

Exemplos de gráficos em **R**

Este documento apresenta alguns exemplos de como criar e manipular gráficos em **R**.

1 Univariados

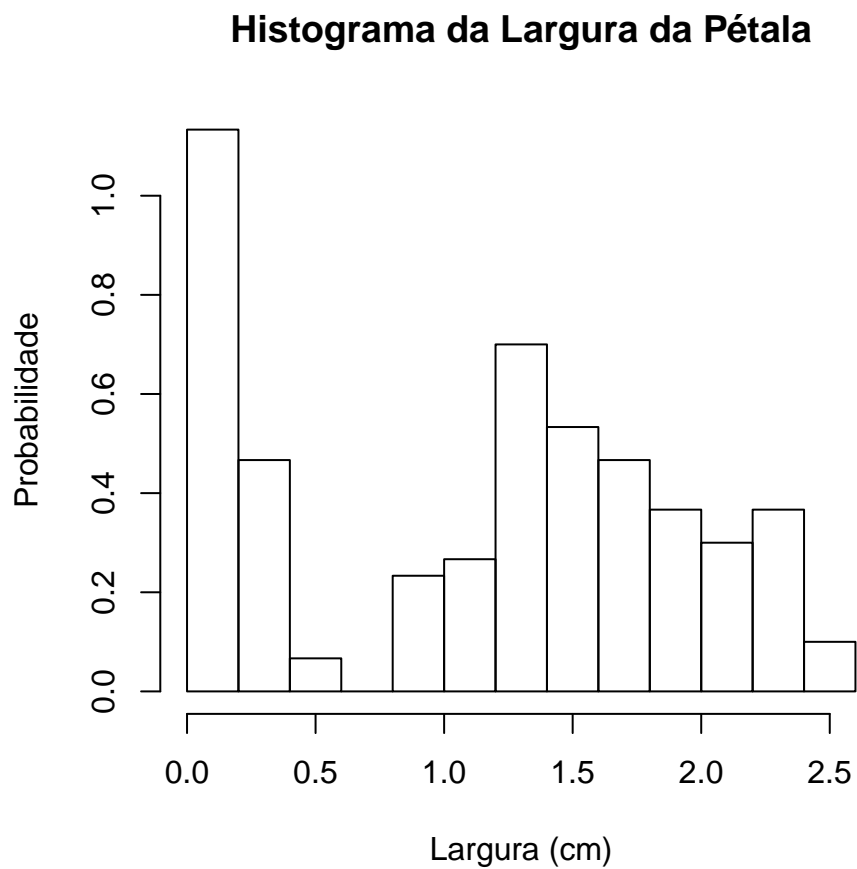
1. Histograma de uma variável.

```
hist(iris$Petal.Width)
```



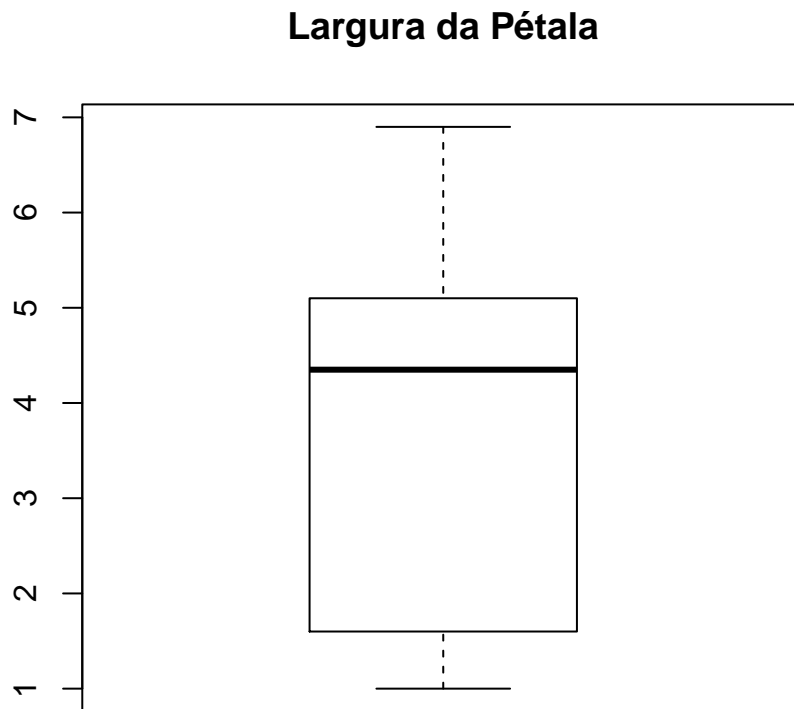
2. Histograma com título, legendas nos eixos, e densidade de probabilidade em vez de frequências.

```
hist(iris$Petal.Width, main = "Histograma da Largura da Pétala",  
     xlab = "Largura (cm)", ylab = "Probabilidade", prob = T)
```



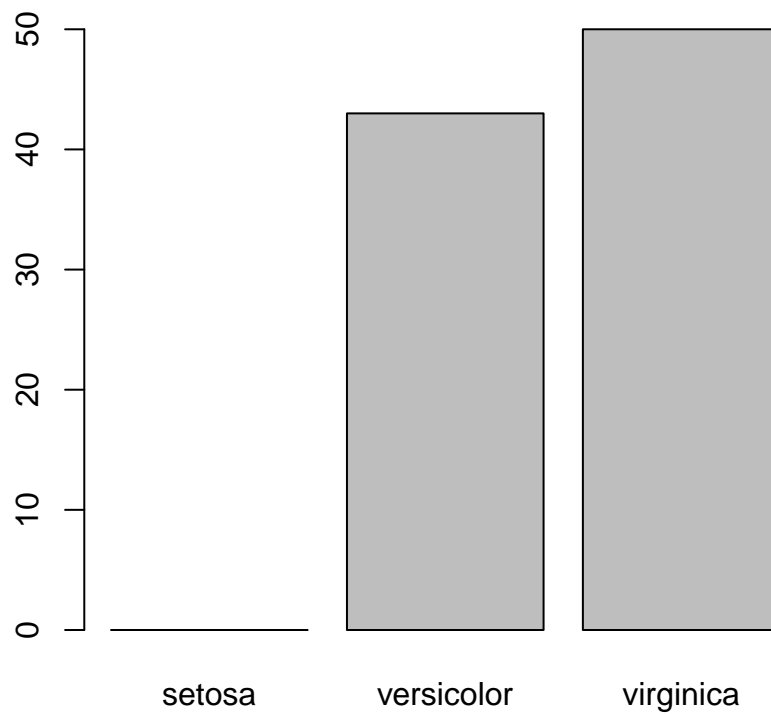
3. Boxplot de uma variável, com título

```
boxplot(iris$Petal.Length, main = "Largura da Pétala")
```



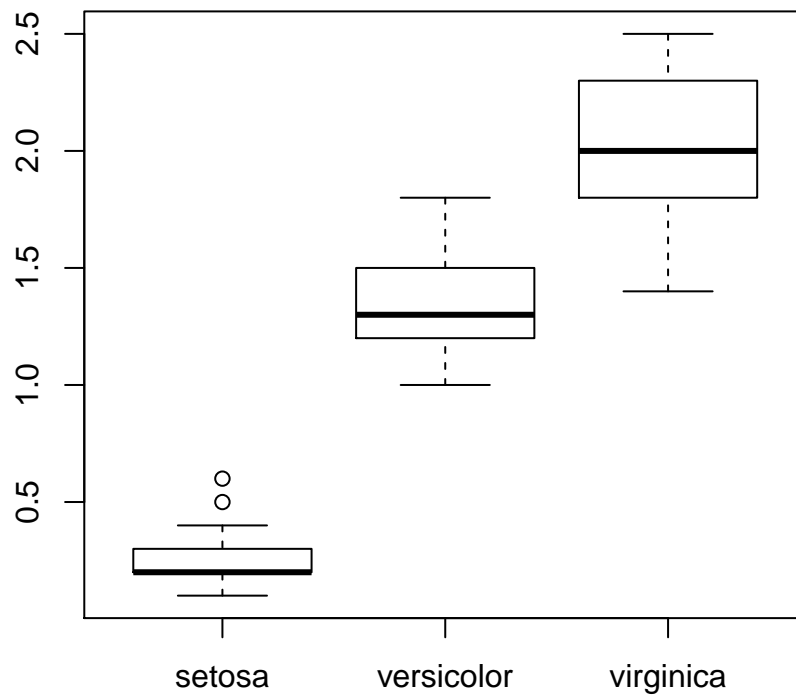
4. Gráfico de barras com a contagem do número de amostras em cada espécie, cuja pétala tem uma largura superior a 1.

```
barplot(table(subset(iris, Petal.Width > 1)$Species))
```



5. Boxplots condicionados, mostrando um para cada espécie.

```
boxplot(Petal.Width ~ Species, iris)
```



2 Três variáveis

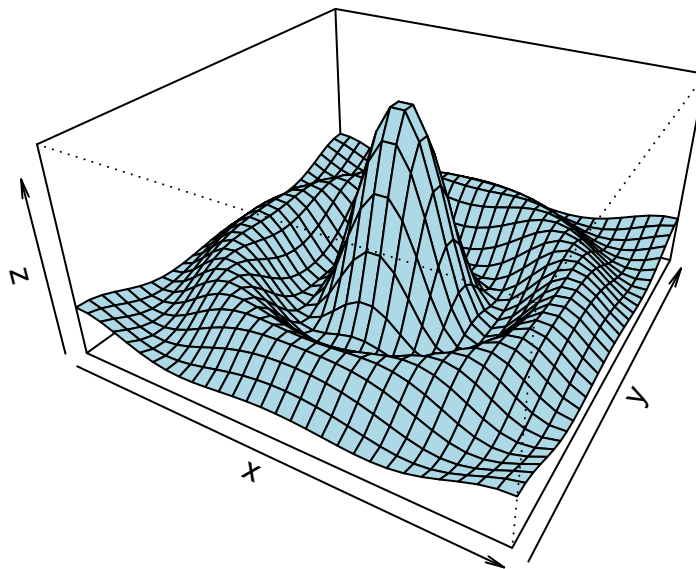
As seguintes funções usam formas diferentes de representar a terceira variável.

- `persp()` para produzir superfícies tridimensionais.
- `countour()` para produzir gráficos com curvas de nível.
- `image()` usando cores para representar a 3ª dimensão.

1. Função matemática, com três variáveis, representada em perspectiva.

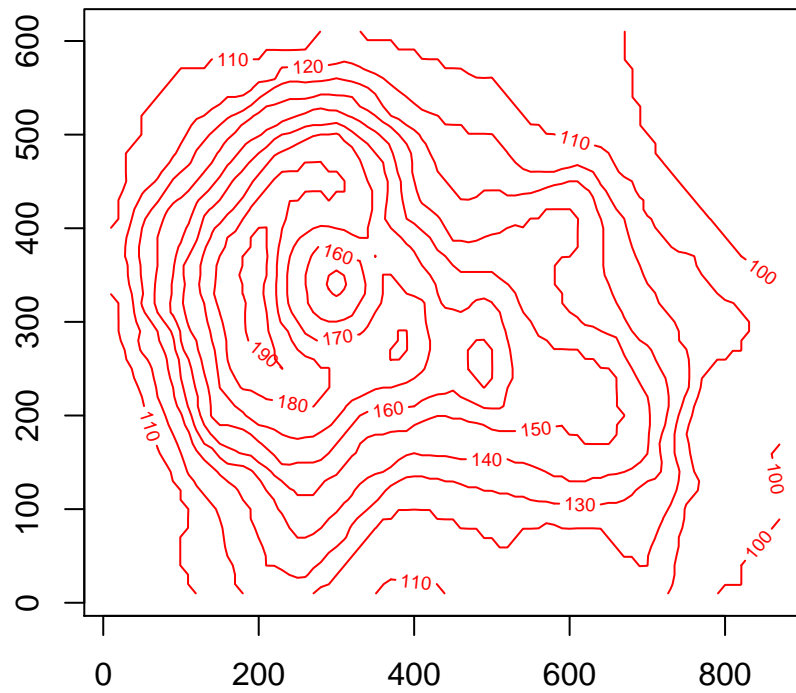
```
x = seq(-10, 10, length = 30)
y = x
f = function(x, y) {
  r = sqrt(x^2 + y^2)
  10 * sin(r)/r
}

z = outer(x, y, f)
z[is.na(z)] = 1
persp(x, y, z, theta = 30, phi = 30, expand = 0.5, col = "lightblue")
```



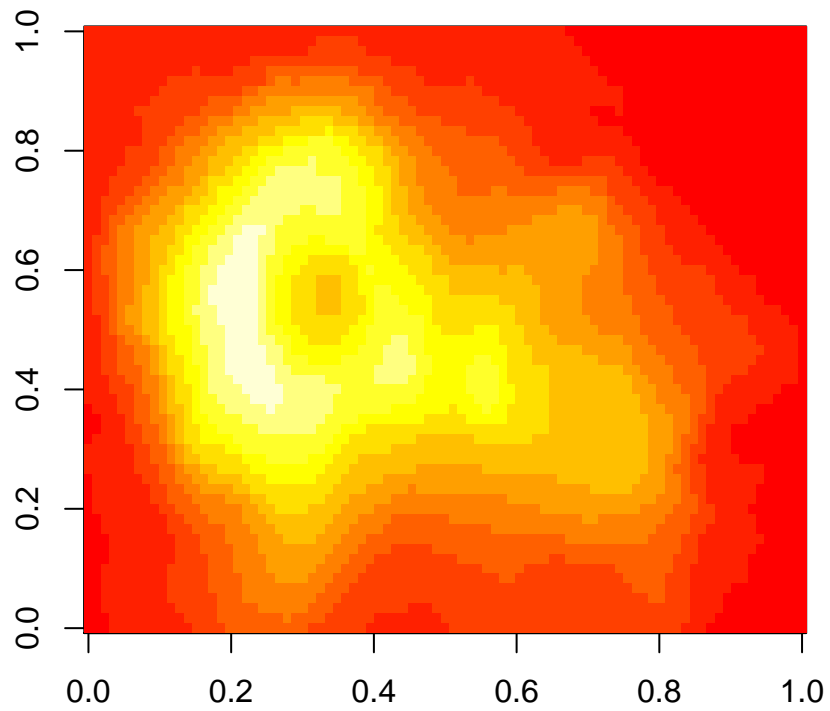
2. Carta altimétrica de um vulcão, com a altitude representada por curvas de nível.

```
x = 10 * 1:nrow(volcano)
y = 10 * 1:ncol(volcano)
contour(x, y, volcano, col = "red", lty = "solid")
```



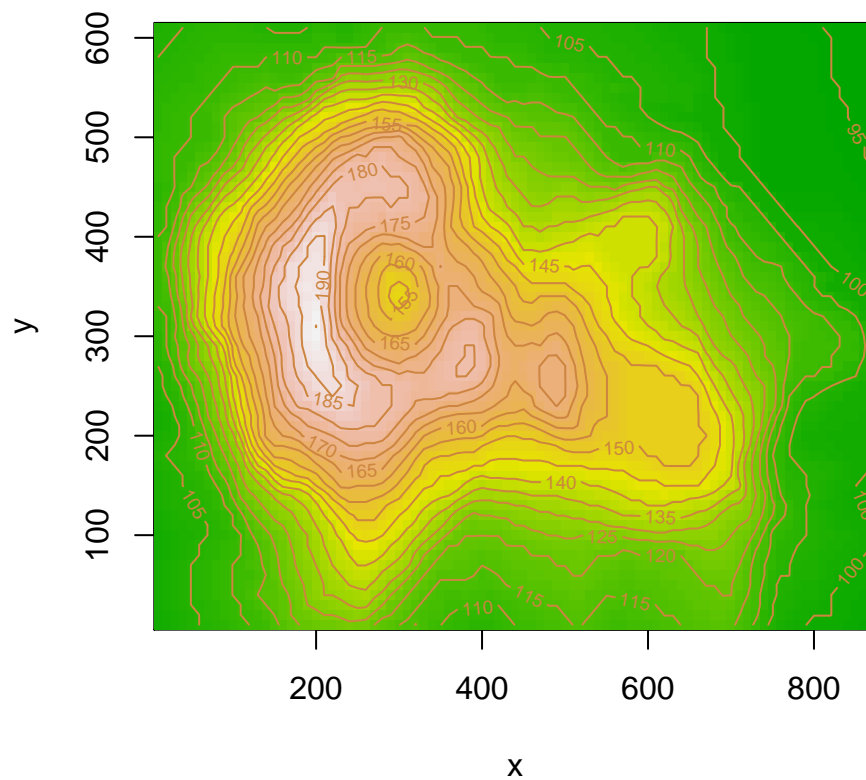
3. Agora com a altitude representada em função da cor (valores mais baixos a vermelho).

```
image(volcano)
```



4. Sobreposição das duas técnicas anteriores.

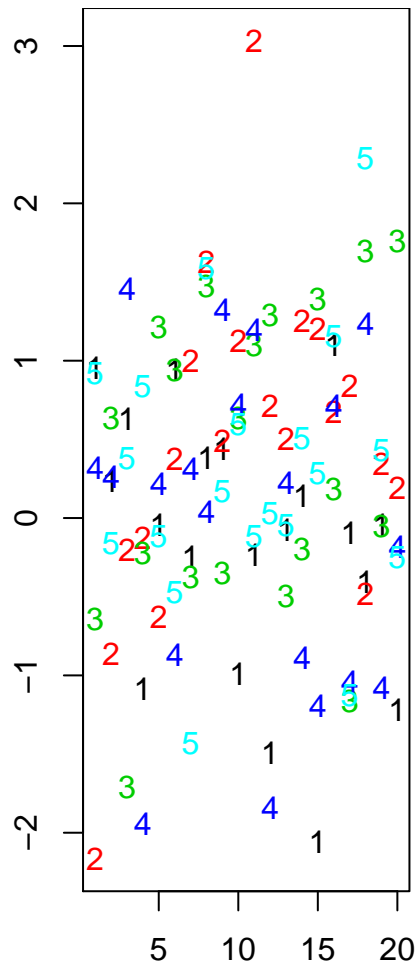
```
x = 10 * (1:nrow(volcano))
y = 10 * (1:ncol(volcano))
image(x, y, volcano, col = terrain.colors(100))
contour(x, y, volcano, levels = seq(90, 200, by = 5), add = TRUE,
        col = "peru")
```



3 Multivariados (suporte para mais de três variáveis)

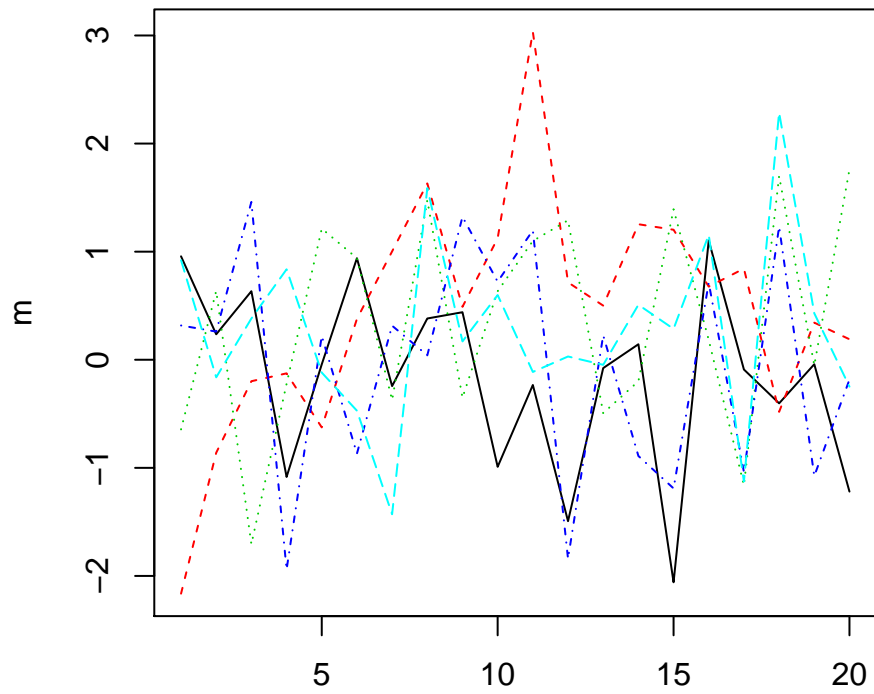
1. Cinco variáveis representadas por números e cores.

```
m = matrix(rnorm(100), 20, 5)
op = par(mfrow = c(1, 2), mar = c(2, 3, 0, 1))
matplot(m)
```



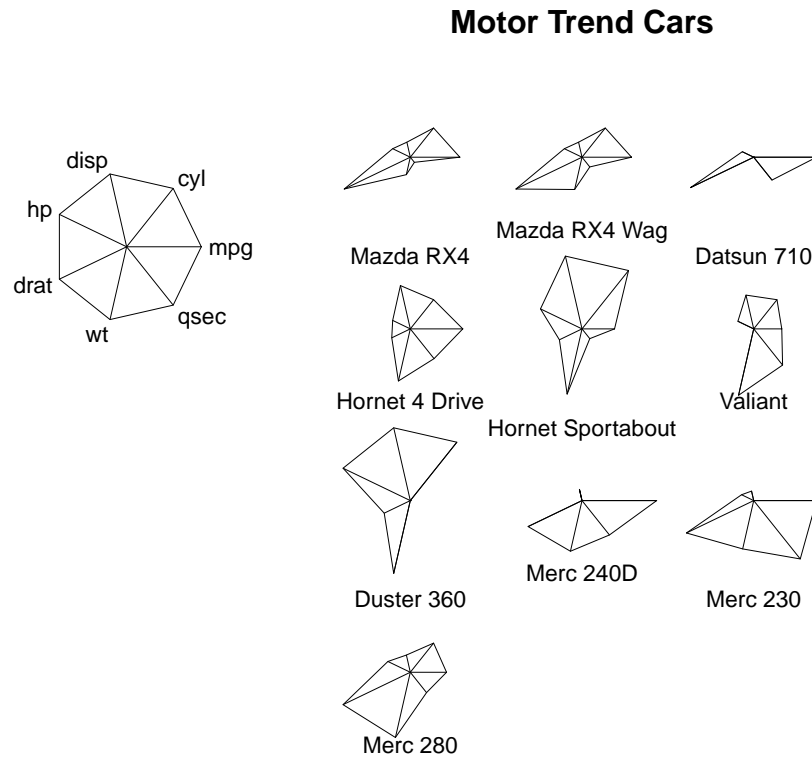
2. Igual, mas com uma representação por linhas.

```
matplot(m, type = "l")
```



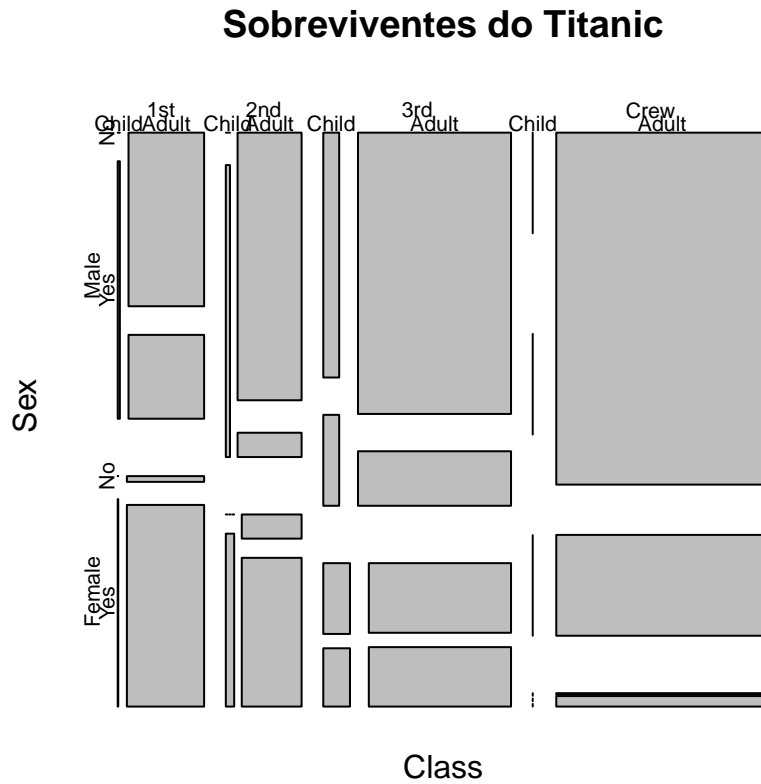
3. Gráfico de "estrelas" (*star/spider/radar plot*), com legenda à esquerda, reflectindo os valores de várias propriedades de automóveis.

```
stars(mtcars[1:10, 1:7], main = "Motor Trend Cars", key.loc=c(-1.5, 8))
```



4. Gráfico em mosaico, mostrando a contagem dos sobreviventes do acidente com o navio Titanic, discriminada por classe, faixa etária e gênero.

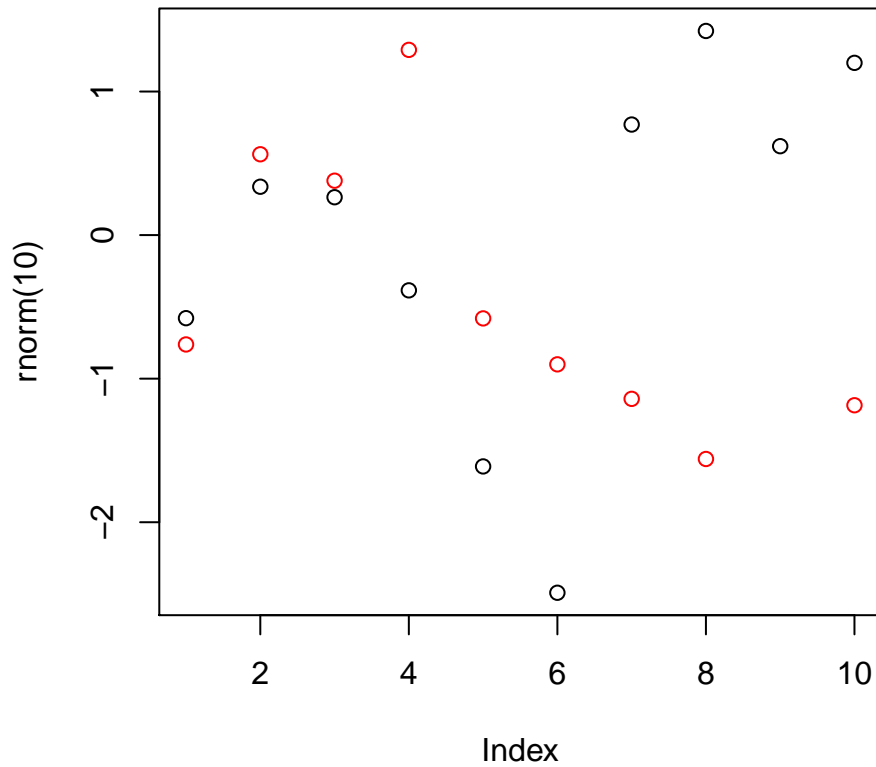
```
mosaicplot(Titanic, main = "Sobreviventes do Titanic")
```



4 Adicionar elementos a gráficos existentes

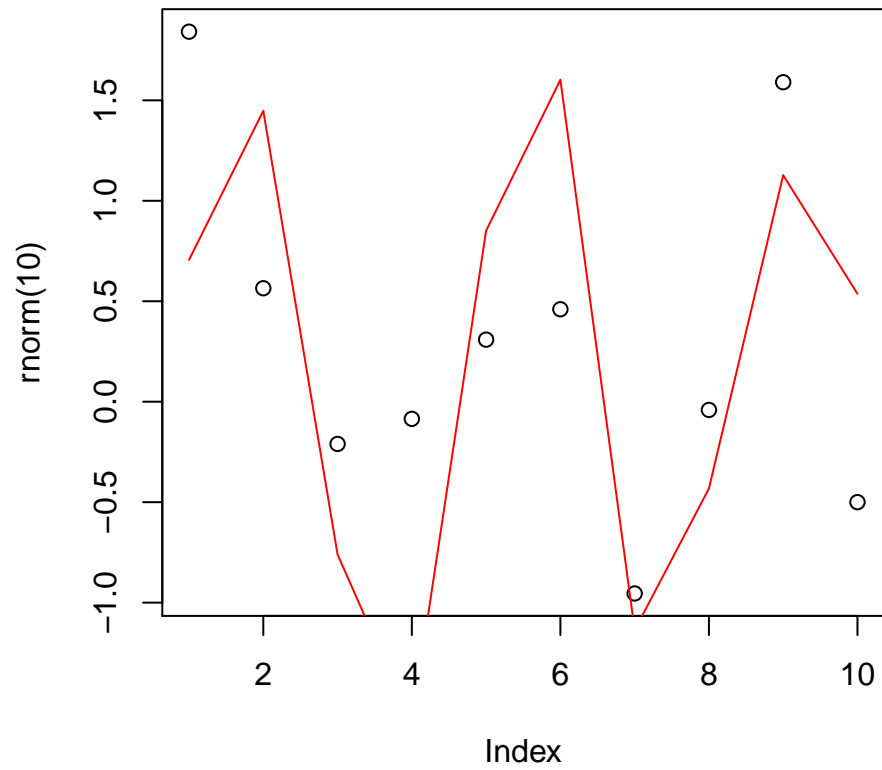
1. Adicionar pontos a um gráfico de pontos.

```
plot(rnorm(10))  
points(rnorm(10), col = "red")
```



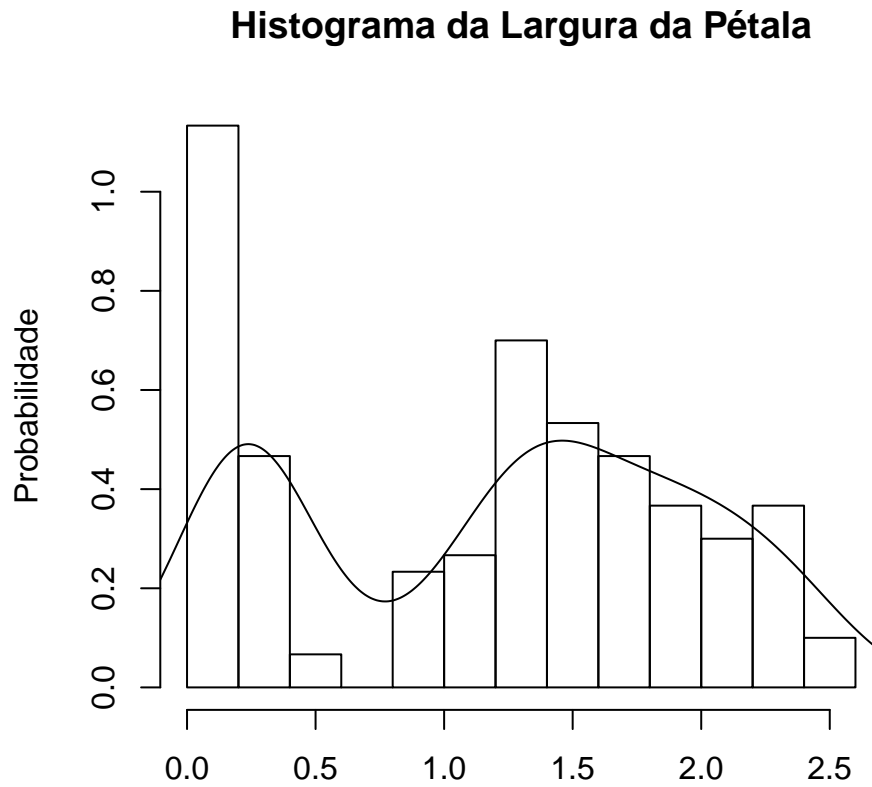
2. Adicionar linhas.

```
lines(rnorm(10), col = "red")
```



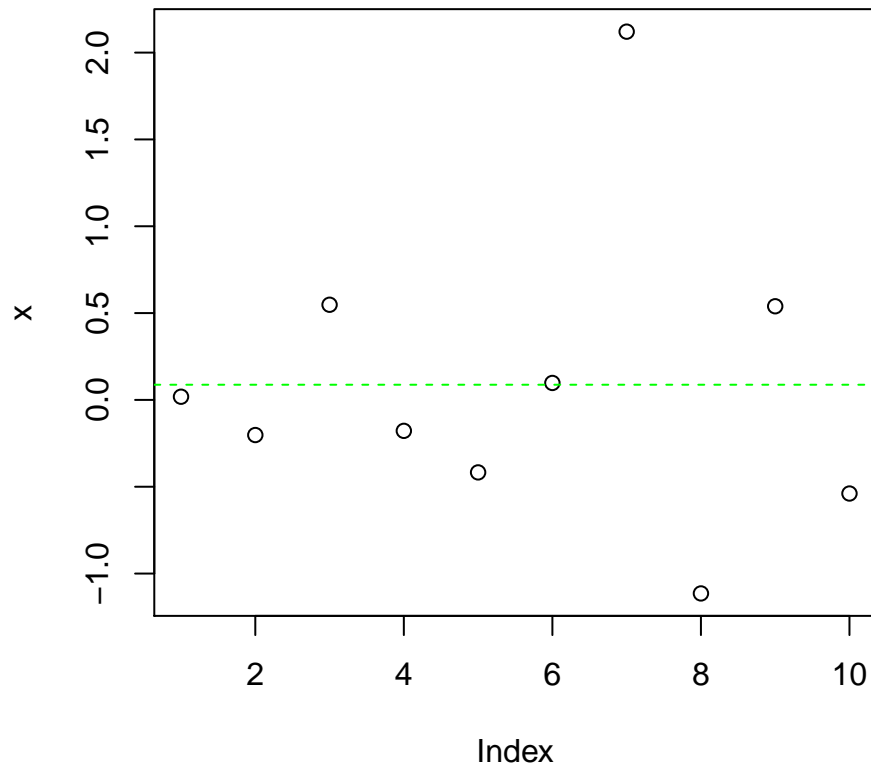
3. Adicionar uma curva com a estimativa da densidade da distribuição representada.

```
hist(iris$Petal.Width, main = "Histograma da Largura da Pétala", xlab = "",  
     ylab = "Probabilidade", prob = T)  
lines(density(iris$Petal.Width))
```



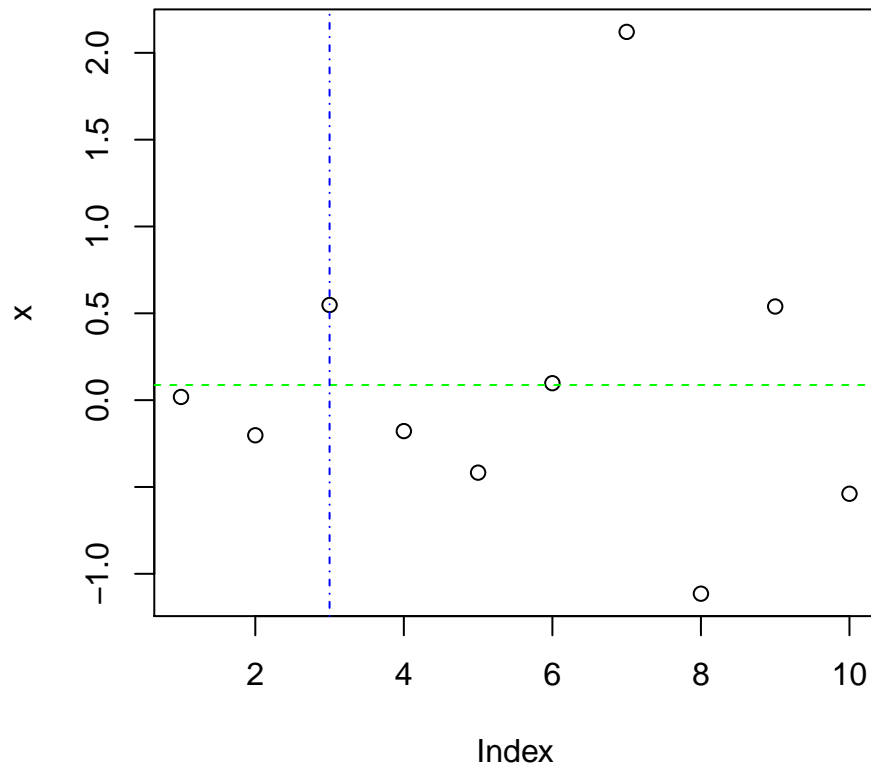
4. Desenhar uma linha horizontal, representando a média dos pontos no gráfico.

```
x = rnorm(10)
plot(x)
abline(h = mean(x), col = "green", lty = 2)
```



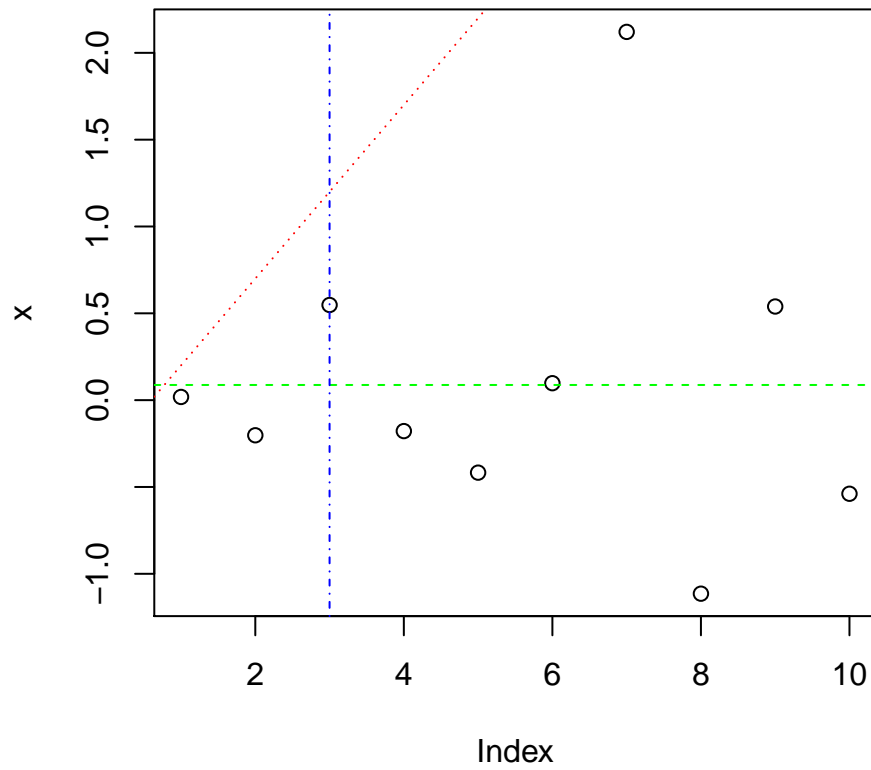
5. Agora uma linha vertical.

```
abline(v = 3, col = "blue", lty = 4)
```



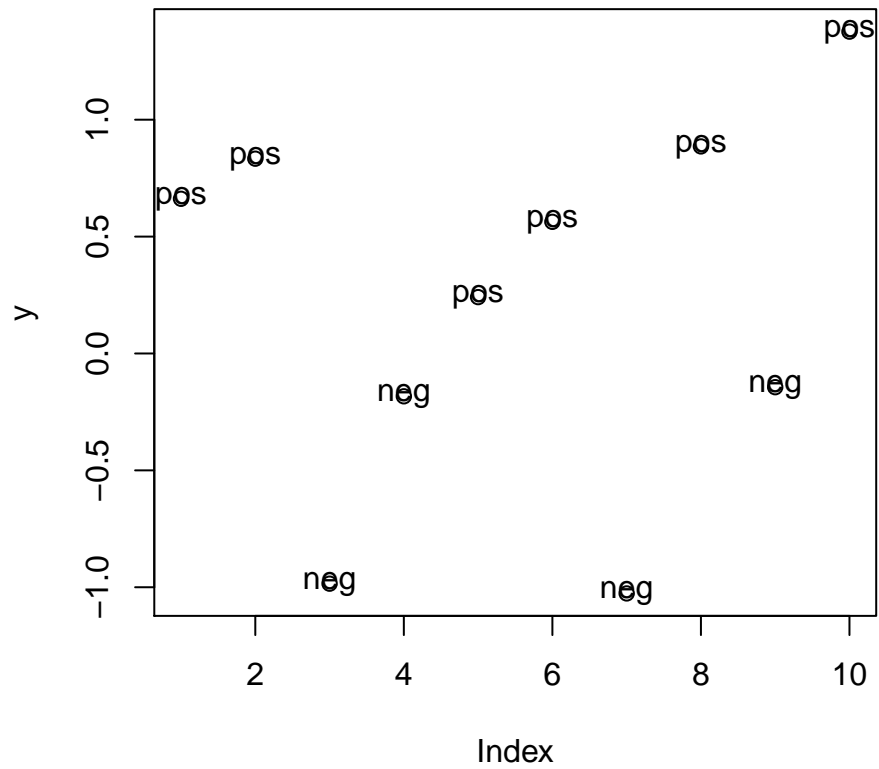
6. A mesma função também suporta serve para representar linhas diagonais.

```
abline(-0.3, 0.5, col = "red", lty = 3)
```



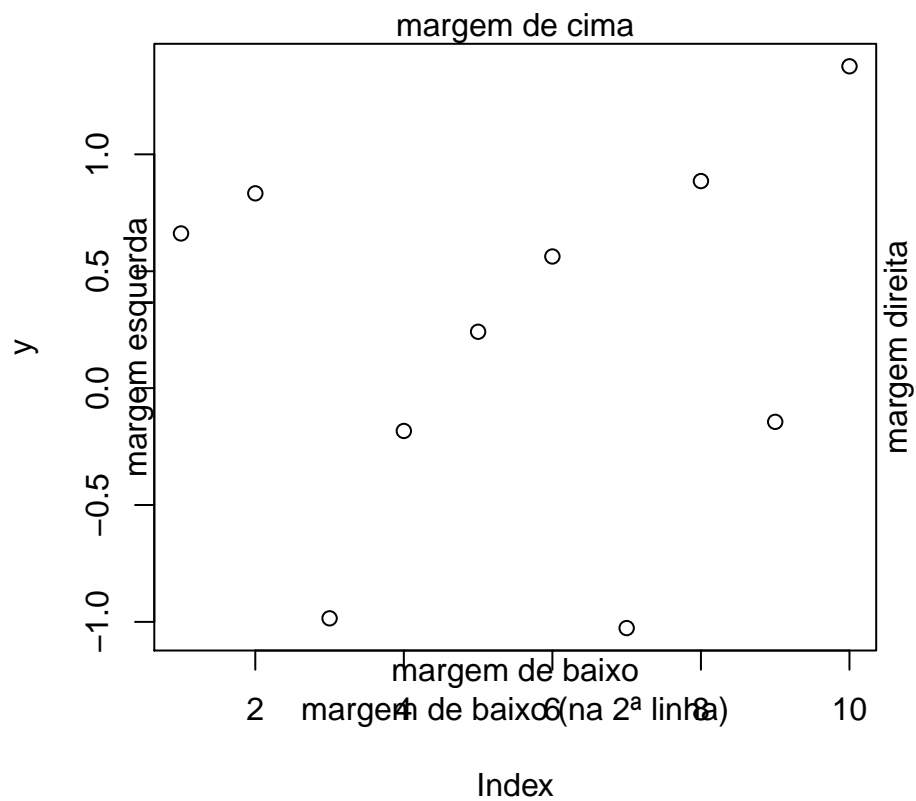
7. Adicionar texto em cima dos pontos do gráfico, de acordo com o respectivo valor.

```
y = rnorm(10)
plot(y)
text(1:10, y, ifelse(y > 0, "pos", "neg"))
```



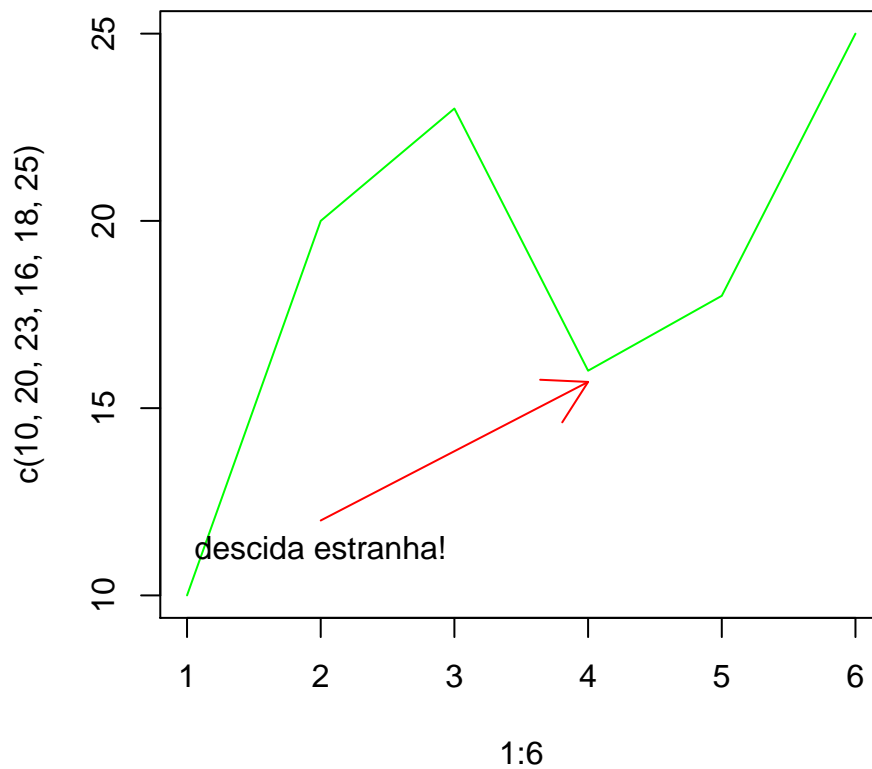
8. Colocar texto nas margens do gráfico.

```
plot(y)
mtext("margem de baixo", side = 1)
mtext("margem de baixo (na 2ª linha)", side = 1, line = 1)
mtext("margem esquerda", side = 2)
mtext("margem direita", side = 4)
mtext("margem de cima", side=3)
```



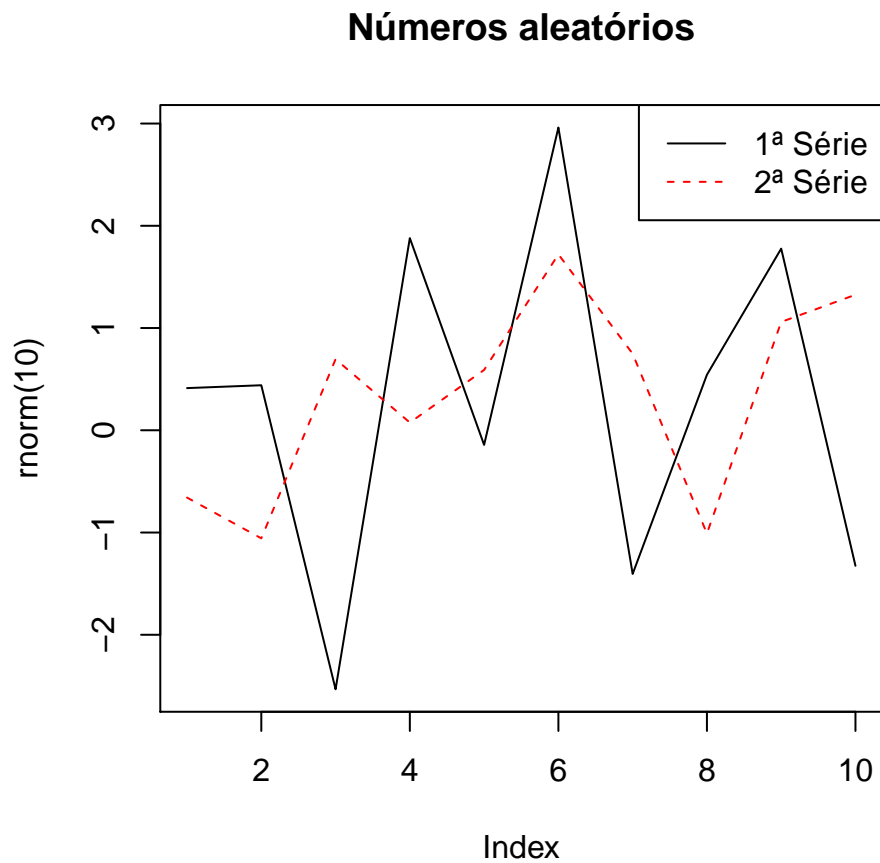
9. Adicionar uma seta, seguida de texto, a um gráfico de linhas.

```
plot(1:6, c(10, 20, 23, 16, 18, 25), type = "l", col = "green")
arrows(2, 12, 4, 15.7, col = "red")
text(2, 12, "descida estranha!", pos = 1)
```



10. Adicionar título e legendas a um gráfico com duas variáveis representadas em linhas.

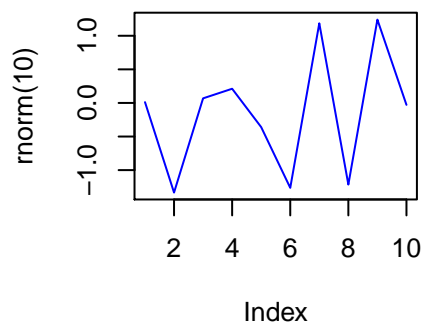
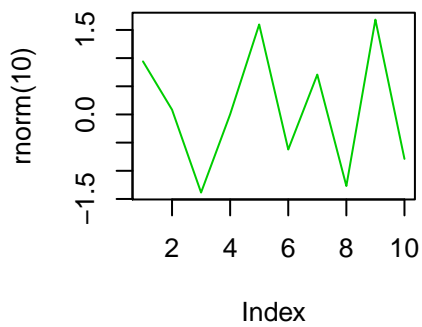
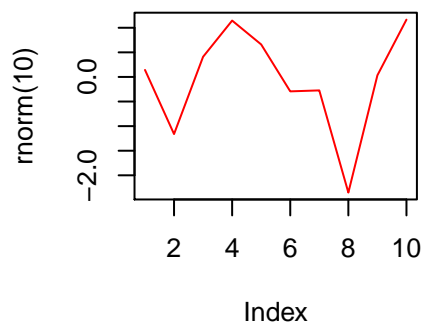
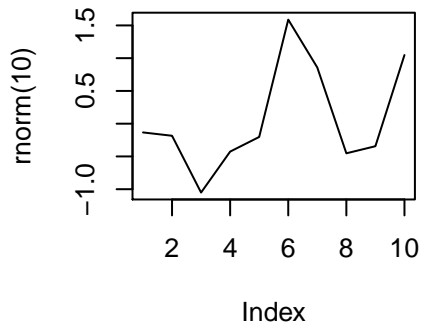
```
plot(rnorm(10), type = "l")  
lines(rnorm(10), col = "red", lty = 2)  
title("Números aleatórios")  
legend("topright", c("1ª Série", "2ª Série"), lty = 1:2, col = 1:2)
```



5 Configurar a janela

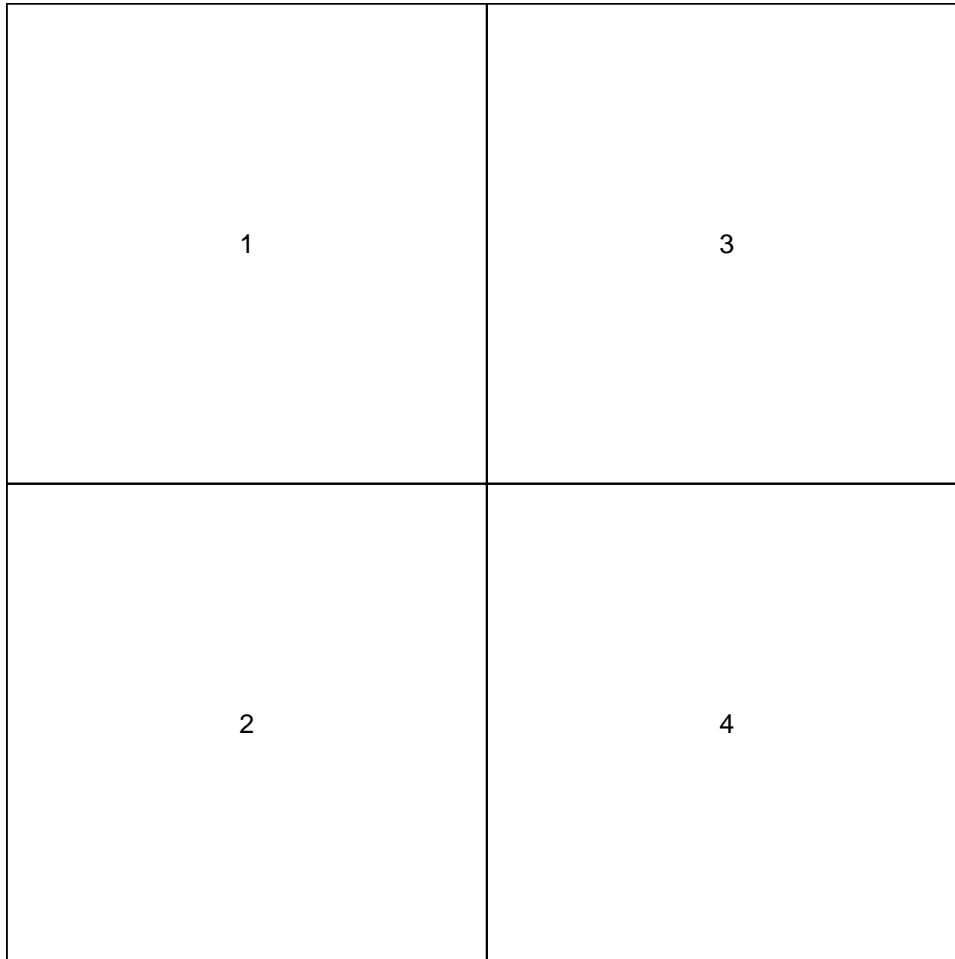
1. Dividir a janela em quatro partes iguais, colocando um gráfico de linhas em cada uma delas.

```
par(mfrow = c(2, 2))  
for (i in 1:4) plot(rnorm(10), col = i, type = "l")
```



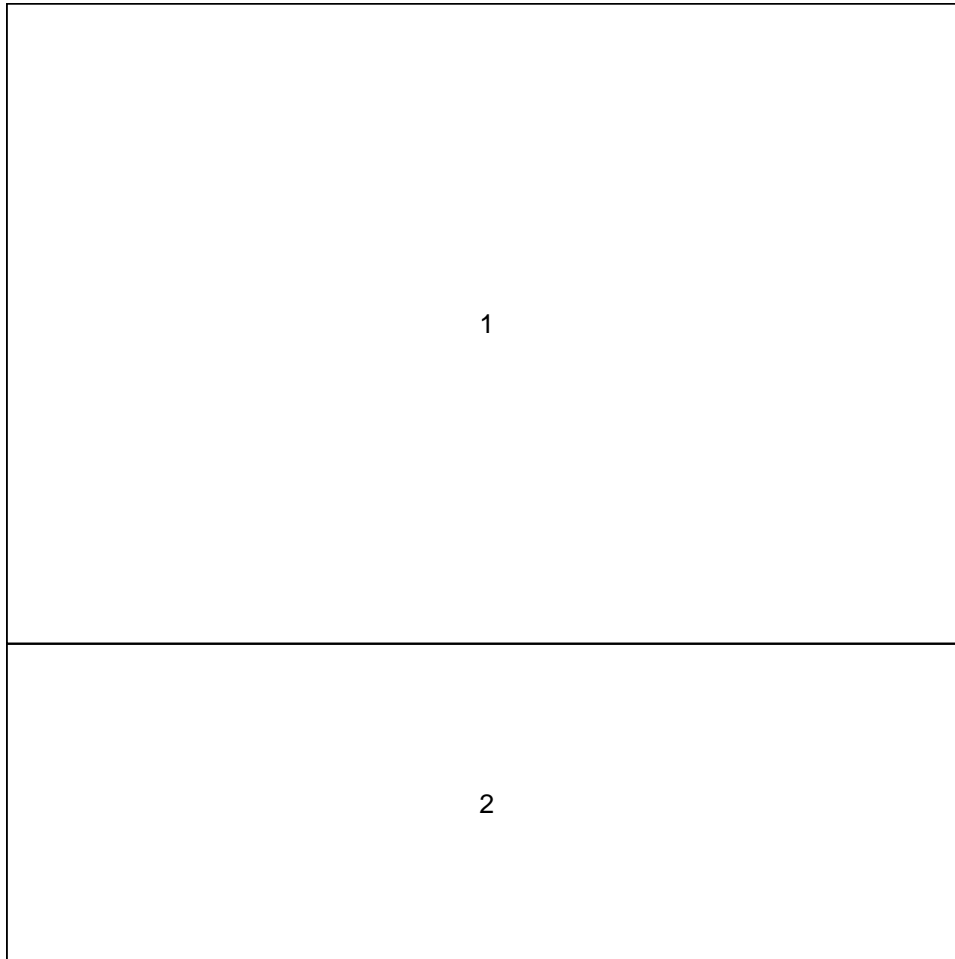
2. Outra forma de dividir a janela em quatro partes iguais, mostrando, de seguida, uma pré-visualização da disposição criada.

```
layout(matrix(1:4, 2, 2))  
layout.show(4)
```



3. A função anterior também permite criar divisões com diferentes dimensões, por exemplo duas divisões, em que a primeira tem o dobro do tamanho da segunda.

```
layout(matrix(1:2, 2, 2), heights = c(2, 1))  
layout.show(2)
```



```
plot(rnorm(10), type="l", col="gray")  
plot(rnorm(10), type="l", col="purple")
```

