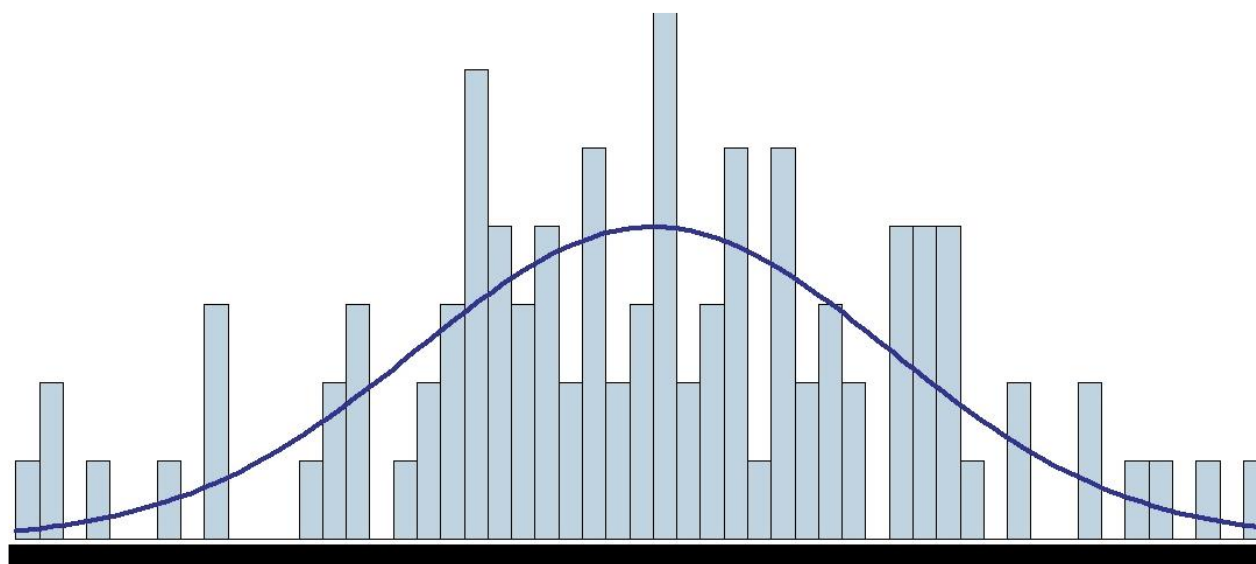




INTRODUÇÃO AO R



PopEA - População, Espaço e Ambiente:
CST-310-3 e SER-457-3 – ano 2018
Dra. Silvana Amaral e Antônio Miguel Vieira Monteiro

INTRODUÇÃO AO R

- R é uma linguagem e ambiente para computação estatística e construção de gráficos.
- É um software livre, na forma de código fonte.
- Roda sobre as principais plataformas computacionais (Linux, Windows e MacOS).



Robert Gentleman

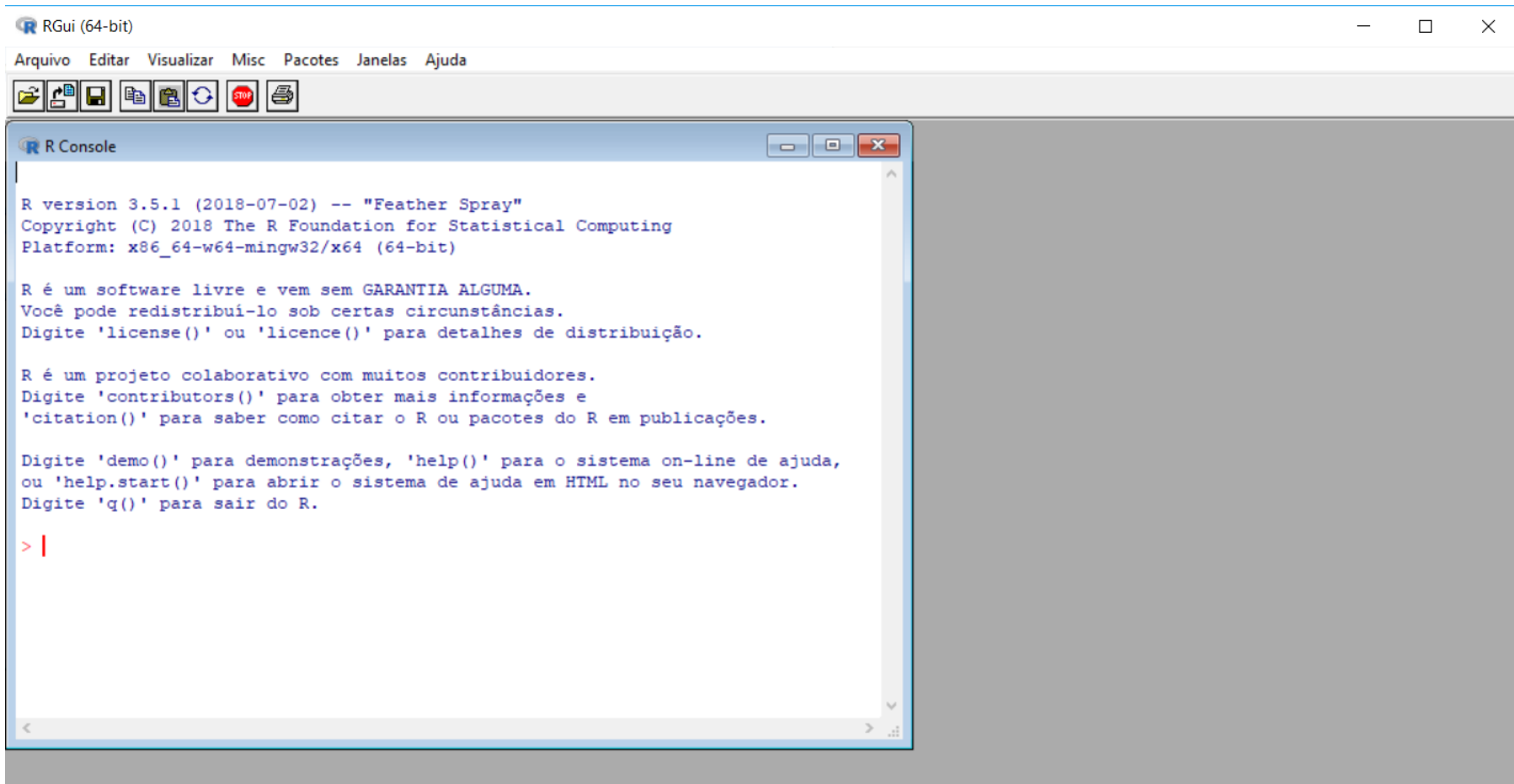
Ross Ihaka

INTRODUÇÃO AO R

R Studio:

- Ambiente de desenvolvimento integrado
- Interface gráfica mais simples e intuitiva
- Vantagens em relação ao R:
 - Highlight do código
 - Autocomplete
 - Match automático de parênteses e chaves
 - Interface intuitiva para objetos, gráficos e scripts
 - Interação com HTML e outras linguagens.

INTERFACE R



INTERFACE RSTUDIO

The image shows the RStudio interface with four main panels highlighted by blue borders and containing text annotations:

- Editor:** The top-left panel, labeled "Editor" in blue. It contains the text "Aqui, escrevemos o código" (Here, we write the code) and a single line of code "1 |".
- Environment:** The top-right panel, labeled "Ambiente" in blue. It shows "Global Environment" and "Environment is empty". It contains the text "Mostra os objetos criados na sessão" (Shows the objects created in the session).
- Console:** The bottom-left panel, labeled "Console" in blue. It shows a terminal prompt "> |". It contains the text "Aqui rodamos o código (o R 'vive' aqui!)" (Here we run the code (R 'lives' here!)).
- Painel Iterativo:** The bottom-right panel, labeled "Painel Iterativo" in blue. It shows a file browser view with columns "Name", "Size", and "Modified". It contains the text "Mostra:" followed by a list of items: "Files = arquivos do diretório de trabalho;" (Files = files in the working directory); "Plots = gráficos;" (Plots = graphics); "Packages = pacotes instalados" (Packages = installed packages); "Help = documentação das funções;" (Help = documentation of functions); "History: histórico dos comandos rodados;" (History: history of commands run).

SINTAXE E TIPOS DE DADOS DO R

Sintaxe: é como se escreve os comandos (códigos) interpretados pelo programa. Ex:

- Distinção entre maiúsculas e minúsculas: “A” é diferente de “a”;
- Distinção entre “[”, “(” e “{”;
- “.” separador decimal;
- “,” separador de objetos;
- “;” separador de comandos

Tipos de dados:

- Numéricos (numeric);
- Caracteres (character);
- Lógicos ou booleanos (logical);
- Complexos (complex).

PRINCIPAIS OPERADORES

Relacionais	
Símbolo	Descrição
<- ou =	Atribuição
<	Menor
<=	Menor igual
>	Maior
>=	Maior igual
==	Igual (comparação)
!=	Diferente
&	AND
	OR
!	NOT
TRUE ou 1	Valor booleano verdadeiro (1)
FALSE ou 0	Valor booleano falso (0)

Matemáticos	
Símbolo	Descrição
+	Adição
-	Subtração
*	Multiplicação
/	Divisão
^	Potenciação
%%	Resto da divisão

Ps.: o símbolo “#” insere um comentário no código.

PRINCIPAIS OPERADORES

Exemplos:

```
> a = 2
```

```
> b = 3
```

```
> c = 4
```

```
> d = a + b + c
```

Ou

```
> a = 2 ; b = 3 ; c = 4 ; d = a + b + c
```

Para saber o tipo (classe) do objeto:
Comando "class()" ou "mode()".

Exemplo:

```
x = 2
```

```
y = "olá"
```

```
z = 2>1
```

```
w = 1 + 2i
```

```
class(x)
```

```
class(y)
```

```
class(z)
```

```
class(w)
```


FUNÇÕES MATEMÁTICAS SIMPLES

O R possui várias funções matemáticas já implementadas. A sintaxe das funções do R é relativamente simples:

***nome_da_função* (argumento)**

Função	Descrição
<i>abs(x)</i>	valor absoluto de x
<i>log(x, b)</i>	logaritmo de x com base b
<i>log(x)</i>	logaritmo natural de x
<i>log10(x)</i>	logaritmo de x com base 10
<i>exp(x)</i>	exponencial elevado a x
<i>sin(x)</i>	seno de x
<i>cos(x)</i>	cosseno de x
<i>tan(x)</i>	tangente de x

FUNÇÕES MATEMÁTICAS SIMPLES

Continuação...

Função	Descrição
<code>round(x, digits = n)</code>	arredonda x com n decimais
<code>ceiling(x)</code>	arredondamento de x para o maior valor
<code>floor(x)</code>	arredondamento de x para o menor valor
<code>length(x)</code>	número de elementos do vetor x
<code>sum(x)</code>	soma dos elementos do vetor x
<code>prod(x)</code>	produto dos elementos do vetor x
<code>max(x)</code>	seleciona o maior elemento do vetor x
<code>min(x)</code>	seleciona o menor elemento do vetor x

Exemplos:

```
> x = 30; y = 60
> (sin(x))^2 + (cos(x))^2
[1] 1
> round(tan(2*y), digits = 3)
[1] 0.713
> floor(tan(2*y))
[1] 0
> ceiling(tan(2*y))
[1] 1
```

FUNÇÕES MATEMÁTICAS SIMPLES

Continuação...

Função	Descrição
<code>round(x, digits = n)</code>	arredonda x com n decimais
<code>ceiling(x)</code>	arredondamento de x para o maior valor
<code>floor(x)</code>	arredondamento de x para o menor valor
<code>length(x)</code>	número de elementos do vetor x
<code>sum(x)</code>	soma dos elementos do vetor x
<code>prod(x)</code>	produto dos elementos do vetor x
<code>max(x)</code>	seleciona o maior elemento do vetor x
<code>min(x)</code>	seleciona o menor elemento do vetor x

Exemplos:

```
> x = 30; y = 60
> (sin(x))^2 + (cos(x))^2
[1] 1
> round(tan(2*y), digits = 3)
[1] 0.713
> floor(tan(2*y))
[1] 0
> ceiling(tan(2*y))
[1] 1
```

NÚMEROS COMPLEXOS

Para utilizar números complexos, o R possui a variável especial “i”, que compõe a parte imaginária do número complexo. Mesmo assim, é necessário evidenciar a parte imaginária, mesmo quando ela seja nula, caso contrário o R retorna valor NaN (Not a Number), que indica uma indefinição matemática.

```
> sqrt(-17)                                     #raiz quadrada de número negativo
[1] NaN                                         #indefinição matemática
Warning message:
In sqrt(-17) : NaNs produced
> sqrt(-17+0i)                                  #raiz quadrada de um número complexo
[1] 0+4.123106i
```

VALORES ESPECIAIS

Existem valores reservados para representar dados faltantes, infinitos e indefinições matemáticas (valores impossíveis).

NULL (Nulo):

retornado há ausência de informação.

NULL= associado à análises lógicas

NaN (Not a Number):

retornado quando há uma indefinição matemática.

Exemplo:

```
> 0 / 0
```

```
[1] NaN
```

inf (infinito):

retornado quando um número é muito grande.

Exemplo:

```
> x = 10^310
```

```
[1] inf
```

NA (Not Available):

retornado quando há um valor faltante e pode ser de diferentes classes.

Exemplo:

```
> x = c(20,30,40,50)
```

```
> x[5]
```

```
[1] NA
```

ESTRUTURA DE DADOS

Tipos básicos de estrutura de dados (tipo de objetos) no R:

Atomic Vector:

- Homogêneo e unidimensional.
- Sequencia de elementos do mesmo tipo.

```
>vec1 = c(3, 6, 7, 8 ,9)
```

Matriz:

- Homogêneo e bidimensional.
- Conjuntos de elementos com linhas e colunas do mesmo tipo.

```
>mat1 = matrix(c(1,2,10,50,60,70),  
              nrow=3, ncol=2)
```

```
> print(mat1)
```

```
  [,1] [,2]
```

```
[1,]  1  50
```

```
[2,]  2  60
```

```
[3,]  0  70
```

ESTRUTURA DE DADOS

Tipos básicos de estrutura de dados (tipo de objetos) no R:

Array:

- Homogêneo e multidimensional.
- Estrutura semelhante a das matrizes, porém podem ser multidimensionais.

```
> arr1 = array (1:12, c(3,2,2))
```

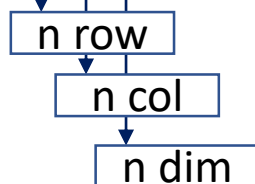
```
> print(arr1)
```

```
, , 1
```

```
  [,1] [,2]
[1,]  1  4
[2,]  2  5
[3,]  3  6
```

```
, , 2
```

```
  [,1] [,2]
[1,]  7 10
[2,]  8 11
[3,]  9 12
```



Lista:

- Heterogêneo
- Vetor especial que armazena um conjunto de dados heterogêneos.

```
> a <- c(3,6,9)
```

```
> b <- c("a","b","c","d")
```

```
> c <- c(TRUE, FALSE, TRUE, TRUE)
```

```
> lista1 <- list(a,b,c)
```

```
> print (lista1)
```

```
[[1]]
```

```
[1] 3 6 9
```

```
[[2]]
```

```
[1] "a" "b" "c" "d"
```

```
[[3]]
```

```
[1] TRUE FALSE TRUE TRUE
```

ESTRUTURA DE DADOS

Tipos básicos de estrutura de dados (tipo de objetos) no R:

Data Frame

- Heterogêneo.
- Listas em que todos os elementos tem o mesmo comprimento.
- Armazena os dados em tabelas, organizando em linhas e colunas.

```
> df1 = data.frame(x = 1:4, y = c("s","n","z","w"), z=T)
```

```
> print(df1)
```

```
  x y  z
1 1 s TRUE
2 2 n TRUE
3 3 z TRUE
4 4 w TRUE
```

Para saber qual é o tipo de um objeto, utilizar a função *typeof()*

GRÁFICOS

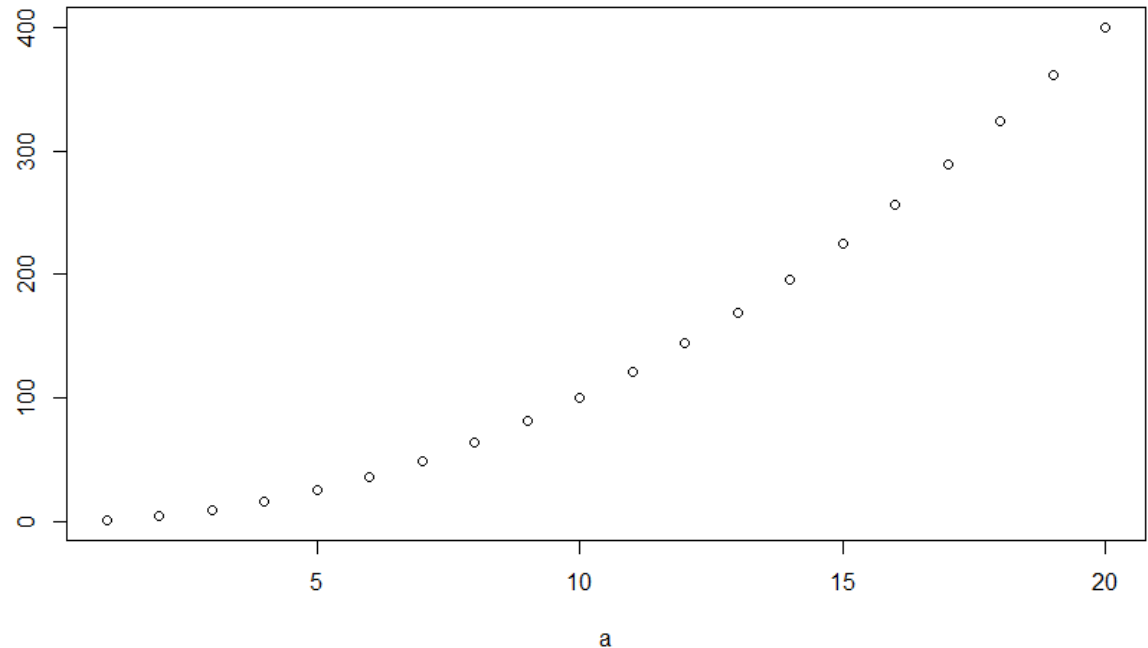
Plotagem de gráficos simples e complexos por meio de comandos simples:

Comandos básicos

- `plot()`
- Gera um gráfico simples, atribuindo pontos e coordenadas cartesianas.

Exemplo:

```
> a = 1:20  
> b = a^2  
> plot(a,b)
```

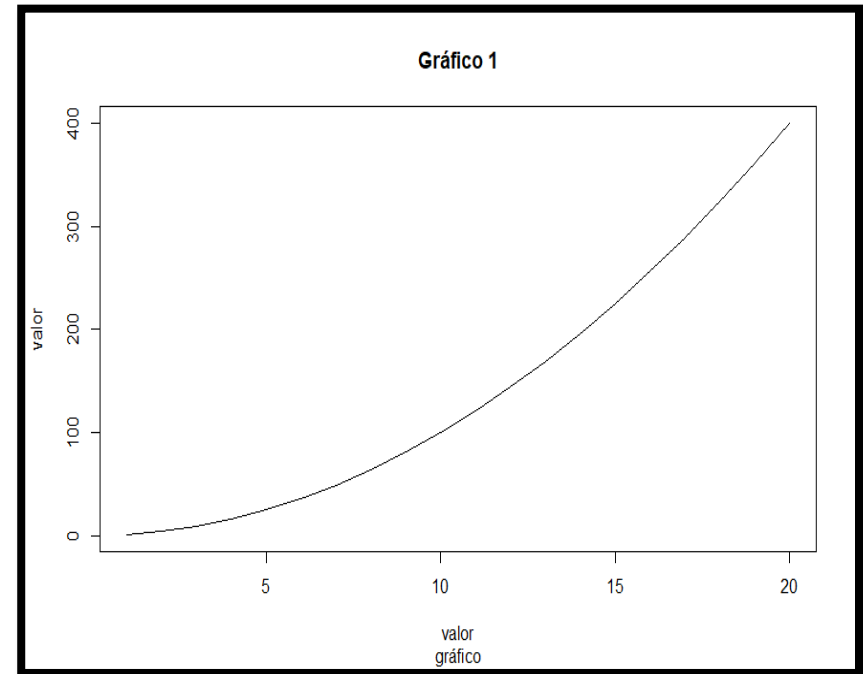


GRÁFICOS

Comandos básicos plot(x, y, ...):

Argumentos básicos:

- type: tipo de gráfico será plotado:
 - “p” = pontos
 - “l” = linhas
 - “b” = linhas e pontos
 - “o” = linhas e pontos sobrepostos
 - “h” = linhas verticais (“histogram”)
- main = título principal do gráfico;
- sub = subtítulo do gráfico;
- xlab = título do eixo x;
- ylab = título do eixo y;



Exemplo:

```
> a = 1:20
```

```
> b = a^2
```

```
> plot(a,b, type = "l", main = "Gráfico 1", sub = "gráfico", xlab = "valor", ylab = "valor")
```

GRÁFICOS

Histograma: realiza contagem

Sintaxe:

```
hist(dados, nclass=k, ...)
```

#k é o número de classes

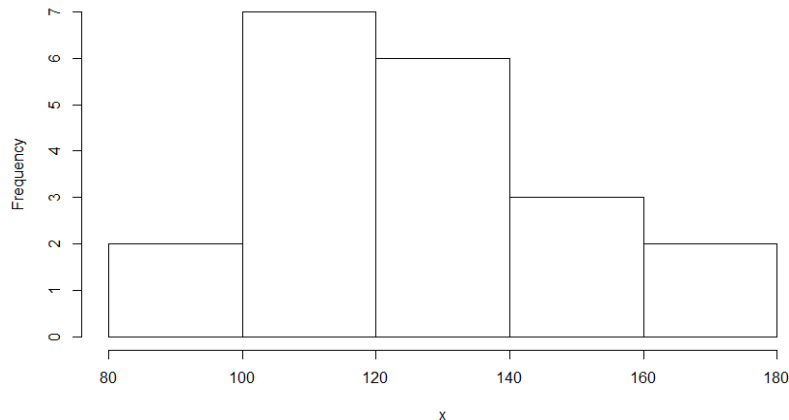
Exemplo:

x=

```
c(96,96,102,102,102,104,104,108,126,128,128,140,156,160,160,164,170,115,121)
```

```
hist(x, nclass = 5)
```

Histogram of x



Barplot: gráfico de barra

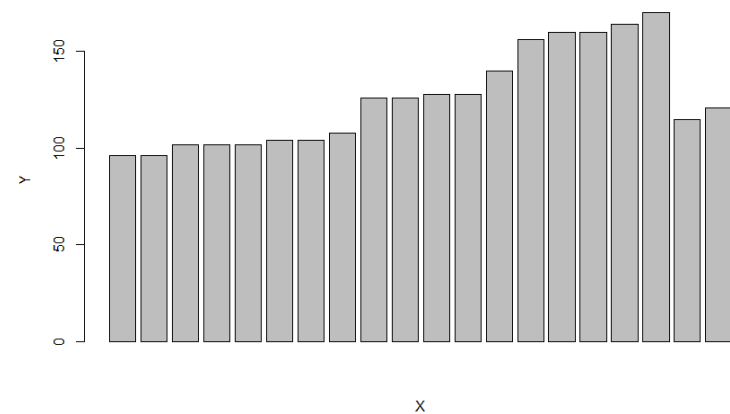
Sintaxe:

```
barplot(x, ...)
```

Exemplo:

```
barplot(x, main = "BAR PLOT de X", xlab = "X", ylab = "Y")
```

BAR PLOT de X



GRÁFICOS

Boxplot: representa a distribuição de um conjunto de dados com base em alguns parâmetros descritivos (mediana e quartis).

Sintaxe:

```
boxplot(x, ...)
```

Exemplo:

```
boxplot(x, main = "Boxplot de X",  
xlab = "X")
```

Boxplot de X

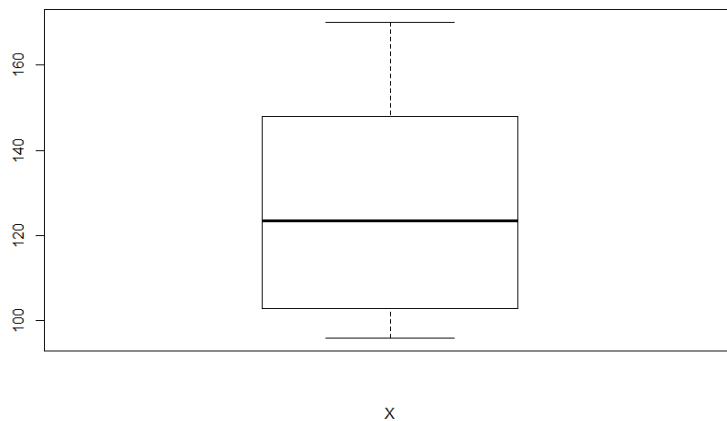


Gráfico de Pizza: proporção de classes

Sintaxe:

```
pie(dados, ...)
```

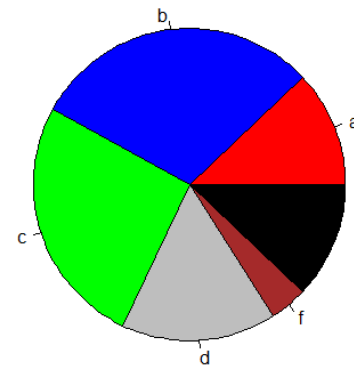
Exemplo:

```
a = c(0.12,0.3,0.26,0.16,0.04,0.12)
```

```
names(a) = c("a", "b", "c", "d", "f")
```

```
pie(a, col = c("red", "blue", "green", "gray",  
"brown", "black"), main = "Proporção de  
classes")
```

Proporção de classes



PACOTES

As funções básicas de estatística já vêm programadas no R. Entretanto, funções mais avançadas requerem o uso de pacotes que executem essas funções.

Exemplo: Para estimar regressões via *Ordinary Least Squares* (OLS) é necessário a utilização de um pacote específico.

Para instalar um pacote no R (Rstudio):

Packages -> Install-> digitar o nome do pacote -> Install

Para carregar um pacote no R (Rstudio):

Packages -> selecionar o pacote e marcar a caixinha

HANDS ON...

Análise de Regressão com o R.

- 1. Baixar e descompactar os arquivos que estão na página do curso;**
- 2. Abrir o script no Rstudio**