



Linguagem R

Aula 5

- Objetivo da aula:
- Introdução ao RStudio;
- Carregando um DataFrame;
- Tipos de dados e operadores;
- Estruturas de dados;
- Funções.

Introdução a Linguagem R:

R é uma linguagem e ambiente para computação estatística e construção de gráficos.

R fornece uma ampla variedade de técnicas estatísticas e gráficas: modelagem linear e não linear, testes estatísticos, análise de séries temporais, classificação, agrupamento, etc.

Instalação:

Antes da instalação do RStudio é necessário instalar o interpretador R.

O download do interpretador R e do RStudio estão na mesma página.

[PRODUCTS](#)[SOLUTIONS](#)[LEARN & SUPPORT](#)[EXPLORE MORE](#)[DOWNLOAD RSTUDIO](#)

Download RStudio Desktop

Used by millions of people weekly, the RStudio integrated development environment (IDE) is a set of tools built to help you be more productive with R and Python. It includes a console, syntax-highlighting editor that supports direct code execution. It also features tools for plotting, viewing history, debugging and managing your workspace.

Select Your Operating System:

[All Installers](#)

Step 1: Install R

RStudio requires R 3.3.0+. Choose a version of R that matches your computer's operating system.

[DOWNLOAD AND INSTALL R](#)

Step 2: Install RStudio Desktop

[DOWNLOAD RSTUDIO DESKTOP FOR WINDOWS](#)

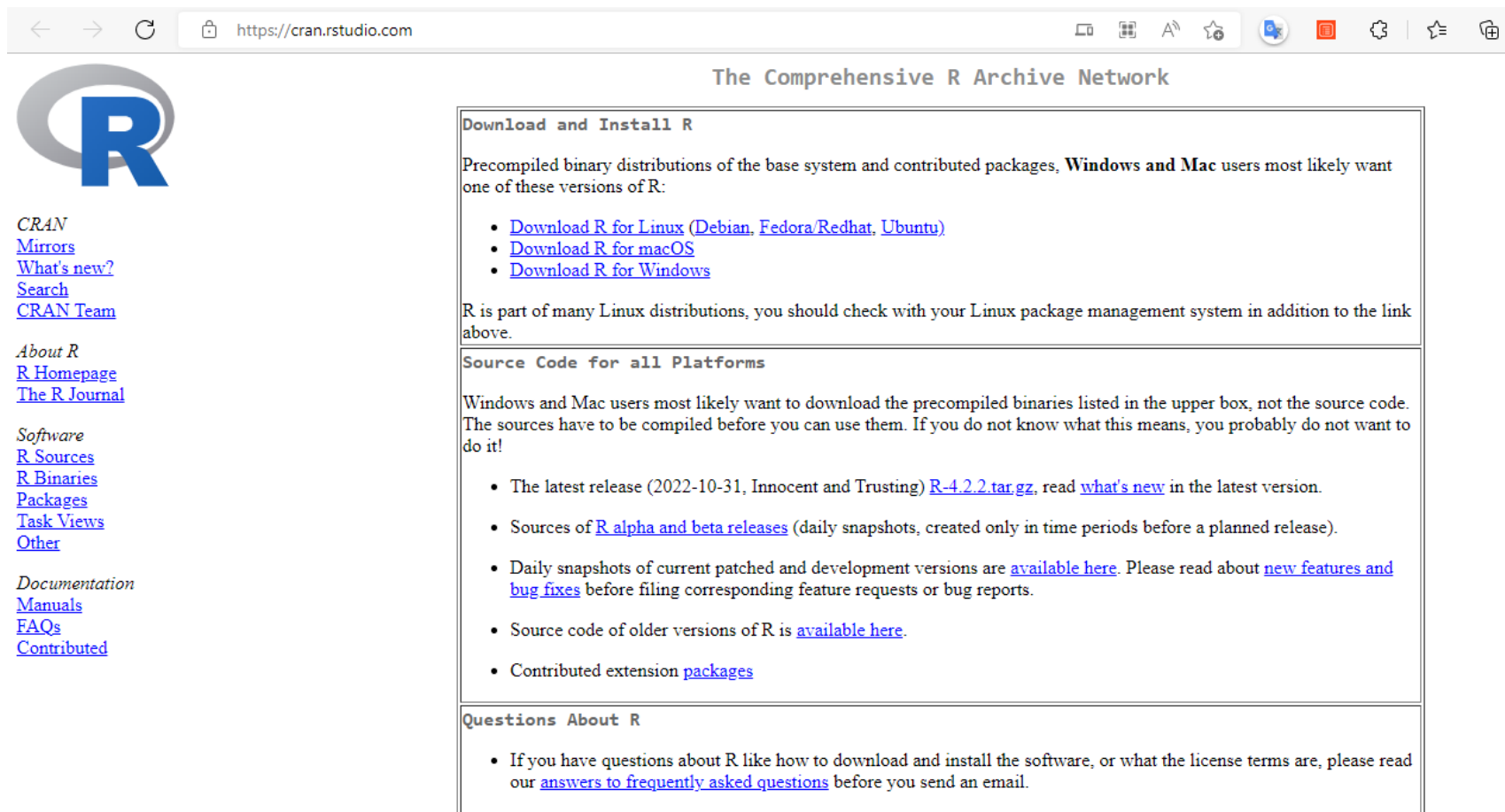
Size: 190.49MB | SHA-256: 83886925 | Version: 2022.07.2+576 |
Released: 2022-09-21

Instalação do interpretador R:

Para a instalação completa do software (RStudio-IDE) é necessário a instalação do R.

Download R em: <https://cran.rstudio.com> de acordo com o sistema operacional do seu desktop.

Download do interpretador R:



The screenshot shows the CRAN website in a web browser. The browser's address bar displays <https://cran.rstudio.com>. The website header reads "The Comprehensive R Archive Network". On the left side, there is a navigation menu with links: [CRAN](#), [Mirrors](#), [What's new?](#), [Search](#), [CRAN Team](#), [About R](#), [R Homepage](#), [The R Journal](#), [Software](#), [R Sources](#), [R Binaries](#), [Packages](#), [Task Views](#), [Other](#), [Documentation](#), [Manuals](#), [FAQs](#), and [Contributed](#). The main content area is titled "Download and Install R" and contains the following text: "Precompiled binary distributions of the base system and contributed packages, **Windows and Mac** users most likely want one of these versions of R:" followed by a bulleted list of links: [Download R for Linux \(Debian, Fedora/Redhat, Ubuntu\)](#), [Download R for macOS](#), and [Download R for Windows](#). Below this, it states: "R is part of many Linux distributions, you should check with your Linux package management system in addition to the link above." The next section is "Source Code for all Platforms", which explains that Windows and Mac users should download precompiled binaries instead of source code. It provides a bulleted list of links for the latest release ([R-4.2.2.tar.gz](#)), alpha and beta releases, daily snapshots, and older versions. The final section is "Questions About R", which directs users to [answers to frequently asked questions](#) if they have queries about installation or licensing.

■

The Comprehensive R Archive Network

Download and Install R

Precompiled binary distributions of the base system and contributed packages, **Windows and Mac** users most likely want one of these versions of R:

- [Download R for Linux \(Debian, Fedora/Redhat, Ubuntu\)](#)
- [Download R for macOS](#)
- [Download R for Windows](#)

R is part of many Linux distributions, you should check with your Linux package management system in addition to the link above.

Source Code for all Platforms

Windows and Mac users most likely want to download the precompiled binaries listed in the upper box, not the source code. The sources have to be compiled before you can use them. If you do not know what this means, you probably do not want to do it!

- The latest release (2022-10-31, Innocent and Trusting) [R-4.2.2.tar.gz](#), read [what's new](#) in the latest version.
- Sources of [R alpha and beta releases](#) (daily snapshots, created only in time periods before a planned release).
- Daily snapshots of current patched and development versions are [available here](#). Please read about [new features and bug fixes](#) before filing corresponding feature requests or bug reports.
- Source code of older versions of R is [available here](#).
- Contributed extension [packages](#)

Questions About R

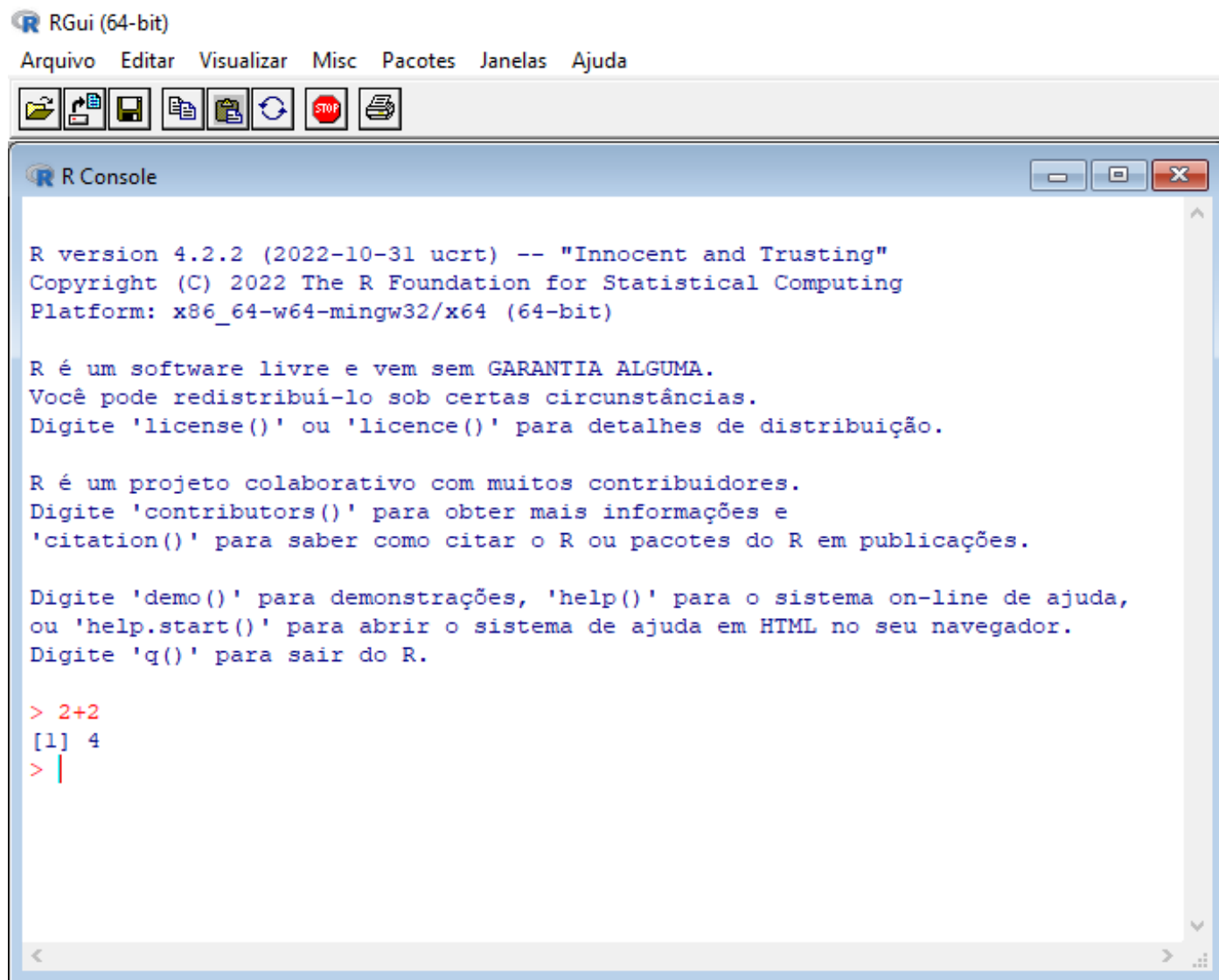
- If you have questions about R like how to download and install the software, or what the license terms are, please read our [answers to frequently asked questions](#) before you send an email.

Testando a Instalação do R:

Após a instalação:

Abra o console do R e faça uma simples checagem digitando $2+2$ no console.

Se a instalação estiver sido feita da maneira correta então o resultado será algo como mostrado a seguir.



RGui (64-bit)

Arquivo Editar Visualizar Misc Pacotes Janelas Ajuda

R Console

```
R version 4.2.2 (2022-10-31 ucrt) -- "Innocent and Trusting"
Copyright (C) 2022 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R é um software livre e vem sem GARANTIA ALGUMA.
Você pode redistribuí-lo sob certas circunstâncias.
Digite 'license()' ou 'licence()' para detalhes de distribuição.

R é um projeto colaborativo com muitos contribuidores.
Digite 'contributors()' para obter mais informações e
'citation()' para saber como citar o R ou pacotes do R em publicações.

Digite 'demo()' para demonstrações, 'help()' para o sistema on-line de ajuda,
ou 'help.start()' para abrir o sistema de ajuda em HTML no seu navegador.
Digite 'q()' para sair do R.

> 2+2
[1] 4
> |
```

Rstudio IDE:

RStudio é um ambiente de desenvolvimento integrado (IDE) para R.

RStudio inclui um console, um editor com destaque para sintaxe que suporta execução direta de código e ferramentas para plotagem, depuração e gerenciamento de espaço de trabalho.

Download RStudio em: <https://posit.co/download/rstudio-desktop/> de acordo com o sistema operacional do seu desktop.



PRODUCTS

SOLUTIONS

LEARN & SUPPORT

EXPLORE MORE



DOWNLOAD RSTUDIO

Download RStudio Desktop

Used by millions of people weekly, the RStudio Integrated development environment (IDE) is a set of tools built to help you be more productive with R and Python. It includes a console, syntax-highlighting editor that supports direct code execution. It also features tools for plotting, viewing history, debugging and managing your workspace.

Select Your Operating System:

[All installers](#)

Step 1: Install R

RStudio requires R 3.3.0+. Choose a version of R that matches your computer's operating system.

[DOWNLOAD AND INSTALL R](#)

Step 2: Install RStudio Desktop

[DOWNLOAD RSTUDIO DESKTOP FOR WINDOWS](#)

Size: 190.49MB | SHA-256: 8388f925 | Version: 2022.07.2+576 |
Released: 2022-09-21

RStudio:

A IDE do RStudio deve estar configurada da maneira mostrada a seguir.

Execute o comando `2+2` no RStudio como mostrado a seguir.

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function Addins

Project: (None)

Environment History Connections Tutorial

Import Dataset 97 MiB

R Global Environment

Environment is empty

Files Plots Packages Help Viewer Presentation

Zoom Export

1:1 (Top Level) R Script

Console Terminal Background Jobs

R 4.2.2 ~/
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> 2+2
[1] 4
>

Insertir o código.

Rodar o código.

Execução do código.

Começando com o RStudio:

Abra uma pasta em C:

Criando um arquivo com extensão .R


Abra a IDE do RStudio, acesse FILE, depois NEWFILE e por fim em RSCRIPT.

Crie um arquivo chamado vendas.R

Abra o Editor de textos e na pasta salve um arquivo com extensão .csv chamado vendas.csv

Carregando o DataFrame:

No bloco de notas no arquivo vendas.csv digite e salve-o os dados abaixo:

 *vendas - Bloco de Notas

Arquivo Editar Formatar Exibir Ajuda

```
ID,Data,Produto,Valor,Sede
14032022,14/3/2022,ford,50,Botafogo
15032022,15/3/2022,bmw,120,Maracanã
20032022,20/3/2022,nissan,90,Barra
220302022,22/03/2022,jeep,80,Ipanema
28032022,28/03/2022,fiat,60,Jacarepaguá
```

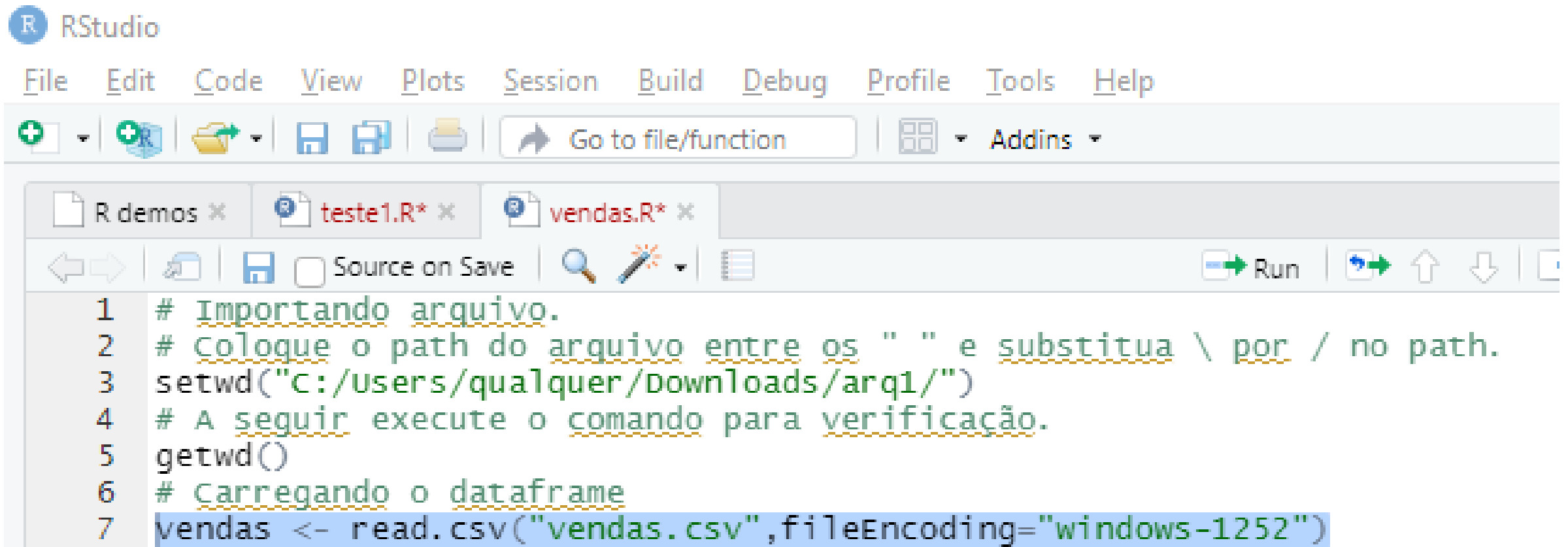
Começando com o Rstudio:

Fazendo a importação do arquivo.

Para isto usa-se a função `setwd("path do arquivo")`
Atenção ao path pois usa-se / por exemplo: `C:/usuário/aluno`

Depois usa-se a função `getwd()` para checar o procedimento anterior.

Importando o arquivo .csv:



The screenshot shows the RStudio application window. The title bar reads 'RStudio'. The menu bar includes 'File', 'Edit', 'Code', 'View', 'Plots', 'Session', 'Build', 'Debug', 'Profile', 'Tools', and 'Help'. The toolbar contains icons for creating a new file, opening a file, saving, printing, and a search bar labeled 'Go to file/function'. Below the toolbar, the file explorer shows three open files: 'R demos', 'teste1.R*', and 'vendas.R*'. The 'vendas.R*' file is active, and its code is displayed in the editor. The code consists of seven lines, with the last line highlighted in blue. The code is as follows:

```
1 # Importando arquivo.
2 # Coloque o path do arquivo entre os " " e substitua \ por / no path.
3 setwd("C:/Users/qualquer/Downloads/arq1/")
4 # A seguir execute o comando para verificação.
5 getwd()
6 # Carregando o dataframe
7 vendas <- read.csv("vendas.csv",fileEncoding="windows-1252")
```

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function Addins

R demos x teste1.R x vendas.R x

Source on Save Run Source

```
1 # Importando arquivo.
2 # Coloque o path do arquivo entre os " " e substitua \ por / no path.
3 setwd("C:/Users/qualquer/Downloads/arq1/")
4 # A seguir execute o comando para verificação.
5 getwd()
6 # Carregando o dataframe
7 vendas <- read.csv("vendas.csv",fileEncoding="windows-1252")
8 # Sumário do dataframe
9 view(vendas)
10 str(vendas)
11 summary(vendas$valor)
12 summary(vendas$sede)
13
```

7:1 (Top Level) R Script

Use esse encoding, pois pode haver palavras com acento.

Console Terminal x Background Jobs x

R 4.2.2 · C:/Users/qualquer/Downloads/arq1/

```
> vendas <- read.csv("vendas.csv",fileEncoding="windows-1252")
> |
```

Observe que o software carregou o dataframe com sucesso.

Environment History Connections Tutorial

Import Dataset 98 MiB

R Global Environment

Data

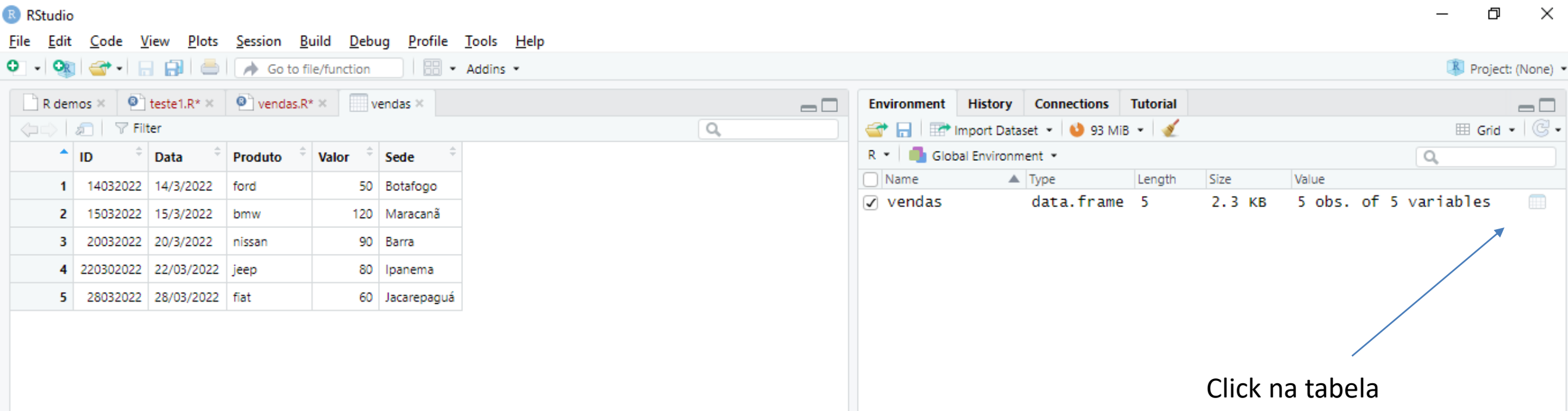
vendas 5 obs. of 5 variables

Observe o dataframe carregado.

Files Plots Packages Help Viewer Presentation

Zoom Export

Vista do DataFrame :



The screenshot shows the RStudio interface. The Editor pane on the left displays a preview of a DataFrame with 5 rows and 6 columns: ID, Data, Produto, Valor, and Sede. The Environment pane on the right shows the 'Global Environment' with a table for the 'vendas' object, which is a data.frame with 5 rows and 5 variables. A blue arrow points from the text below to the 'vendas' table in the Environment pane.

	ID	Data	Produto	Valor	Sede
1	14032022	14/3/2022	ford	50	Botafogo
2	15032022	15/3/2022	bmw	120	Maracanã
3	20032022	20/3/2022	nissan	90	Barra
4	220302022	22/03/2022	jeep	80	Ipanema
5	28032022	28/03/2022	fiat	60	Jacarepaguá

Environment | History | Connections | Tutorial

R | Global Environment

<input type="checkbox"/>	Name	Type	Length	Size	Value
<input checked="" type="checkbox"/>	vendas	data.frame	5	2.3 KB	5 obs. of 5 variables

Click na tabela
para que esta
apareça.

Informações e Sumário do DataFrame:

Use `str(nome do arquivo)` para obter as informações.

Use `summary(nome do arquivo)` para obter um sumário dos dados.

Use `summary(nome do arquivo$nome da variável)` para obter um sumário dos dados de uma coluna.

str:

```
8 #Sumário do dataframe
9 str(vendas)
10 summary(vendas$valor)
11 summary(vendas$Sede)
12
```

10:1 (Top Level) ↕

R Script ↕

Console

Terminal ×

Background Jobs ×

R 4.2.2 · C:/Users/qualquer/Downloads/arq1/ ↗

> view(vendas)

Error in view(vendas) : could not find function "view"

> #Sumário do dataframe

> str(vendas)

'data.frame': 5 obs. of 5 variables:

\$ ID : int 14032022 15032022 20032022 220302022 28032022

\$ Data : chr "14/3/2022" "15/3/2022" "20/3/2022" "22/03/2022" ...

\$ Produto: chr "ford" "bmw" "nissan" "jeep" ...

\$ valor : int 50 120 90 80 60

\$ Sede : chr "Botafogo" "Maracanã" "Barra" "Ipanema" ...

> |

Summary:

```
8 #Sumário do dataframe
9 str(vendas)
10 summary(vendas)
11 summary(vendas$valor)
12 summary(vendas$Sede)
13
```

10:1

(Top Level) ↕

R Script ↕

Console

Terminal ×

Background Jobs ×

R 4.2.2 · C:/Users/qualquer/Downloads/arq1/ ↗

Sede : chr Botafogo Maracana Barra Ipanema ...

> summary(vendas)

ID	Data	Produto	valor
Min. : 14032022	Length:5	Length:5	Min. : 50
1st Qu.: 15032022	Class :character	Class :character	1st Qu.: 60
Median : 20032022	Mode :character	Mode :character	Median : 80
Mean : 59486022			Mean : 80
3rd Qu.: 28032022			3rd Qu.: 90
Max. :220302022			Max. :120
Sede			
Length:5			
Class :character			
Mode :character			

Summary:

```
8 #Sumário do dataframe
9 str(vendas)
10 summary(vendas)
11 summary(vendas$valor)
12 summary(vendas$Sede)
13
```

11:1 (Top Level) ↕

R Script ↕

Console

Terminal ×

Background Jobs ×

R 4.2.2 · C:/Users/qualquer/Downloads/arq1/ ↗

Min. : 14032022	Length:5	Length:5	Min. : 50
1st Qu.: 15032022	Class :character	Class :character	1st Qu.: 60
Median : 20032022	Mode :character	Mode :character	Median : 80
Mean : 59486022			Mean : 80
3rd Qu.: 28032022			3rd Qu.: 90
Max. :220302022			Max. :120

Sede
Length:5
Class :character
Mode :character

```
> summary(vendas$valor)
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   50     60     80     80     90    120

> |
```

Tipos de Dados em R:

- numeric - (10.5, 55, 787.333)
- integer - (55L, 100L, onde "L" declara que o valor é inteiro)
- complex - ($9 + 3i$, onde "i" é o imaginário da função)
- character (string) - ("k", "R é show", "FALSE", "11.5")
- logical (boolean) - (TRUE ou FALSE)

Tipos de dados e operadores:

O R tem as seguintes operações básicas:

sinal	operação
+	Adição
-	Subtração
/	Divisão
*	Multiplicação
^	Potência
sqrt	Raiz quadrada

Variáveis:

Para guardar os valores em uma variável em R, usamos `<-`.

Para imprimir a variável usa-se o `print`.

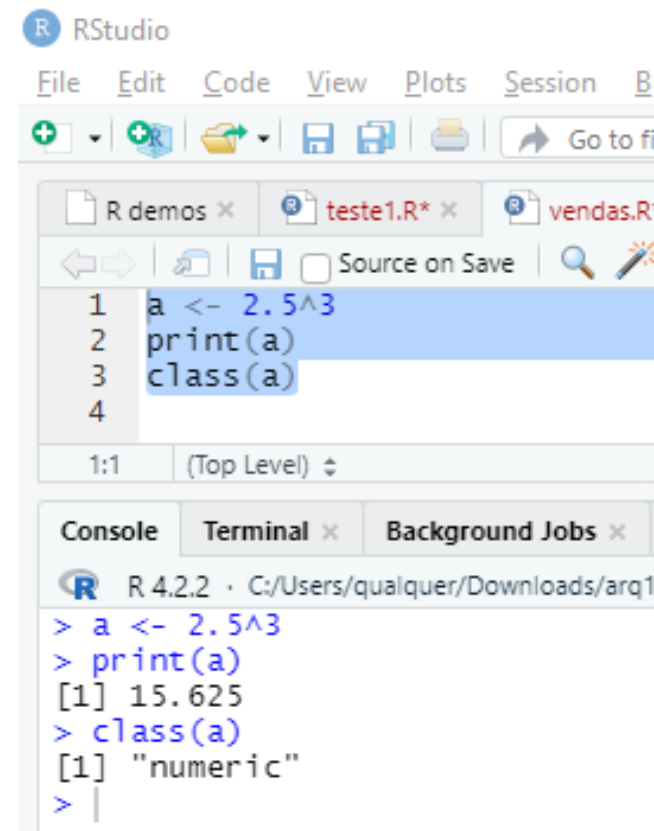
`Class` vai imprimir o tipo de variável.

Exemplo:

```
a <- 2.5^3
```

```
print(a)
```

```
class(a)
```



The screenshot shows the RStudio environment. The script editor on the left contains the following code:

```
1 a <- 2.5^3
2 print(a)
3 class(a)
4
```

The console on the right shows the output of the executed code:

```
R 4.2.2 · C:/Users/qualquer/Downloads/arq1
> a <- 2.5^3
> print(a)
[1] 15.625
> class(a)
[1] "numeric"
>
```

Variáveis:

Se quiser trabalhar com número inteiro, deve-se definí-los como integer, através da função `as.integer()` ou `L` no final.

Por Exemplo:

```
a <- 3
```

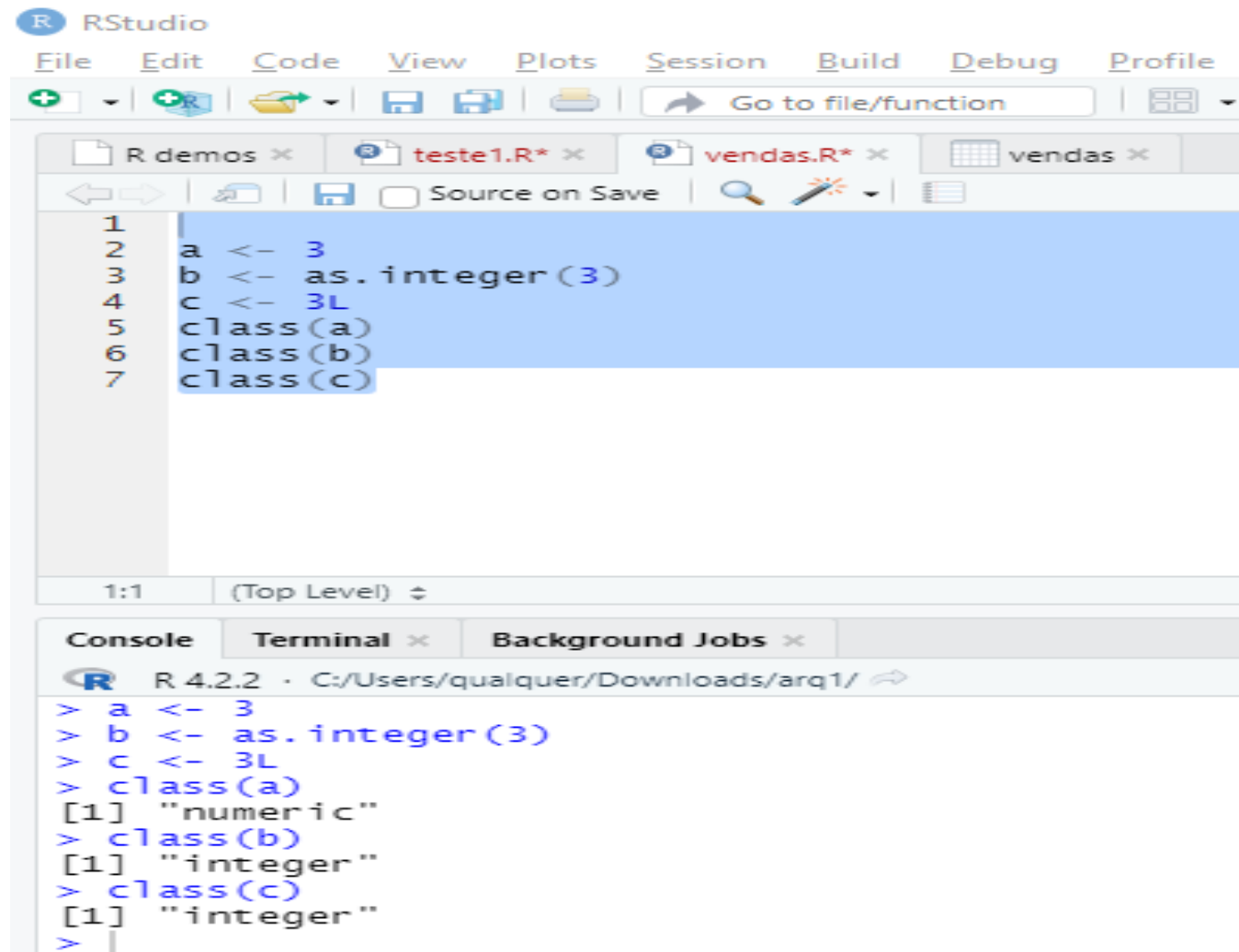
```
b <- as.integer(3)
```

```
c <- 3L
```

```
class(a)
```

```
class(b)
```

```
class(c)
```



The screenshot displays the RStudio environment. The top menu bar includes File, Edit, Code, View, Plots, Session, Build, Debug, and Profile. Below the menu is a toolbar with icons for file operations and a search bar labeled 'Go to file/function'. The editor pane shows a script with the following code:

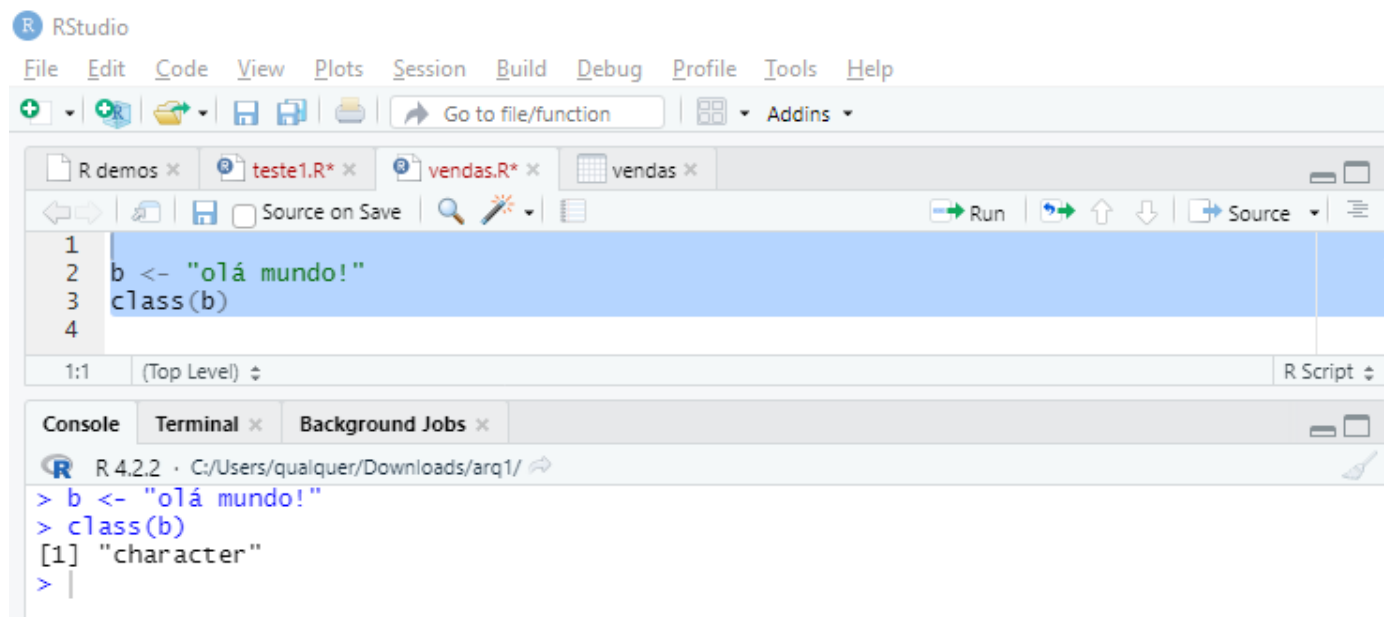
```
1  
2 a <- 3  
3 b <- as.integer(3)  
4 c <- 3L  
5 class(a)  
6 class(b)  
7 class(c)
```

The console pane at the bottom shows the output of the code execution:

```
> a <- 3  
> b <- as.integer(3)  
> c <- 3L  
> class(a)  
[1] "numeric"  
> class(b)  
[1] "integer"  
> class(c)  
[1] "integer"  
>
```

Variável (Character):

Em R, os dados também podem ser do tipo Character.



The screenshot shows the RStudio interface. The script editor contains the following code:

```
1  
2 b <- "olá mundo!"  
3 class(b)  
4
```

The console output shows the execution of the code:

```
R 4.2.2 · C:/Users/qualquer/Downloads/arq1/  
> b <- "olá mundo!"  
> class(b)  
[1] "character"  
>
```

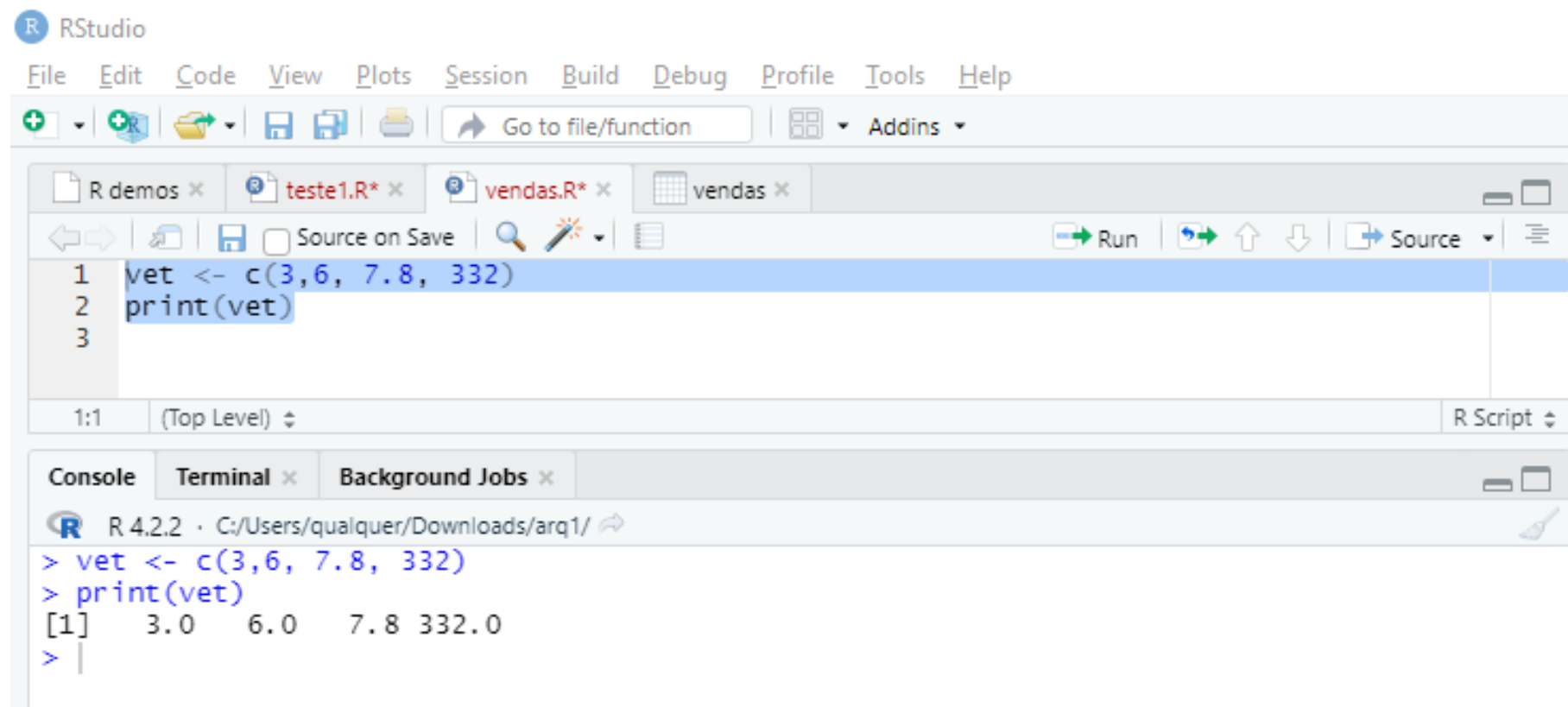
Estruturas de dados no R:

VETOR:

Um vetor é um conjunto de dados do mesmo tipo.

Cria-se vetores com mais de um elemento dentro de um `c()`.

Exemplo: `vet <- c(2,3,4)`



The screenshot shows the RStudio application window. At the top is the menu bar with options: File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, and Help. Below the menu bar is a toolbar with icons for creating a new file, opening a file, saving, printing, and navigating to a file/function. The main editor area has a tab bar with four tabs: 'R demos', 'teste1.R*', 'vendas.R*', and 'vendas'. The 'vendas.R*' tab is active, showing a script with three lines of code: `1 vet <- c(3,6, 7.8, 332)`, `2 print(vet)`, and `3`. The first two lines are highlighted in blue. Below the editor is a toolbar with icons for undo, redo, save, source on save, search, and run. The 'Run' button is highlighted. Below the editor is a status bar showing '1:1' and '(Top Level)'. At the bottom is a console area with tabs for 'Console', 'Terminal', and 'Background Jobs'. The 'Console' tab is active, showing the output of the script: `> vet <- c(3,6, 7.8, 332)`, `> print(vet)`, and `[1] 3.0 6.0 7.8 332.0`. The console also shows the R version 'R 4.2.2' and the current directory 'C:/Users/qualquer/Downloads/arq1/'.

```
RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
+ + + + + Go to file/function + Addins
R demos x teste1.R* x vendas.R* x vendas x
< > | | | Source on Save | | | Run | | | Source
1 vet <- c(3,6, 7.8, 332)
2 print(vet)
3
1:1 (Top Level) R Script
Console Terminal x Background Jobs x
R 4.2.2 · C:/Users/qualquer/Downloads/arq1/
> vet <- c(3,6, 7.8, 332)
> print(vet)
[1] 3.0 6.0 7.8 332.0
> |
```

Estruturas de dados no R:

MATRIZ:

É um conjunto bidimensional de dados.

Constrói-se uma matriz fornecendo os dados e o número de linhas e colunas.

Observe no exemplo a seguir a diferença da matriz 3x2 da matriz 2x3.


```
1 mat_32 <- matrix(  
2   c(1,5,10,30,15,8),  
3   nrow=3,  
4   ncol=2,  
5   byrow=TRUE)  
6 print(mat_32)  
7  
8 mat_23 <- matrix(  
9   c(1,5,10,30,15,8),  
10  nrow=2,  
11  ncol=3,  
12  byrow=TRUE)  
13 print(mat_23)
```

```
> mat_32 <- matrix(
+   c(1,5,10,30,15,8),
+   nrow=3,
+   ncol=2,
+   byrow=TRUE)
> print(mat_32)
      [,1] [,2]
[1,]    1    5
[2,]   10   30
[3,]   15    8
>
> mat_23 <- matrix(
+   c(1,5,10,30,15,8),
+   nrow=2,
+   ncol=3,
+   byrow=TRUE)
> print(mat_23)
      [,1] [,2] [,3]
[1,]    1    5   10
[2,]   30   15    8
> |
```

Outra forma de se construir uma matriz:

Pode-se construir uma matriz a partir da definição de vetores através da função `rbind`.

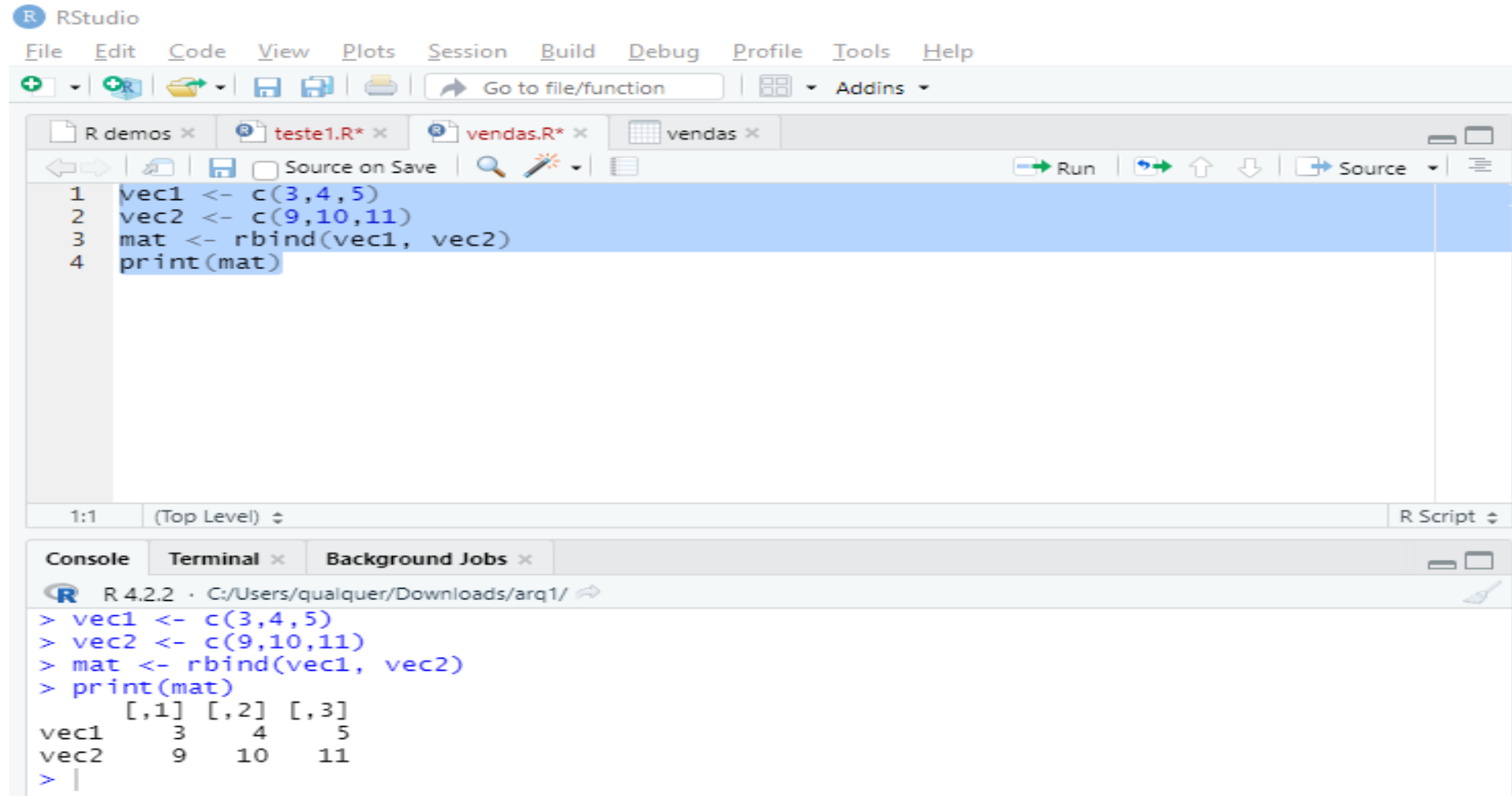
Para isto define-se as linhas e depois se contrói-se a matriz na sequência das linhas.

Exemplo:

```
vec1 <- c(1,2,3)
```

```
vec2 <- c(4,5,6)
```

```
Matriz <- rbind(vec1,vec2)
```



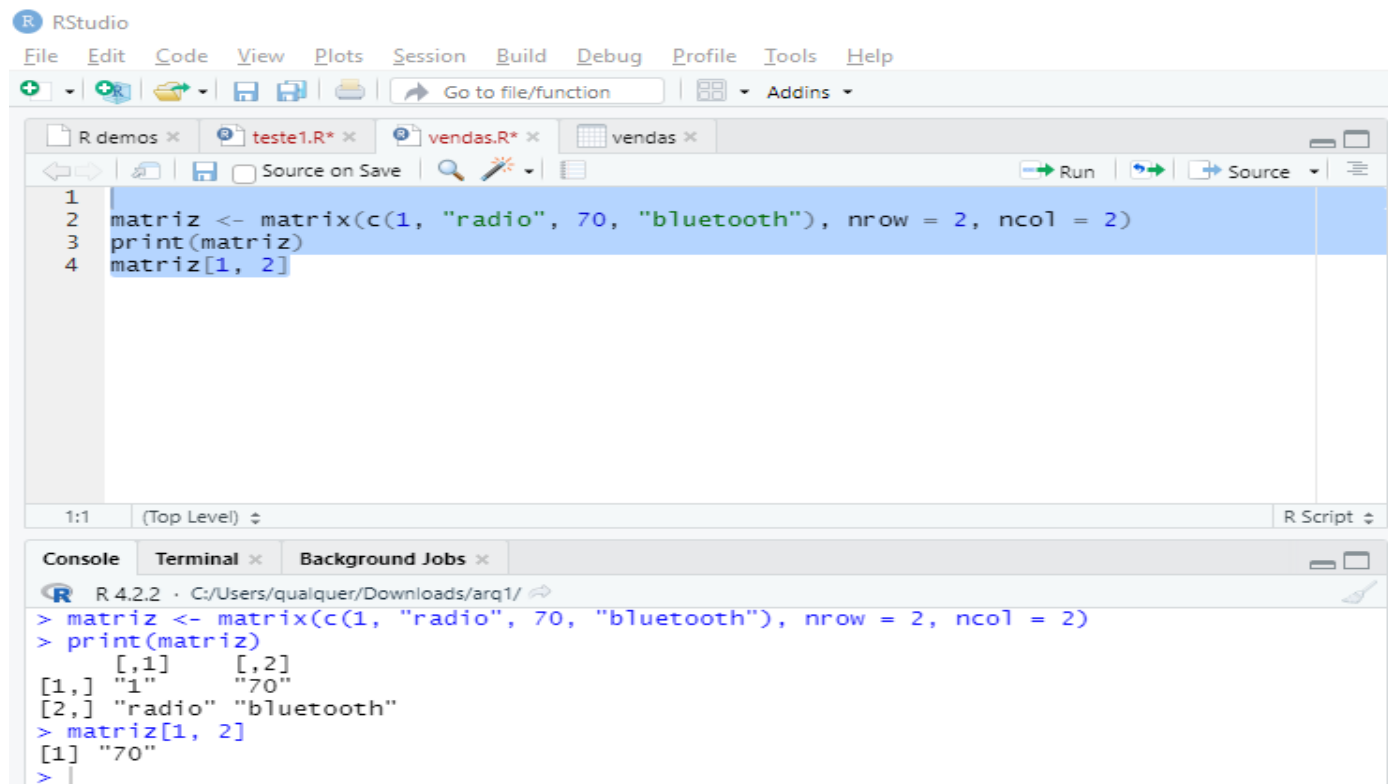
The screenshot shows the RStudio environment. The top menu bar includes File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, and Help. Below the menu is a toolbar with icons for file operations and a search bar labeled 'Go to file/function'. The editor pane shows a script with four lines of R code, with the first three lines highlighted in blue. The console pane at the bottom shows the output of the script, including the creation of vectors and a matrix, and the resulting matrix structure.

```
1 vec1 <- c(3,4,5)
2 vec2 <- c(9,10,11)
3 mat <- rbind(vec1, vec2)
4 print(mat)
```

```
> vec1 <- c(3,4,5)
> vec2 <- c(9,10,11)
> mat <- rbind(vec1, vec2)
> print(mat)
      [,1] [,2] [,3]
vec1    3    4    5
vec2    9   10   11
> |
```

Selecionando um elemento de uma matriz:

Basta usar-se `matriz[i,j]`.



The screenshot shows the RStudio interface. The script editor contains the following R code:

```
1 matriz <- matrix(c(1, "radio", 70, "bluetooth"), nrow = 2, ncol = 2)
2 print(matriz)
3 matriz[1, 2]
```

The console output shows the execution of the code:

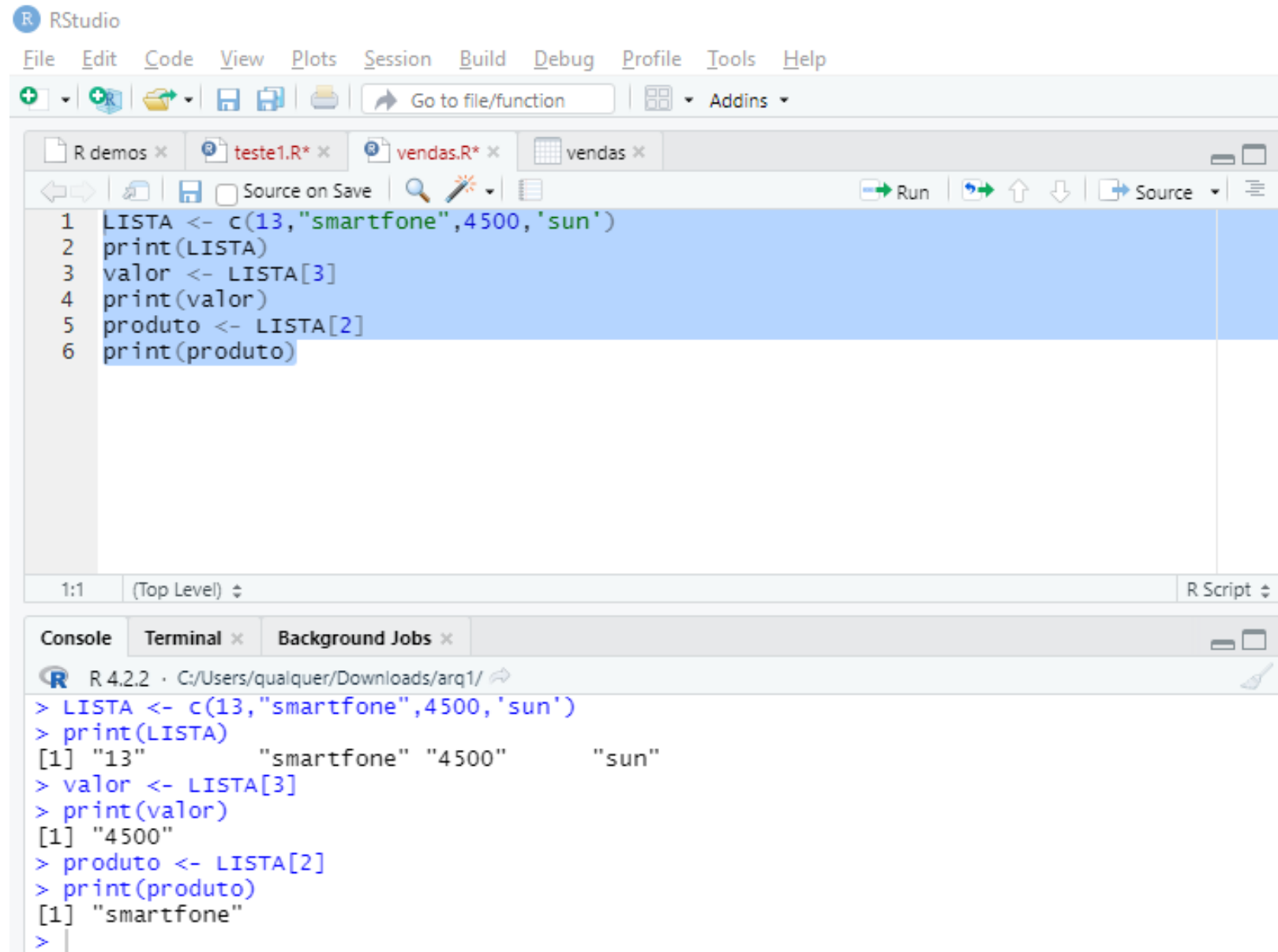
```
> matriz <- matrix(c(1, "radio", 70, "bluetooth"), nrow = 2, ncol = 2)
> print(matriz)
      [,1] [,2]
[1,]  "1"  "70"
[2,] "radio" "bluetooth"
> matriz[1, 2]
[1] "70"
>
```

Estruturas de dados no R:

LISTA:

São estruturas formadas por diversos tipos de elementos.

Para localizar um valor na lista usa-se:
`atributo <- nome_da_lista[índice]`



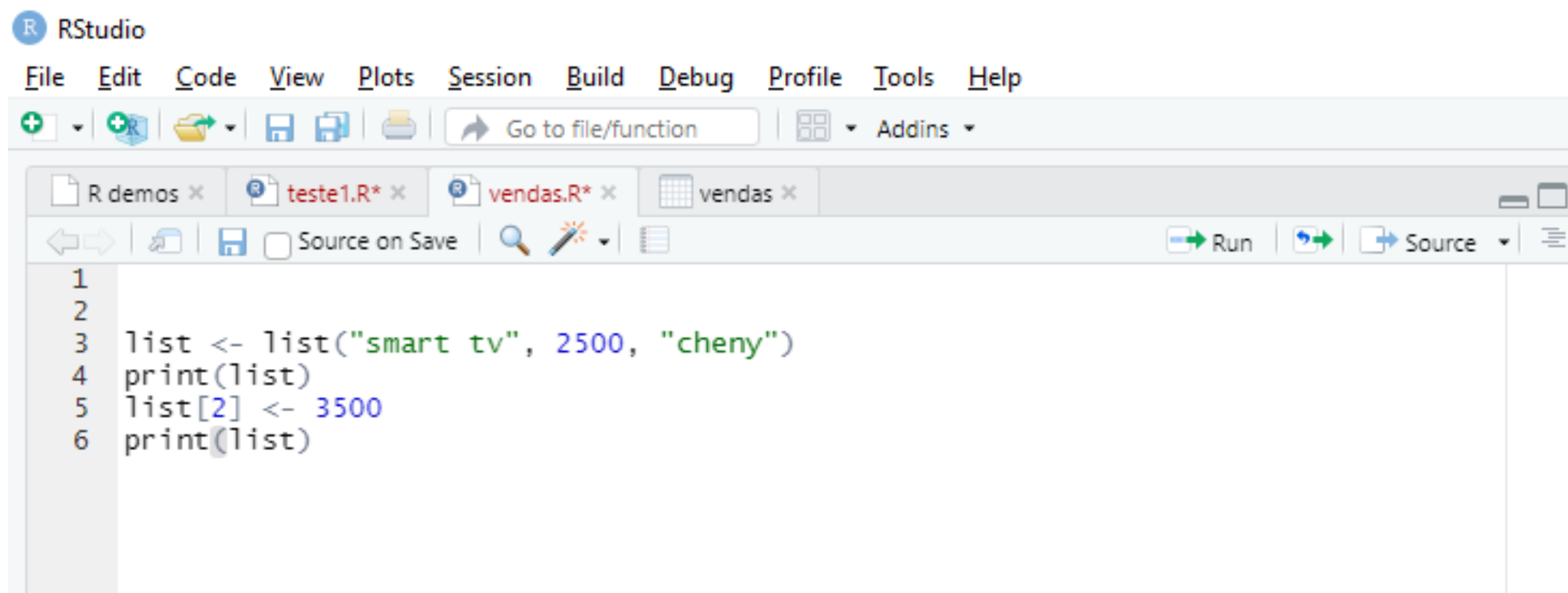
The screenshot displays the RStudio interface. The top menu bar includes File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, and Help. Below the menu is a toolbar with icons for file operations and a search bar. The editor pane shows a script with six lines of R code, with lines 1 through 6 highlighted in blue. The console pane at the bottom shows the execution of these lines, with the output of each command displayed. The status bar at the bottom indicates the current file is 'teste1.R*' and the session is at the 'Top Level'.

```
1 LISTA <- c(13,"smartfone",4500,'sun')
2 print(LISTA)
3 valor <- LISTA[3]
4 print(valor)
5 produto <- LISTA[2]
6 print(produto)
```

```
> LISTA <- c(13,"smartfone",4500,'sun')
> print(LISTA)
[1] "13"          "smartfone"  "4500"       "sun"
> valor <- LISTA[3]
> print(valor)
[1] "4500"
> produto <- LISTA[2]
> print(produto)
[1] "smartfone"
>
```

Alterar um elemento de uma lista:

Usa-se: `nome_da_lista[índice do elemento]<- novo valor`



The screenshot shows the RStudio environment. The menu bar includes File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, and Help. The toolbar contains icons for file operations and a search bar. The script editor shows the following code:

```
1  
2  
3 list <- list("smart tv", 2500, "cheny")  
4 print(list)  
5 list[2] <- 3500  
6 print(list)
```

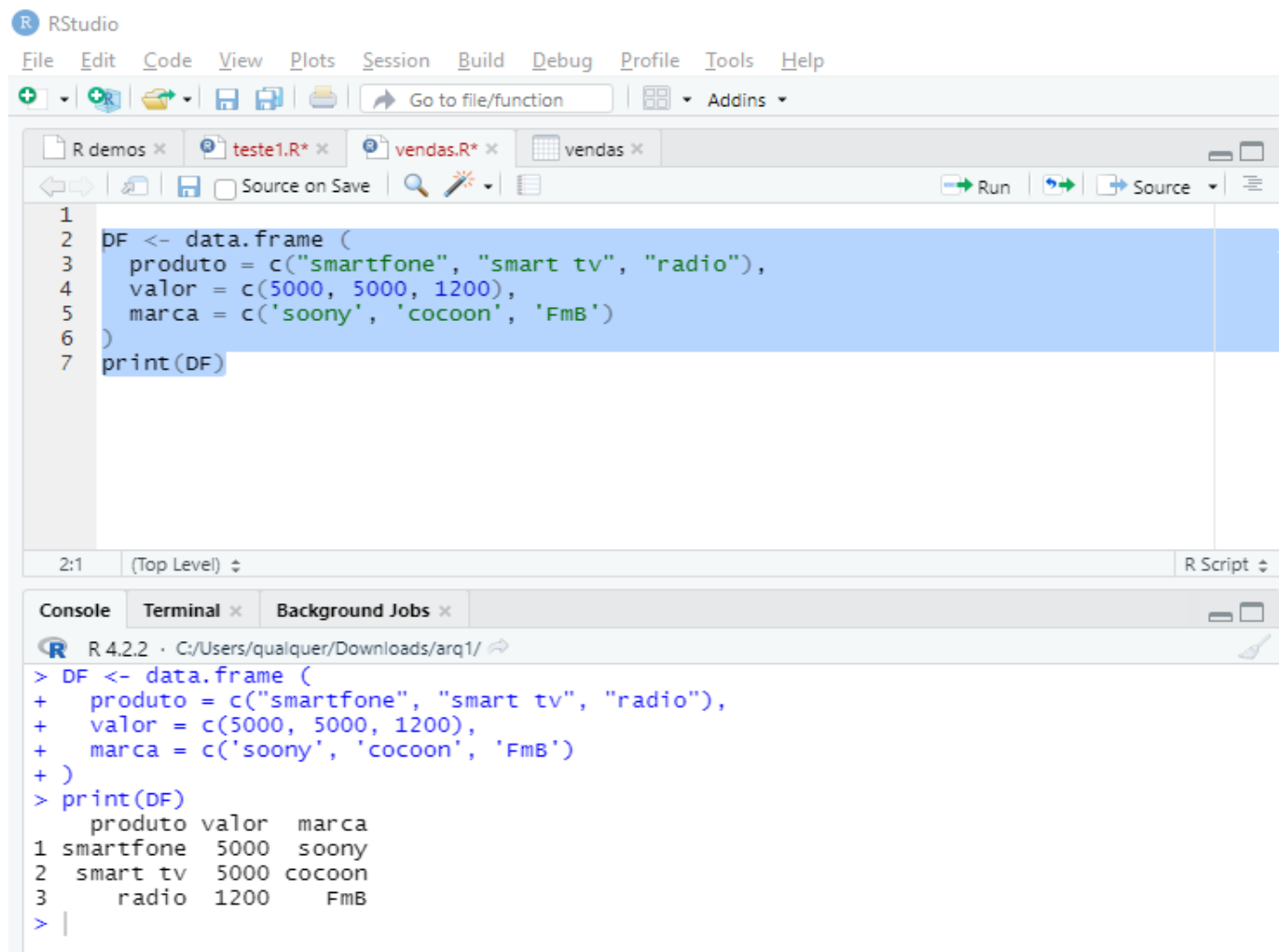
Estruturas de dados no R:

DataFrame:

É uma tabela de dados.

É uma estrutura de dados muito usada em Ciência de Dados.

Construção de um DataFrame:



The screenshot shows the RStudio interface. The top menu bar includes File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, and Help. Below the menu is a toolbar with icons for file operations and a search bar. The main editor window displays an R script with the following code:

```
1
2 DF <- data.frame (
3   produto = c("smartfone", "smart tv", "radio"),
4   valor = c(5000, 5000, 1200),
5   marca = c('soony', 'cocoon', 'FmB')
6 )
7 print(DF)
```

The bottom console window shows the output of the script:

```
R 4.2.2 · C:/Users/qualquer/Downloads/arq1/
> DF <- data.frame (
+   produto = c("smartfone", "smart tv", "radio"),
+   valor = c(5000, 5000, 1200),
+   marca = c('soony', 'cocoon', 'FmB')
+ )
> print(DF)
  produto valor marca
1 smartfone 5000 soony
2 smart tv  5000 cocoon
3   radio  1200   FmB
> |
```

Fatores:

FATORES são usadas para categorizar dados.

Exemplos de fatores:

- Sexo: Masculino/Feminino
- Tipo de Música: Rock, Pop, Clássica, Jazz

Para criar um fator usa-se a função `fator()`

Fatores:

Criando um fator:

```
tipo_música <-  
factor(c("Jazz", "Rock", "Classic", "Classic", "Pop", "Jazz", "Rock",  
        ", "Jazz"))
```

Imprimindo um fator:

```
tipo_música
```

Funções Básicas em R:

A função:

- `length(x)` retorna o comprimento de um objeto `x`.
- `sum(x)` soma todos os elementos de um objeto `x`.
- `rep(x,n)` repete o número `x`, `n` vezes.
- `seq(a, b, by=c)` gera uma sequência de números contidos entre `a` e `b`, distantes `c` unidades um do outro.
- `table(x)` retorna uma tabela com as frequências absolutas de ocorrência de cada elemento de `x`.

Funções em R:

Uma função é um bloco de código que só é executado quando este é chamado.

Pode-se passar dados (parâmetros) para uma função e esta função pode retornar dados como resultado.

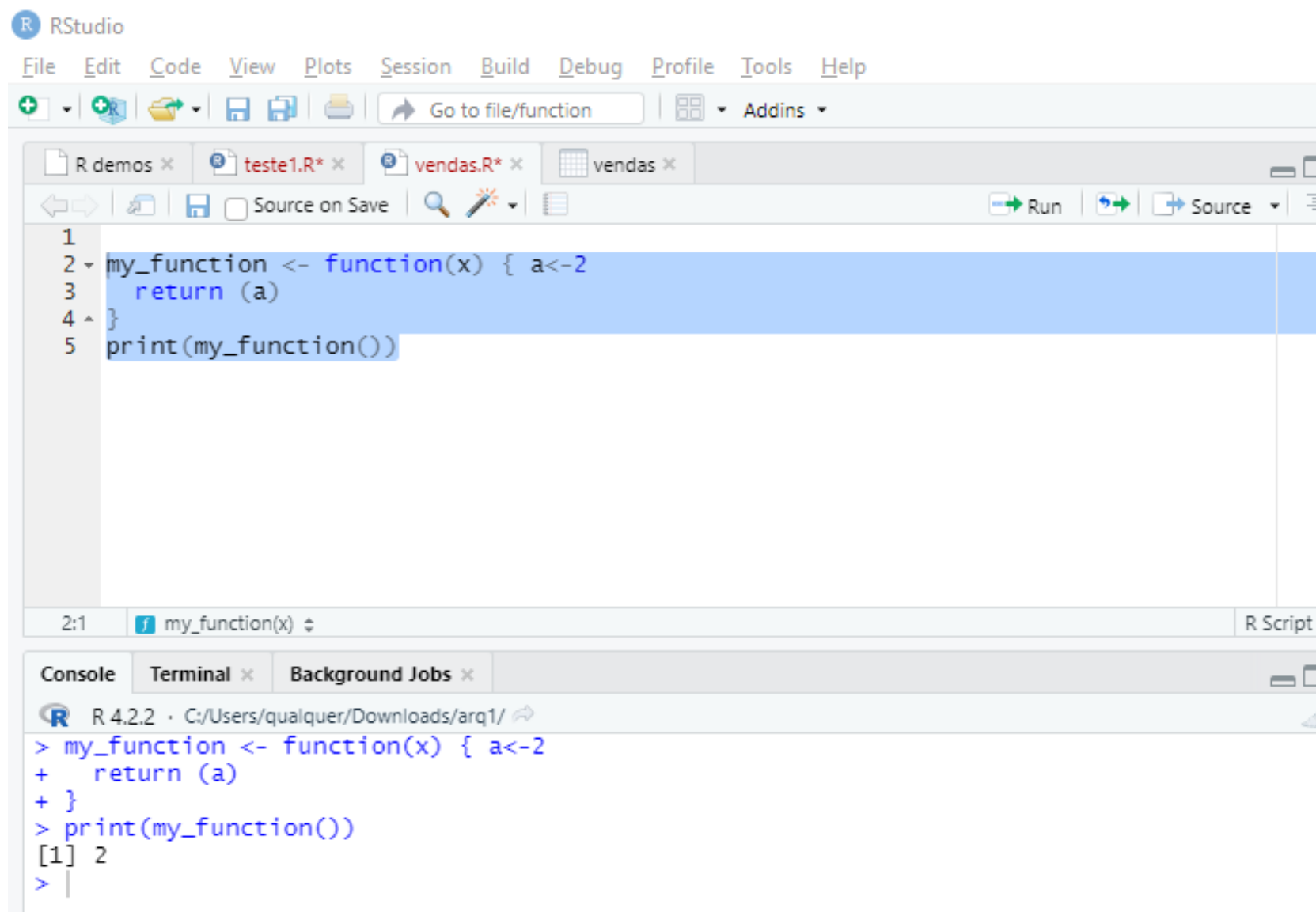
Criando uma função:

Para criar uma função use: `function()`

```
my_function <- function() { código }
```

Chamar uma função:

Para chamar uma função use: `my_function()`
`my_function <- function() { a<-2`
 `return (a)`
`}`
`print(my_function())`



The screenshot displays the RStudio interface. The top menu bar includes File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, and Help. Below the menu is a toolbar with icons for file operations and a search bar labeled 'Go to file/function'. The editor pane shows a script with the following R code:

```
1  
2 my_function <- function(x) { a<-2  
3   return (a)  
4 }  
5 print(my_function())
```

The code is highlighted in blue. The status bar at the bottom of the editor indicates '2:1' and 'my_function(x)'. The bottom pane shows the Console with the following output:

```
R 4.2.2 · C:/Users/qualquer/Downloads/arq1/  
> my_function <- function(x) { a<-2  
+   return (a)  
+ }  
> print(my_function())  
[1] 2  
>
```

The Console output shows the function definition and the result of the print statement, which is [1] 2.

Outro exemplo:

```
my_function <- function(x, y) {a <- x + y  
  return(a)  
}  
print(my_function(2,5))
```

Variáveis Globais:

São as variáveis criadas fora de uma função.

Exemplo:

```
txt <- "sensacional"
my_function <- function() {
  paste("R é", txt)
}
my_function()
```

Variáveis Globais:

```
imposto <- 0.05 #imposto de 5%  
my_function <- function() {  
  valor_imposto_produto=230*imposto  
  print(valor_imposto_produto)  
}  
my_function()
```