função que constrói uma superfície sobre o plano x-y. Seus argumentos são:

- x: vetor das coordenadas x
- y: vetor das coordenadas y
- z: uma **matriz** com os valores para serem plotados
- xlim, ylim, zlim: limites dos eixos x, y e z
- theta: rotaciona o gráfico no sentido horizontal
- phi: rotaciona o gráfico no sentido vertical
- ticktype: apenas setas nos eixos se "*simple*" e números nos eixos se "*detailed*"
- nticks: quantidade (aproximada) de números nos eixos
- col: cor da superfície

## Exemplo 20 (cont.) no R:

```
persp(x, y, z, theta = 30, phi = 30,
      col = "lightblue")
persp(x, y, z, theta = 20, phi = 30,
      col = "lightblue",
      ticktype="detailed",nticks=5)
```

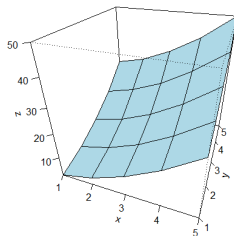


Figura 19: Gráfico do Exemplo 20.

# Gráficos de funções

## Exemplo 21

Faça os gráficos das seguintes funções:

(a)  $f(x, y) = \sqrt{x^2 + y^2}; -1 \leq x, y \leq 1$

(b)  $f(x, y) = \cos(x) \cos(y) e^{-\frac{\sqrt{x^2 + y^2}}{4}}; -5 \leq x, y \leq 5$

(c) A função a seguir é conhecida como função densidade de probabilidade normal bivariada,

$$f(x, y) = \frac{1}{2\pi\sigma_1\sigma_2\sqrt{1-\rho^2}} e^{\frac{1}{1-\rho^2} \left[ \left( \frac{x-\mu_1}{\sigma_1} \right)^2 - 2\rho \left( \frac{x-\mu_1}{\sigma_1} \right) \left( \frac{y-\mu_2}{\sigma_2} \right) + \left( \frac{y-\mu_2}{\sigma_2} \right)^2 \right]}$$

considere  $\sigma_1 = 1, \sigma_2 = 1, \mu_1 = 0, \mu_2 = 0, \rho = 0$  e  $-3 \leq x, y \leq 3$ .

# Gráficos de funções

## Exemplo 21 (a)

O gráfico da função

$$f(x, y) = \sqrt{x^2 + y^2}; -1 \leq x, y \leq 1$$

no intervalo  $-1 \leq x \leq 1$  pode ser construído da seguinte forma no R:

## Exemplo 21 (a) no R:

```
x<-seq(-1,1,l=50)
y<-seq(-1,1,l=50)

f1 <- function(x, y) sqrt(x^2+y^2)
z <- outer(x, y, f1)
## superficie
persp(x, y, z, theta = 30,
      phi = 30,
      col = "lightgreen",
      ticktype="detailed",
      nticks=5)
```

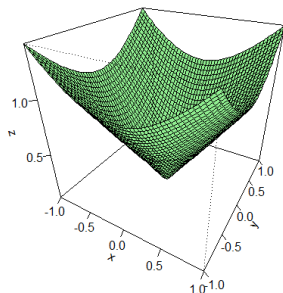


Figura 20: Gráfico do Exemplo 21 (a).

# Gráficos de funções

## Exemplo 21 (b)

O gráfico da função

$$f(x, y) = \cos(x) \cos(y) e^{-\frac{\sqrt{x^2+y^2}}{4}}$$

no intervalo  $-5 \leq x, y \leq 5$  pode ser construído da seguinte forma no R:

## Exemplo 21 (b) no R:

```
x<-seq(-5,5,l=50)
y<-seq(-5,5,l=50)
f1 <- function(x, y)
  cos(x)*cos(y)*exp((-sqrt(x^2+y^2))/4)
z <- outer(x, y, f1)
## superficie
persp(x, y, z, theta = 30,
      phi = 30,
      col = "lightsalmon",
      ticktype="detailed",
      nticks=5)
```

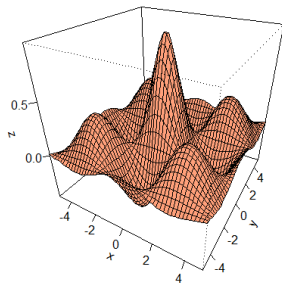


Figura 21: Gráfico do Exemplo 21 (b).



# Gráficos de funções: formas alternativas

Gráfico de contorno: comandos *image* e *contour*

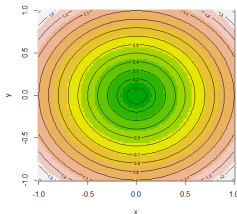


Figura 22: Gráfico de contorno do Exemplo 21 (a).

Exemplo 21 (a) no R:

```
## contornos
image(x, x, z,
      xlab = "x", ylab = "y",
      col=terrain.colors(10))
contour(x, x, z, col = "black",
        add = TRUE, nlev=10)
```

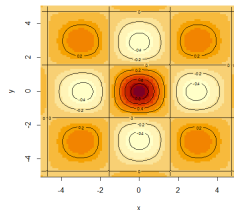


Figura 23: Gráfico de contorno do Exemplo 21 (b).

Exemplo 21 (b) no R:

```
## contornos
image(x, x, z,
      xlab = "x", ylab = "y")
contour(x, x, z, col = "black",
        add = TRUE, nlev=10)
```

# Gráficos de funções: formas alternativas

Visualização tridimensional: comando `plot_ly` do pacote `plotly`

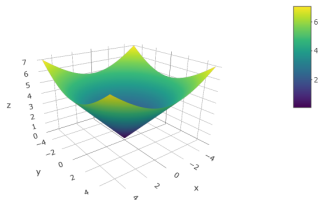


Figura 24: Gráfico do Exemplo 21 (a).  
Exemplo 21 (a) no R:

```
x<-seq(-1,1,l=100)
y<-seq(-1,1,l=100)
f1 <- function(x, y) sqrt(x^2+y^2)
z <- outer(x, y, f1)
library(plotly)
plot_ly(x=x,y=y,z=z, type="surface")
```

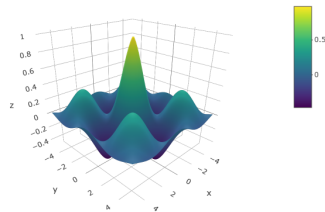


Figura 25: Gráfico do Exemplo 21 (b).  
Exemplo 21 (b) no R:

```
x<-seq(-5,5,l=100)
y<-seq(-5,5,l=100)
f1 <- function(x, y)
  cos(x)*cos(y)*exp((-sqrt(x^2+y^2))/4)
z <- outer(x, y, f1)
library(plotly)
plot_ly(x=x,y=y,z=z, type="surface")
```

# Prática

## Exercício

Faça o gráfico do exemplo 18 (c) e gráficos de superfície e contorno do exemplo 21 (c).

## SOLUÇÃO:

### Exemplo 18 (c) no R:

```
x1 <- seq(0,50, l=100)
y1 <- 1 - (1/x1) * sin(x1)
plot(x1, y1, type="l",
     col=2,lwd=2)
```

### Exemplo 21 (c) no R:

```
x <- seq(-3,3,length=100)
y <- x

f<-function(x,y){
  1/(2*pi*sqrt(s1*s2*(1-rho^2)))*
  exp(-1/(2*(1-rho^2))*
      ((x-m1)^2/s1+
       (y-m2)^2/s2-
       2*rho*((x-m1)*(y-m2))/(sqrt(s1)*sqrt(s2))))
}

#definindo os parâmetros
m1 <- 0
m2 <- 0
s1 <- 1
s2 <- 1
rho <- 0
z <- outer(x,y,f) # calcula f(x,y)
## superficie e contornos por conta do aluno
```

# Referência Bibliográfica

- WILKINSON, L. The Grammar of Graphics. Springer-Verlag New York, 2 ed, 2005. 691 p, doi 10.1007/0-387-28695-0.



Fonte:  
[https://pixabay.com/  
images/id-42701/](https://pixabay.com/images/id-42701/)