

**UNIVERSIDADE DA REGIÃO DE JOINVILLE - UNIVILLE**

Bacharelado em Engenharia de Software (BES)

## **Estatística para computação**

**Professora Priscila Ferraz Franczak**

Engenheira Ambiental - UNIVILLE

Mestre em Ciência e Engenharia de Materiais - UDESC

Doutora em Ciência e Engenharia de Materiais - UDESC

[priscila.franczak@gmail.com](mailto:priscila.franczak@gmail.com)

# Plano de Aula

1. Criando gráficos com o R
2. Uso do comando plot( )
3. Histogramas
4. Exercícios



# 1. Criando gráficos com o R

- Na estatística, em especial, o R possibilita a criação de histogramas, ogivas, *boxplots*, curvas de distribuições, regressões e muito mais.

# Conceitos básicos

Os comandos gráficos do R podem ser divididos em três categorias:

- Comandos de alto nível, que criam gráficos completos.
- Comandos de baixo nível, que adicionam informações à algum gráfico já existente.
- Comandos interativos, que permitem que o usuário interaja com a janela gráfica.



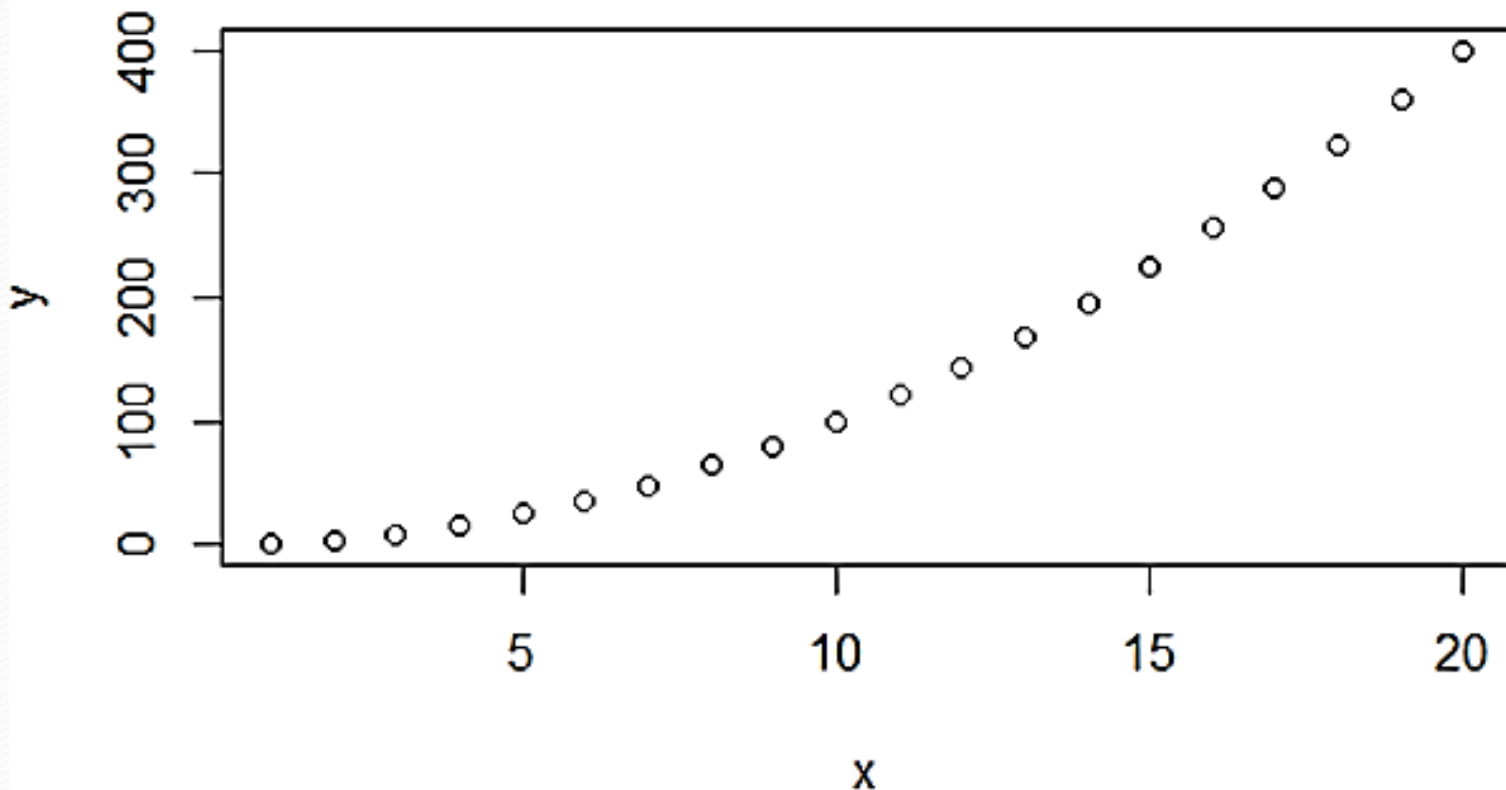
## 2. Uso do comando `plot( )`

### Um gráfico simples

- O `plot( )` é o comando mais simples para a criação de gráfico no R.
- É possível criar desde simples gráficos de dispersão até aqueles com imagens de satélites.

- Em sua forma mais simples, o comando recebe valores de coordenadas para plotar nas abcissas e ordenadas:

```
> x<-1:20  
> y<-x^2  
> plot(x,y)
```



- O comando `plot( )` tem inúmeros argumentos que permitem personalizar o gráfico de acordo com a necessidade.
- O argumento `type`, por exemplo, é usado para definir como os pontos de coordenadas descritas pelos valores no eixo das abcissas e no eixo das ordenadas (x e y no nosso exemplo) são desenhados.



Type pode assumir diferentes caracteres,  
dentre eles:

- “l” – segmentos de reta são usados para ligar os pontos;
- “b” – tanto segmentos de retas quanto pontos são desenhados;
- “o” – o mesmo que o anterior, mas as retas tocam os pontos;
- “c” – como “b” mas sem os pontos;
- “n” – cria um gráfico vazio, omitindo pontos e segmentos de reta.

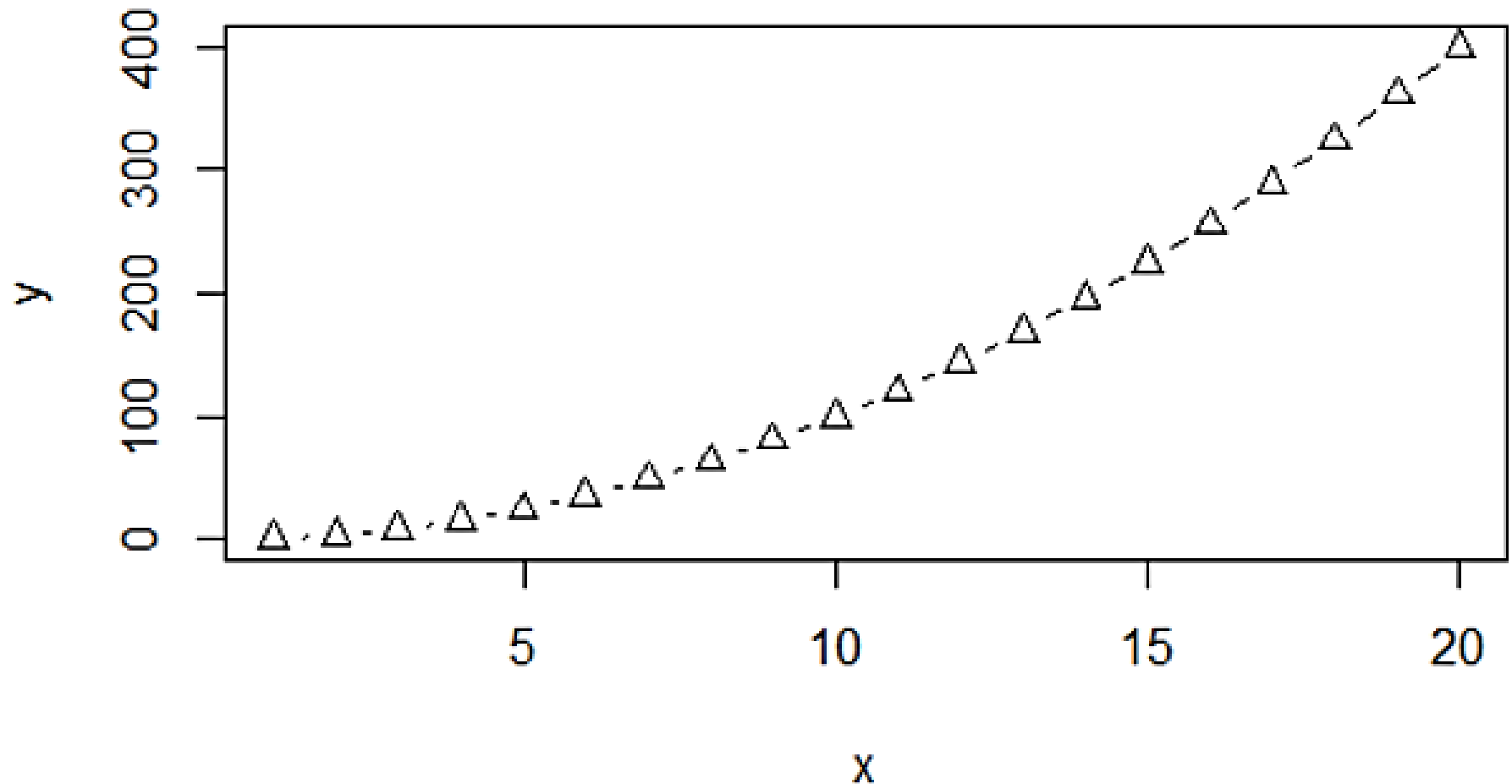


## Alterando o padrão dos pontos

- Além do `type`, outro argumento bastante utilizado no comando `plot( )` é o `pch`, que pode ser usado para mudar o padrão dos pontos.

Exemplo:

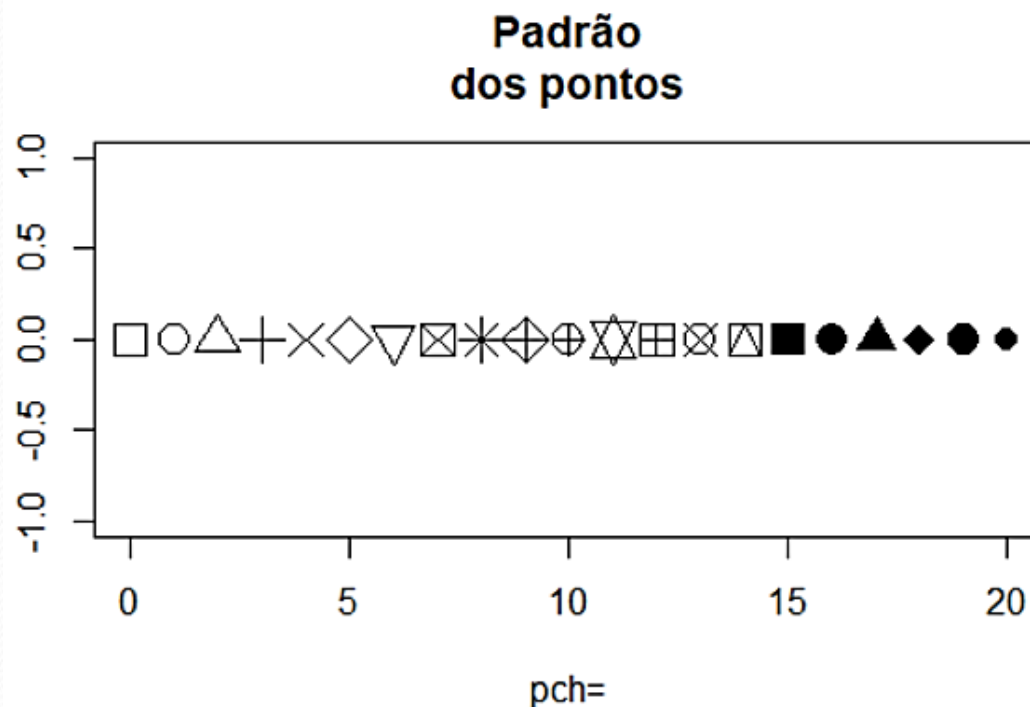
```
> plot(x,y,type="b",pch=2)
```





- Observe o comando e a figura que ele gera:


```
> plot(0:20,          #coord. eixo das abscissas
+      rep(0,21),      #coord. eixo das ordenadas
+      pch=0:20,        #padrão dos pontos variando
+      cex=2,           #tamanho dos pontos
+      main = "Padrão\ndos pontos", #título (note o \n)
+      xlab = "pch=",    #texto do eixo das abscissas
+      ylab="")          #sem texto nas ordenadas
```



O padrão dos pontos, representados por números no argumento `pch`, são:

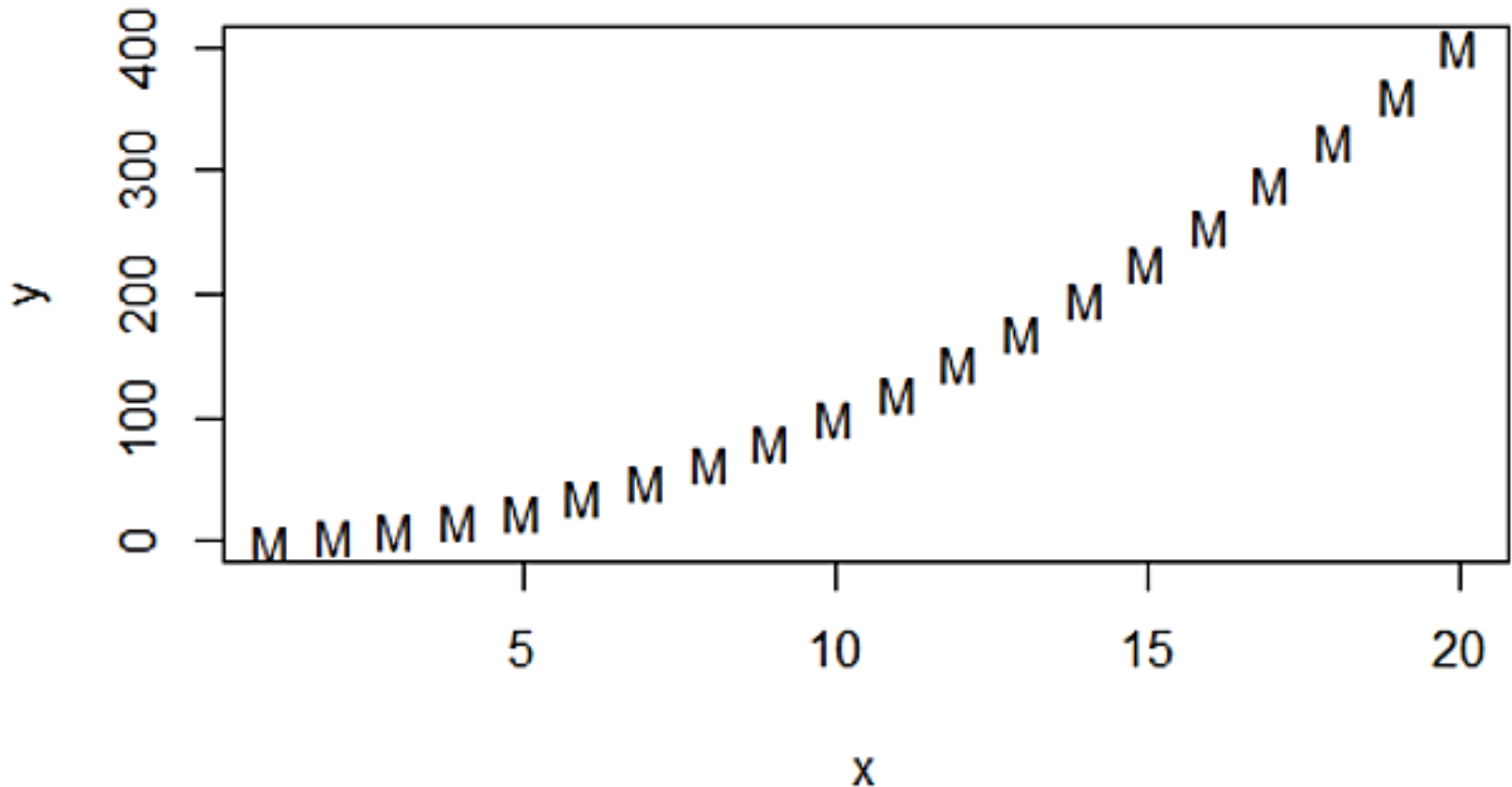
- Números de 0 a 6 mostram os padrões básicos;
- De 7 a 14 são composições de padrões obtidos por sobreposição dos padrões básicos;
- E de 15 a 17 são versões sólidas dos padrões de 0 a 2.



- 
- Além dos padrões definidos por números (desde `pch=0` até `pch=20`), ainda há a possibilidade de utilizar um caractere qualquer como padrão dos pontos.

Exemplo:

```
> plot(x,y,pch="M")
```





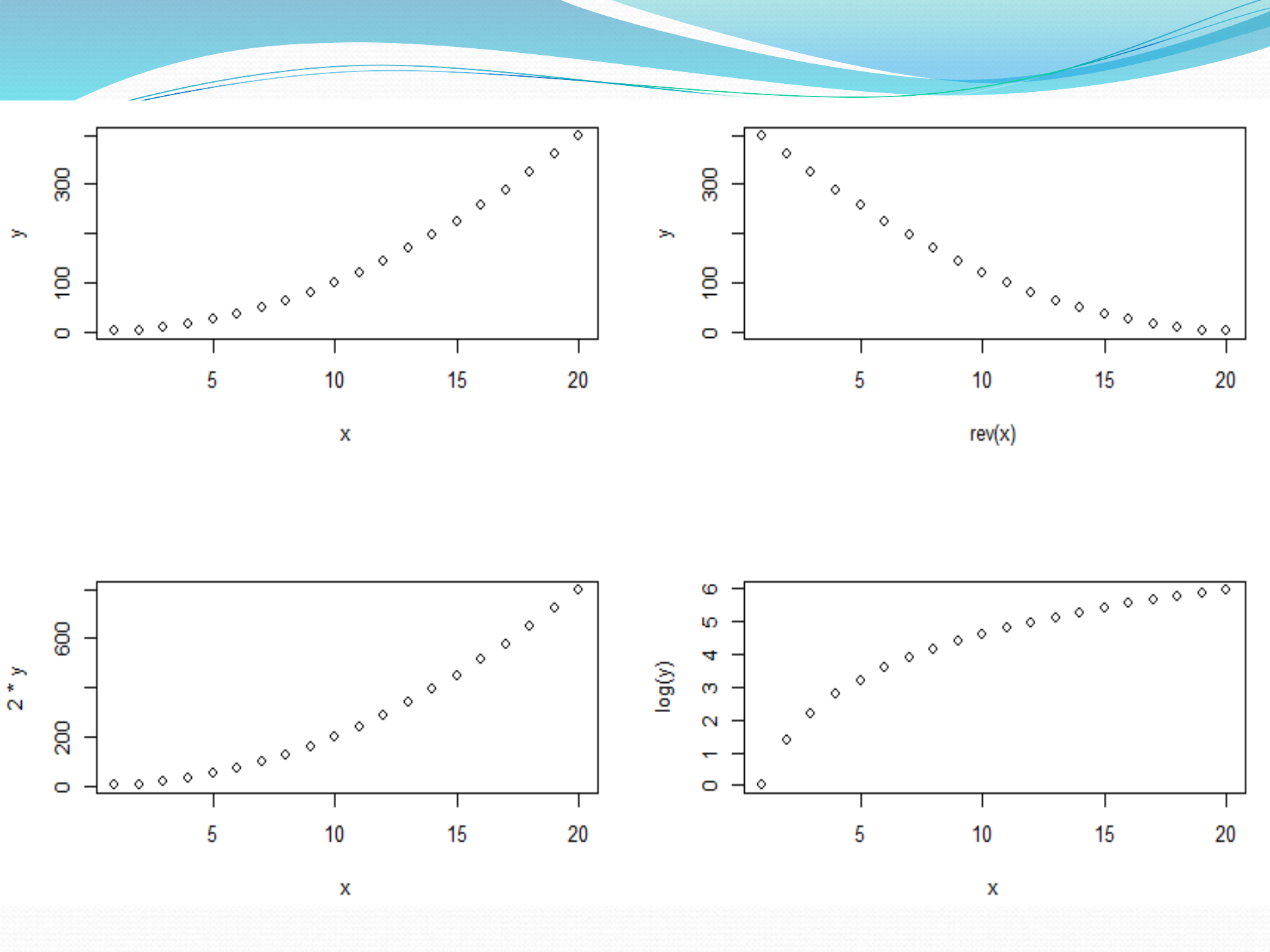
## Vários gráficos na mesma janela gráfica

- É possível fazer com que o R mostre diversos gráficos em uma mesma janela gráfica ao invés de um gráfico em cada janela.
- Para isso, use o comando `par()`, juntamente com o argumento `mfrow`.

```
> plot(x,y,pch="M")
> par(mfrow=c(2,2))      #arranjo "2 por 2"
> plot(x,y)              #gráfico 1
> plot(rev(x),y)         #gráfico 2
> plot(x,2*y)            #gráfico 3
> plot(x,log(y))         #gráfico 4
```

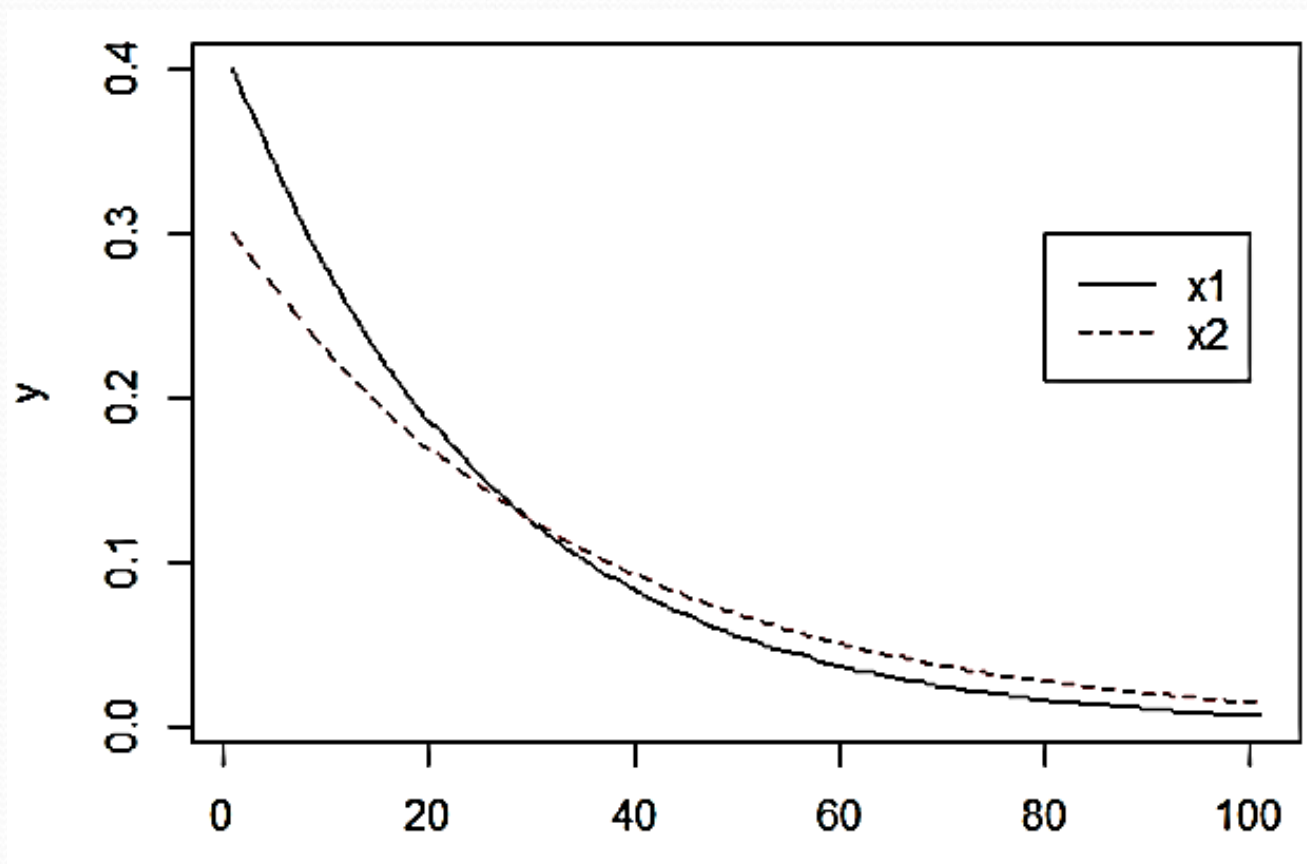
- O primeiro número dentro do `c( )` no argumento `mfrow` informa o número de divisões horizontais e o segundo número indica o número de divisões verticais na janela gráfica.





- Para apresentar várias curvas num único gráfico, cada uma originada de uma coluna de uma matriz, use o comando `matplot( )`.
- Legendas podem ser adicionadas aos gráficos utilizando o comando `legend( )`.

```
> x<-seq(0,10,0.1)
> x1<-0.4*exp(-0.4*x)
> x2<-0.3*exp(-0.3*x)
> y<-cbind(x1,x2)
> matplot(y,type="l")
> legend(80,0.3,c("x1","x2"),lty=c(1,2),col=c(1,2))
```





# Personalizando gráficos

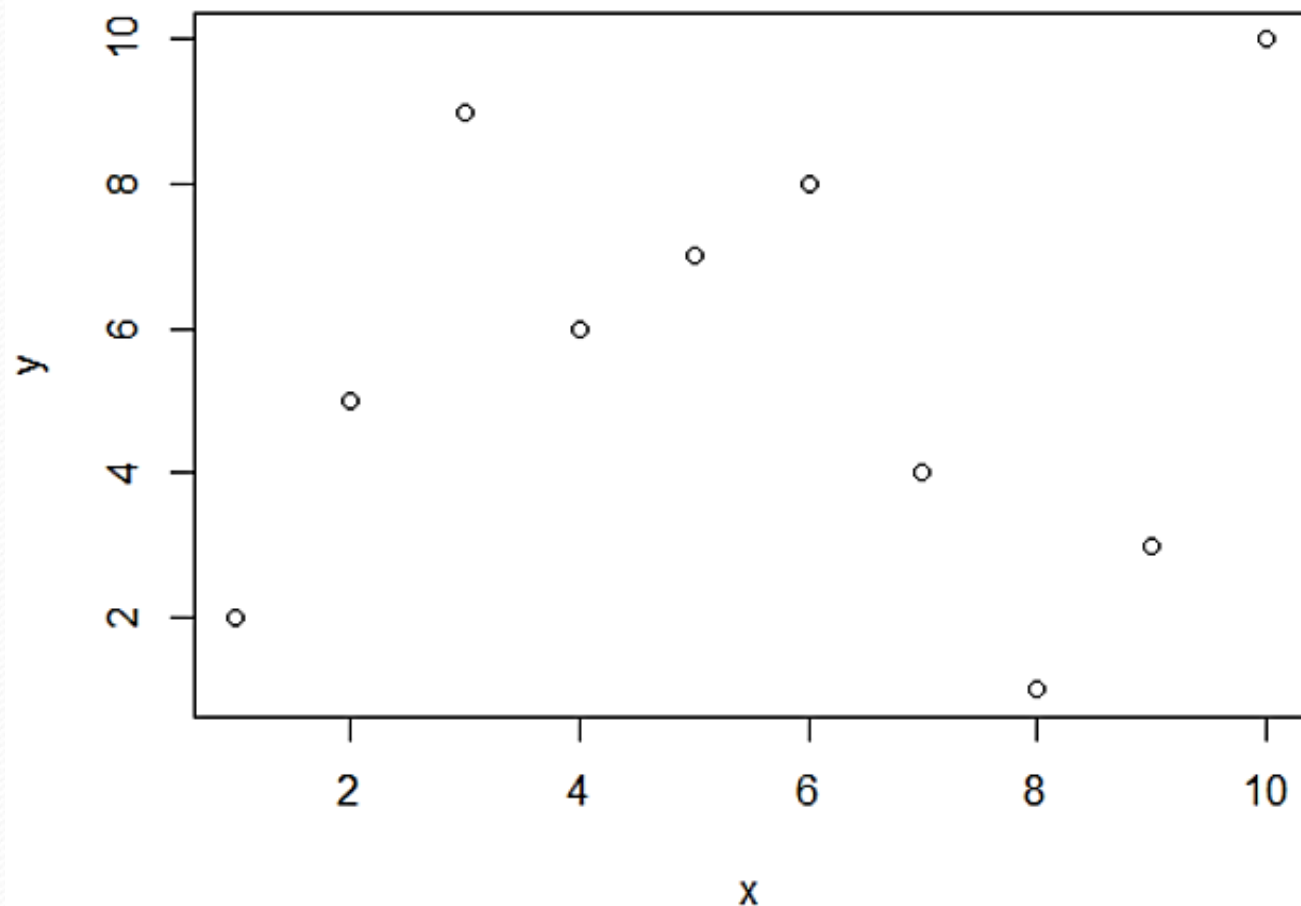
- Exemplos:

Vamos criar dois conjuntos de 10 números cada um.

```
> x<-1:10
> y<-c(2,5,9,6,7,8,4,1,3,10)
> x;y
[1] 1 2 3 4 5 6 7 8 9 10
[1] 2 5 9 6 7 8 4 1 3 10
> |
```

- 1º gráfico:

```
> plot(x,y) #plota x e y
```

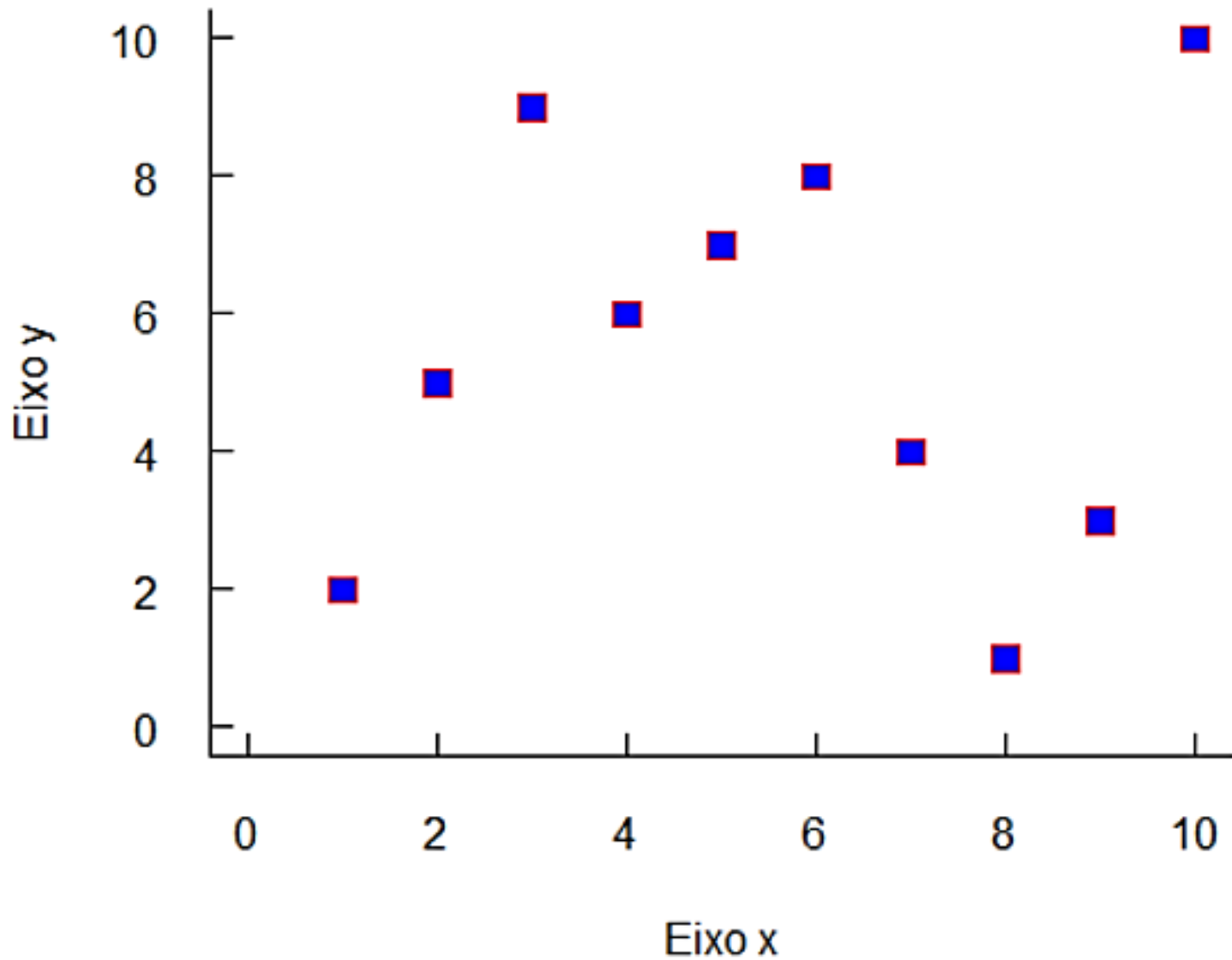


- 2º gráfico:

```
plot(x,y,                                #plota x e y
     xlab = "Eixo x", #nomeia o eixo x
     ylab = "Eixo y", #nomeia o eixo y
     main = "Personalizando um gráfico", #título
     xlim = c(0,10), #limites do eixo x
     ylim = c(0,10), #limites do eixo y
     col = "red",     #cor dos pontos
     pch = 22,        #formato dos pontos
     bg = "blue",     #cor de preenchimento
     tcl = 0.4,       #tamanho de traços dos eixos
     las=1,           #orientação dos valores nos eixos
     cex=1.5,         #tamanho do ponto
     bty="l")         #altera as bordas
```



## Personalizando um gráfico



### 3. Histogramas

- A ideia deste gráfico é categorizar uma variável quantitativa, dividindo-a em intervalos ou classes.
- Contar quantos valores se encaixam em cada intervalo e construir um gráfico de colunas com o resultado.



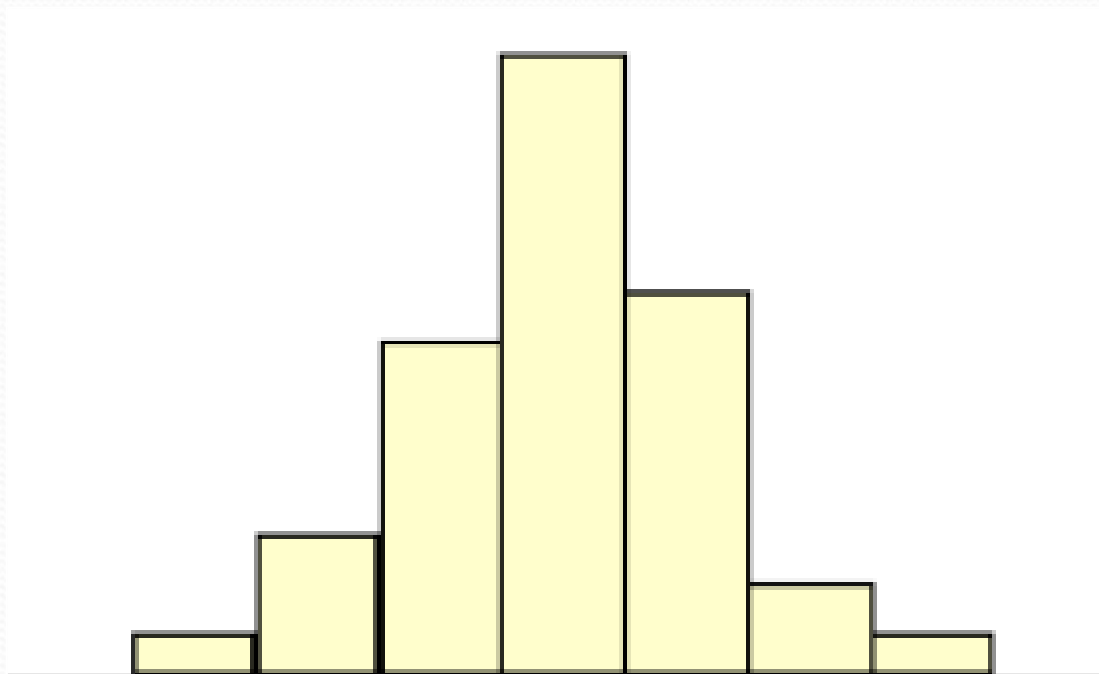
Ao se interpretar um histograma, deve-se tentar responder às seguintes questões:


- Qual é a forma da distribuição dos dados?
- Existe um ponto central bem definido?
- Como é a amplitude de variação dos dados?
- Existe apenas um pico isolado?
- A distribuição é simétrica?



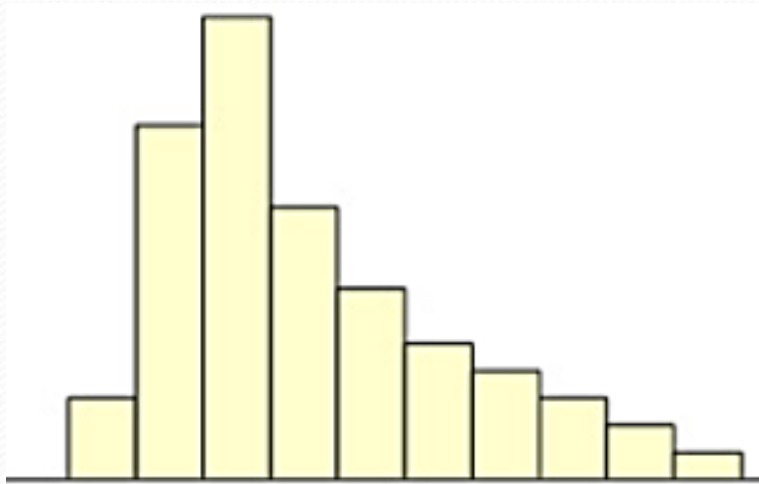
# Tipos de histogramas

- **Histograma simétrico:** A frequência de dados é mais alta no centro e decresce gradualmente à esquerda e à direita de forma aproximadamente simétrica, em forma de sino.

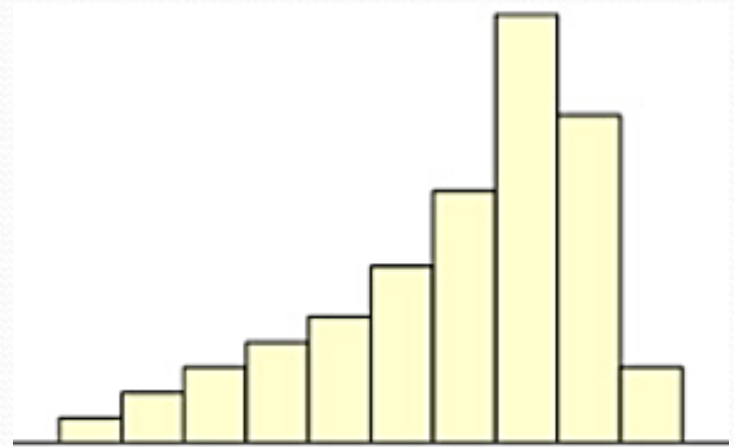


- 
- **Histograma fortemente assimétrico:** A frequência dos dados decresce rapidamente num dos lados e muito lentamente no outro, provocando uma **assimetria** na distribuição dos valores.

- A distribuição dos salários numa empresa é um exemplo comum de histograma assimétrico: **muitas pessoas ganham pouco e poucas pessoas ganham muito (A)**. A situação (B), apesar de mais rara, também pode acontecer.




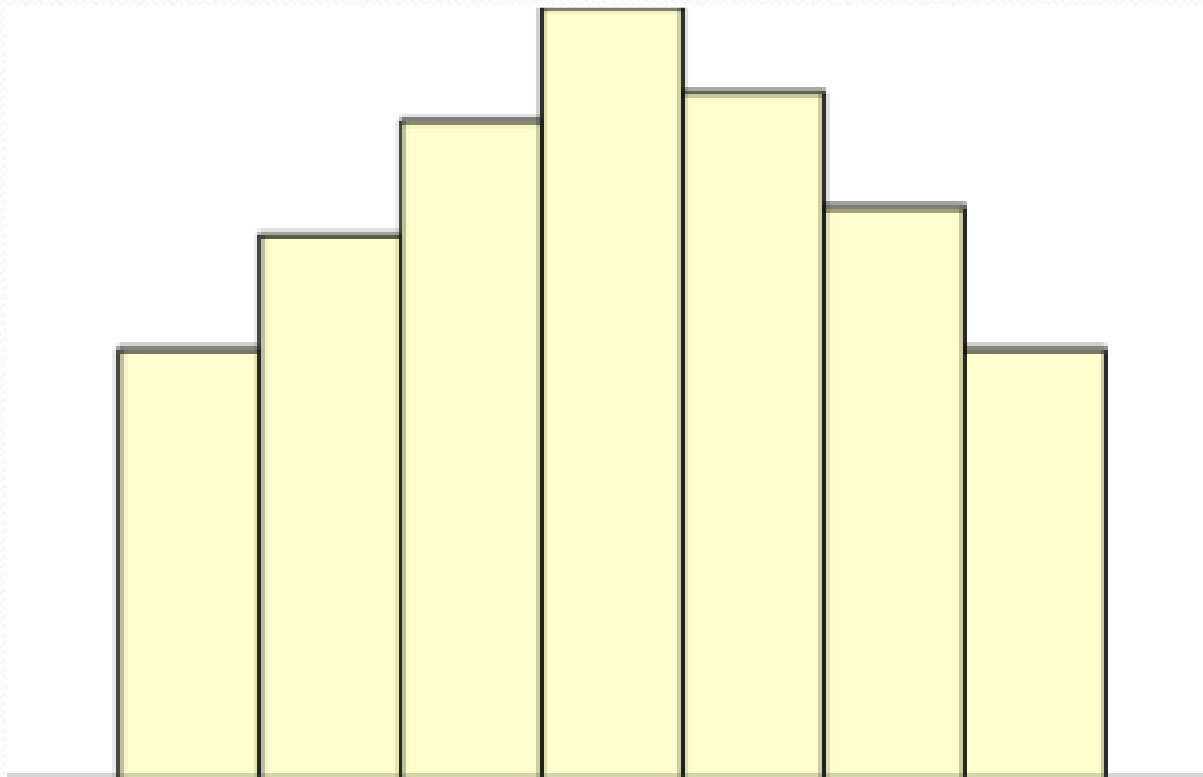
A



B

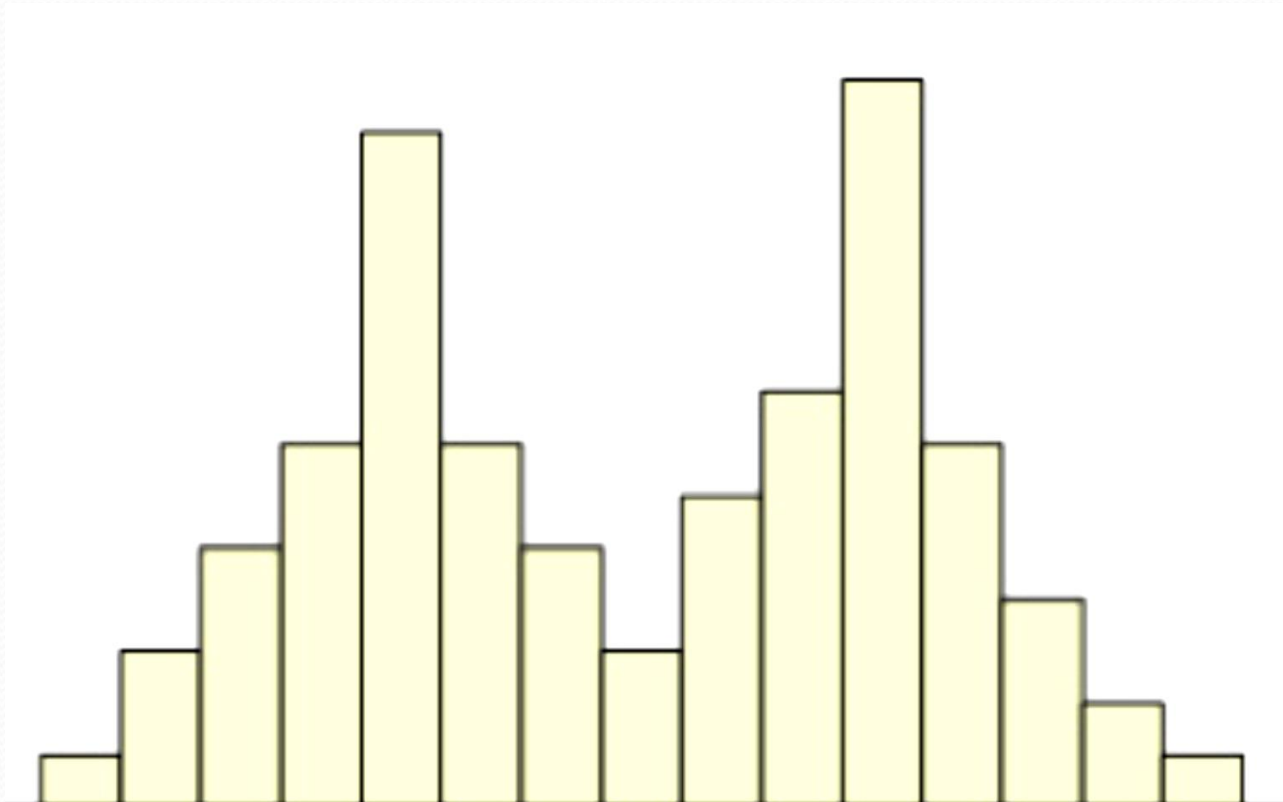



- 
- **Histograma tipo despenhadeiro:** O histograma termina abruptamente em um ou nos dois lados, dando a impressão de que faltam dados.
  - Possivelmente os dados muito pequenos e/ou muito grandes foram eliminados da amostra.

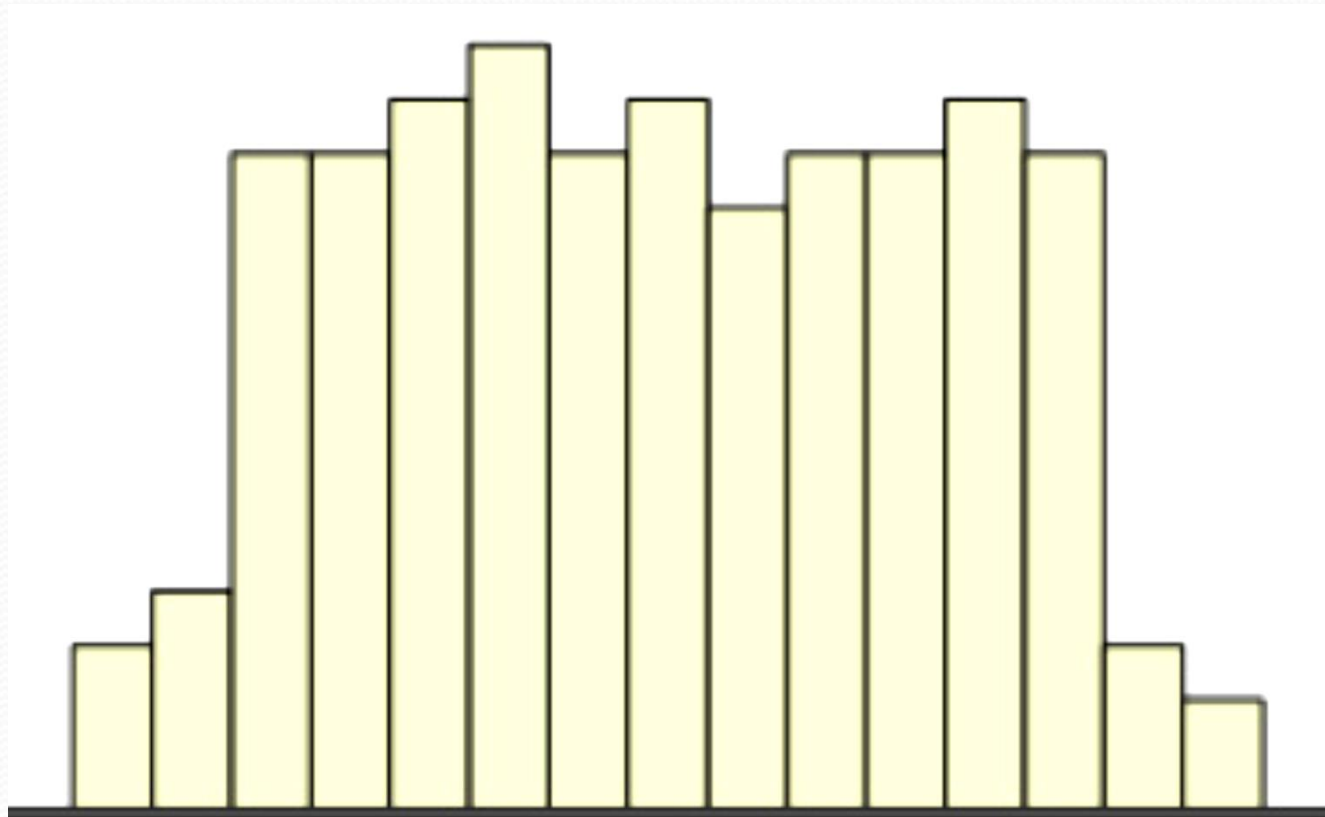


- **Histograma com dois picos:** Ocorrem picos na distribuição e a frequência é baixa entre os picos.
- Possivelmente, os dados se referem a uma mistura de valores de diferentes populações, devendo ser avaliados com cuidado.
- Se houve mistura dos dados, é melhor separá-los.



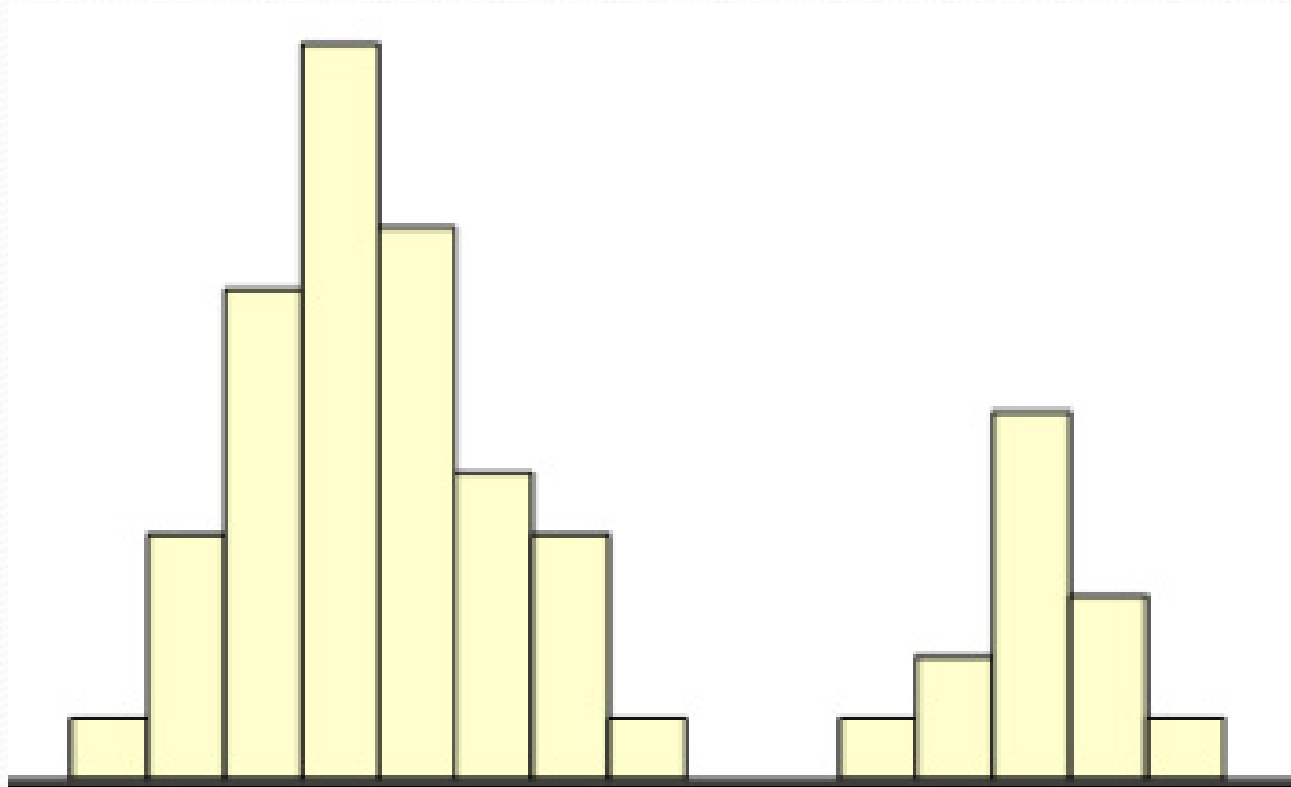


- 
- **Histograma tipo platô:** As classes de valores centrais apresentam aproximadamente a mesma frequência.
  - Essa situação também sugere mistura de valores de diferentes populações.

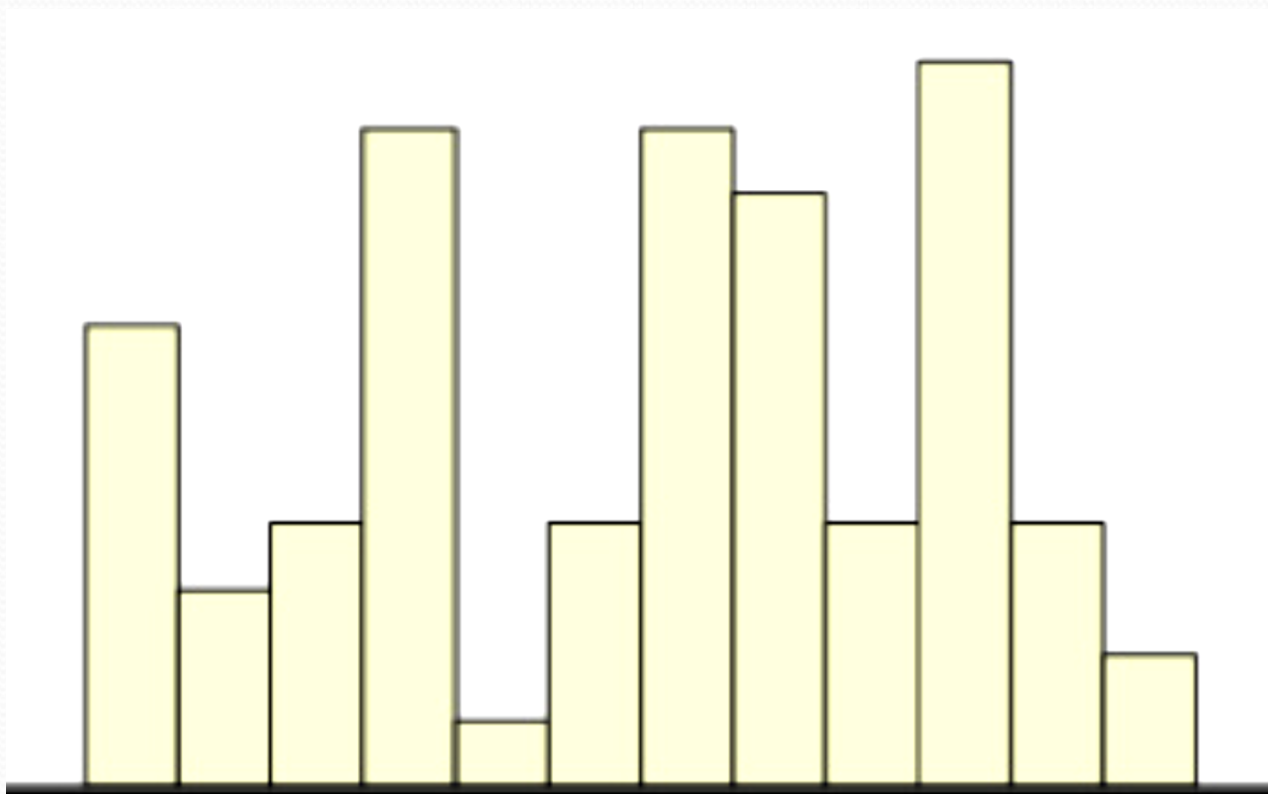




- **Histograma com uma pequena ilha isolada:**  
Alguns valores isolados têm frequência elevada, formando uma espécie de ilha.
- Também pode ter ocorrido uma mistura de dados.



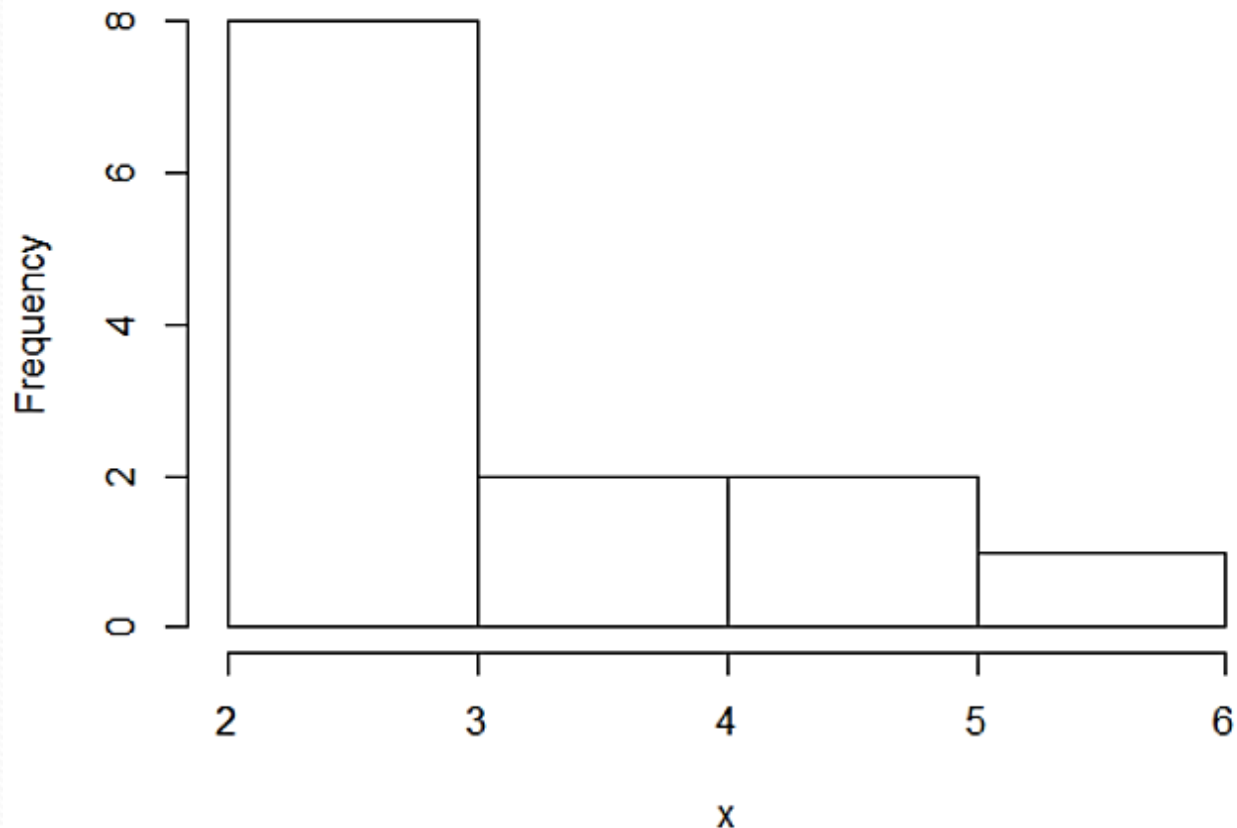
- **Histograma tipo serrote:** As frequências de valores se alternam formando vários dentes.
- Pode indicar algum problema na obtenção (leitura) dos dados.



- O comando `hist( )` produz um histograma dos dados informados em seu argumento:

```
> x<-c(2,2,2,2,2,3,3,3,4,4,5,5,6) #vetor qualquer  
> hist(x)
```

Histogram of x





- Para auxiliar na interpretação do histograma, pode-se usar:

```
> table(x)  #valores de x e suas frequências
x
2 3 4 5 6
5 3 2 2 1
```

- Observe que a coluna de 2 a 3 do histograma indica que há oito elementos nessa classe.
- Vale frisar, entretanto, que o padrão do comando `hist( )` considera intervalos de classes fechados a direita (`right=TRUE`), ou seja, o 3 também está incluído na primeira classe do histograma acima, sendo o intervalo  $(2,3]$ .

- Então, por que o 2 foi contabilizado se o intervalo é aberto em 2?
- Isso acontece por causa do argumento `include.lowest`, que, por *default*, é definido como `TRUE`.
- Isso inclui o primeiro valor do vetor na primeira **classe**, quando os intervalos de classes são fechados a direita e o último quando fechados a esquerda.



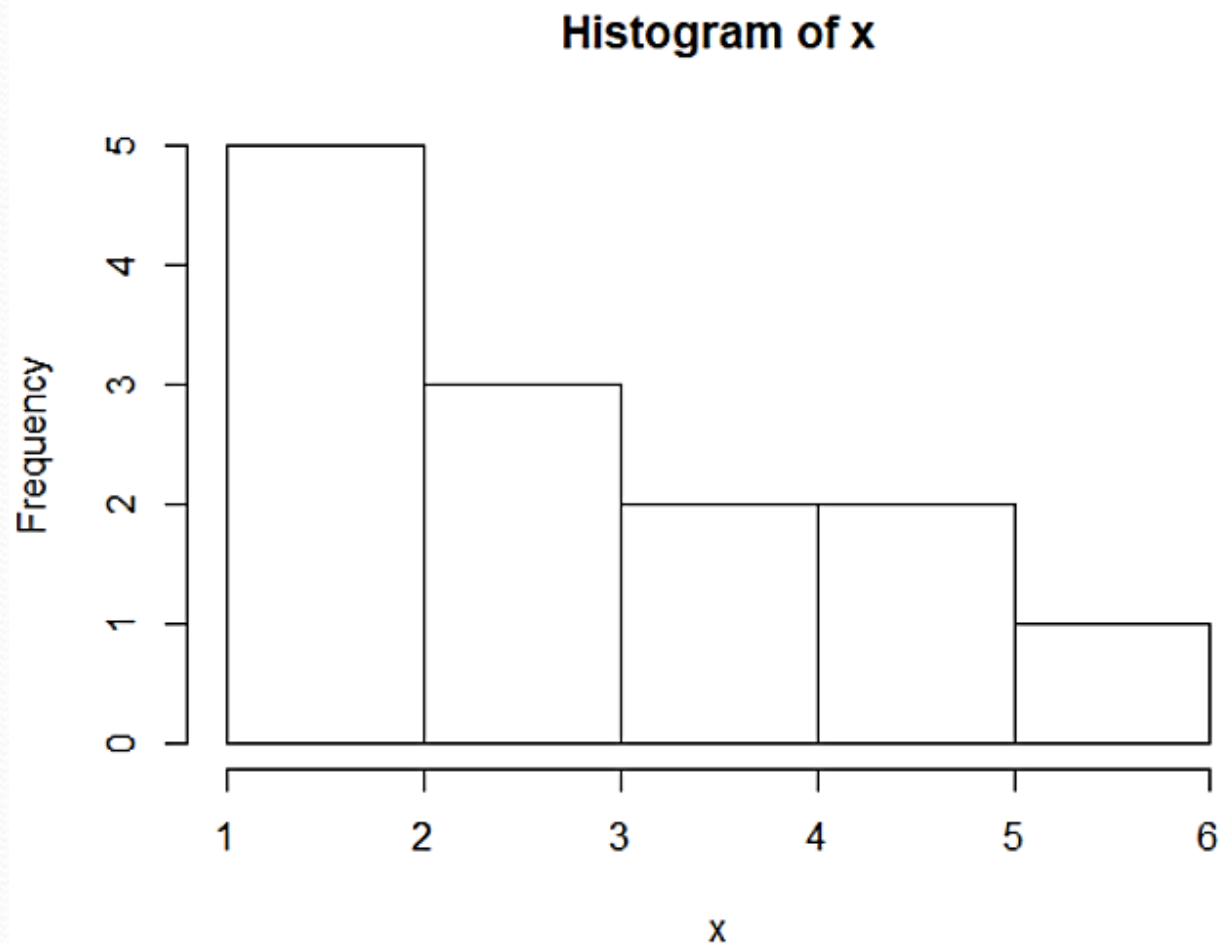
- 
- Outro argumento importante é o breaks, que define os intervalos a serem usados no eixo das abscissas do histograma.



# Personalizando histogramas

- O comando que gera histogramas pode ser manipulado de forma que, ao combinarmos os argumentos **right**, **include.lowest** e **breaks**, entre outros, podemos gerar histogramas da maneira que quisermos.

```
hist(x,      #histograma de x  
     right = T, #intervalos fechados à direita  
     include.lowest = F, #não soma extremos do vetor  
     breaks = 1:6) #intervalos de classes
```



## Exemplo:

- Suponha um conjunto de dados, coletados por um professor, que se refere ao tempo gasto (em minutos) pelos alunos para a resolução de um problema de álgebra.

25 27 18 16 21 22 21 20 18 23 27 21 19 20 21 16

- Construa um histograma do conjunto de dados com intervalos fechados à esquerda.



25 27 18 16 21 22 21 20 18 23 27 21 19 20 21 16

## Dados agrupados sem intervalo de classe

<b>Tempo (em minutos)</b>	<b>Frequência</b>
16	2
18	2
19	1
20	2
21	4
22	1
23	1
25	1
27	2



Como agrupar dados com intervalo  
de classe?



Na coluna da esquerda serão representadas as variáveis em **intervalos de classe**. Os extremos de uma classe são denominados limites da classe e são representados por:

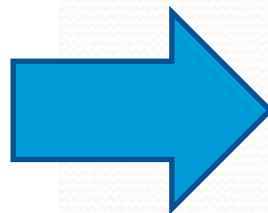
$I_i$  se for limite inferior da classe  $i$  e  $L_i$  é o limite superior da classe  $i$ . No extrato da tabela abaixo podemos verificar um exemplo de classe em que o limite inferior é  $I_i = 2$  e o limite superior é  $L_i = 6$  com frequência de classe  $f_i = 3$

$x_i$	$f_i$
$I_i \mid L_i$	
2 $\mid$ 6 (intervalo de classe)	3



$x_i$	$f_i$
$I_i \mid L_i$	
2   6 (intervalo de classe)	3

Nascimentos $x_i$	Frequências $f_i$
2	1
4	2
6	4
8	5
10	6
12	7
14	8
16	12
18	8
20	8
22	5
24	4
26	3
28	2
30	1
<i>Total</i>	$\sum f_i = 76$



Nascimentos $x_i$	frequência $f_i$
2 $\mapsto$ 6	3
6 $\mapsto$ 10	9
10 $\mapsto$ 14	13
14 $\mapsto$ 18	20
18 $\mapsto$ 22	16
22 $\mapsto$ 26	9
26 $\mapsto$ 30	5
30 $\mapsto$ 34	1
<i>Total</i>	76

O número " $i$ ", de classes ideal para esse problema em geral o pesquisador decide. Em geral, os autores aconselham usar o número de classes entre 5 e 15. Uma regra bastante usada é a regra de STURGES dada por:

$$i = 1 + 3,3\log(n)$$

$$i = 1 + 3,3\log(16)$$

$$i = 4,97$$

$i = 5$  classes

Tempo (em minutos)	Frequência
16	2
18	2
19	1
20	2
21	4
22	1
23	1
25	1
27	2

$n$  = número de amostras

A amplitude da amostra é dada pela diferença entre o maior e o menor valor das variáveis da amostra.

$$H_t = (\text{maior } x_i) - (\text{menor } x_i)$$

$$H_t = (27) - (16)$$

$$H_t = 11$$

Para saber o intervalo de classe  $h_i$  dividimos a amplitude total da distribuição  $H_t$  pelo número  $i$  de classes.

$$h_i = \frac{H_t}{i}$$

$$h_i = \frac{11}{5}$$

$$h_i = 2,2$$

$$h_i = 2$$

Tempo (em minutos)	Frequência
16	2
18	2
19	1
20	2
21	4
22	1
23	1
25	1
27	2

Onde  $h_i = L_i - l_i$



$$h_i = 2$$

Tempo (em minutos)	Frequência
16	2
18	2
19	1
20	2
21	4
22	1
23	1
25	1
27	2
Soma	16

Tempo (em minutos)	Frequência
16   18	2
18   20	3
20   22	6
22   24	2
24   26	1
26   28	2
Soma	16

*Na verdade tenho 6 classes, então terei 6 colunas no histograma*

## Voltando ao Exemplo:

- Suponha um conjunto de dados, coletados por um professor, que se refere ao tempo gasto (em minutos) pelos alunos para a resolução de um problema de álgebra.

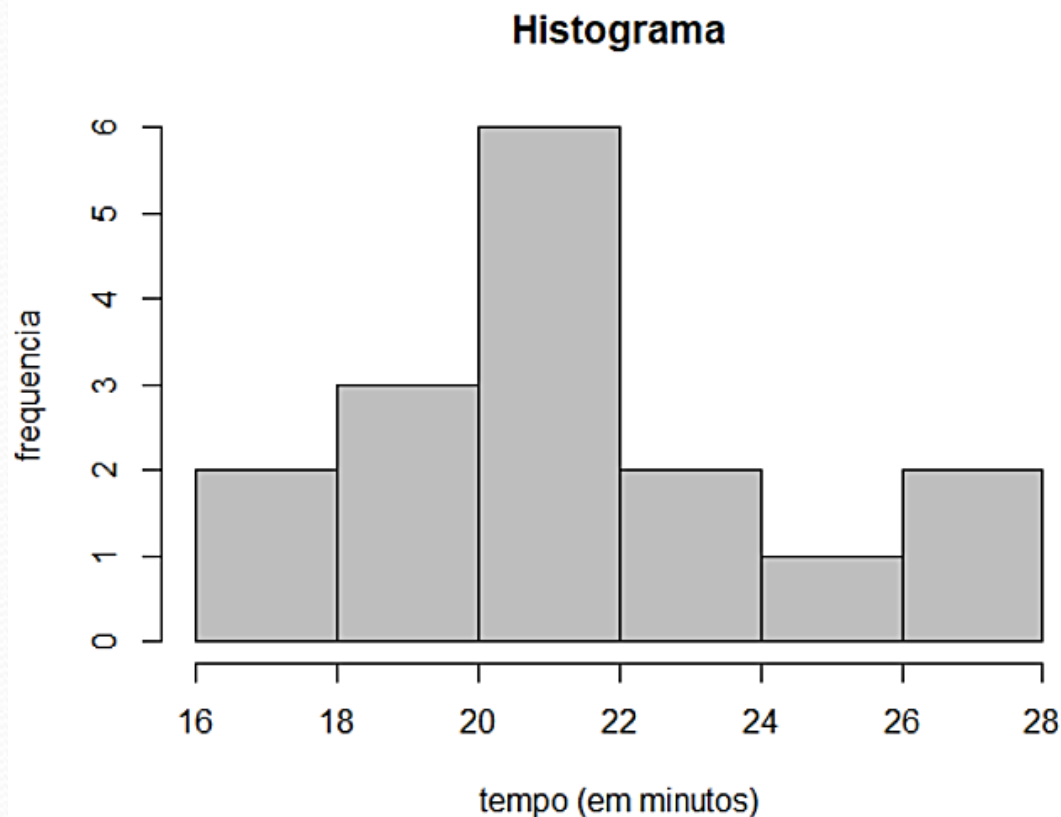
25 27 18 16 21 22 21 20 18 23 27 21 19 20 21 16

- Construa um histograma do conjunto de dados com intervalos fechados à esquerda.

```

dados<-c(25,27,18,16,21,22,21,20,
         18,23,27,21,19,20,21,16)
hist(dados,      #histograma de dados
     nc=6,      #número de classes = 6
     right=F,   #intervalo fechado à esquerda
     main = "Histograma", #título
     xlab = "tempo (em minutos)", #texto eixo x
     ylab = "frequencia", #texto do eixo y
     col=8      #usar a cor cinza nas barras
     )

```



Tempo (em minutos)	Frequência
16   18	2
18   20	3
20   22	6
22   24	2
24   26	1
26   28	2
Soma	16





Caso você não informe o número de colunas, ele distribui as colunas de acordo com os dados.

Para alterar os limites dos eixos é só determinar os valores através dos comandos:

```
ylin = c(0,10)
```

```
xlin = c(130,170)
```



Para ver nomes de cores possíveis, digite `colors( )` no *prompt* do Rstudio e veja as possibilidades.

```
colors()
[1] "white"
[3] "antiquewhite"
[5] "antiquewhite2"
[7] "antiquewhite4"
[9] "aquamarine1"
[11] "aquamarine3"
[13] "azure"
[15] "azure2"
[17] "azure4"
[19] "bisque"
[21] "bisque2"
[23] "bisque4"
[25] "blanchedalmond"
[27] "blue1"
[29] "blue3"
[31] "blueviolet"
[33] "brown1"
[35] "brown3"
[37] "burlywood"
[39] "burlywood2"
[41] "burlywood4"
[43] "cadetblue1"
[45] "cadetblue3"
[47] "chartreuse"
"aliceblue"
"antiquewhite1"
"antiquewhite3"
"aquamarine"
"aquamarine2"
"aquamarine4"
"azure1"
"azure3"
"beige"
"bisque1"
"bisque3"
"black"
"blue"
"blue2"
"blue4"
"brown"
"brown2"
"brown4"
"burlywood1"
"burlywood3"
"cadetblue"
"cadetblue2"
"cadetblue4"
"chartreuse1"
```





## 4. Exercícios