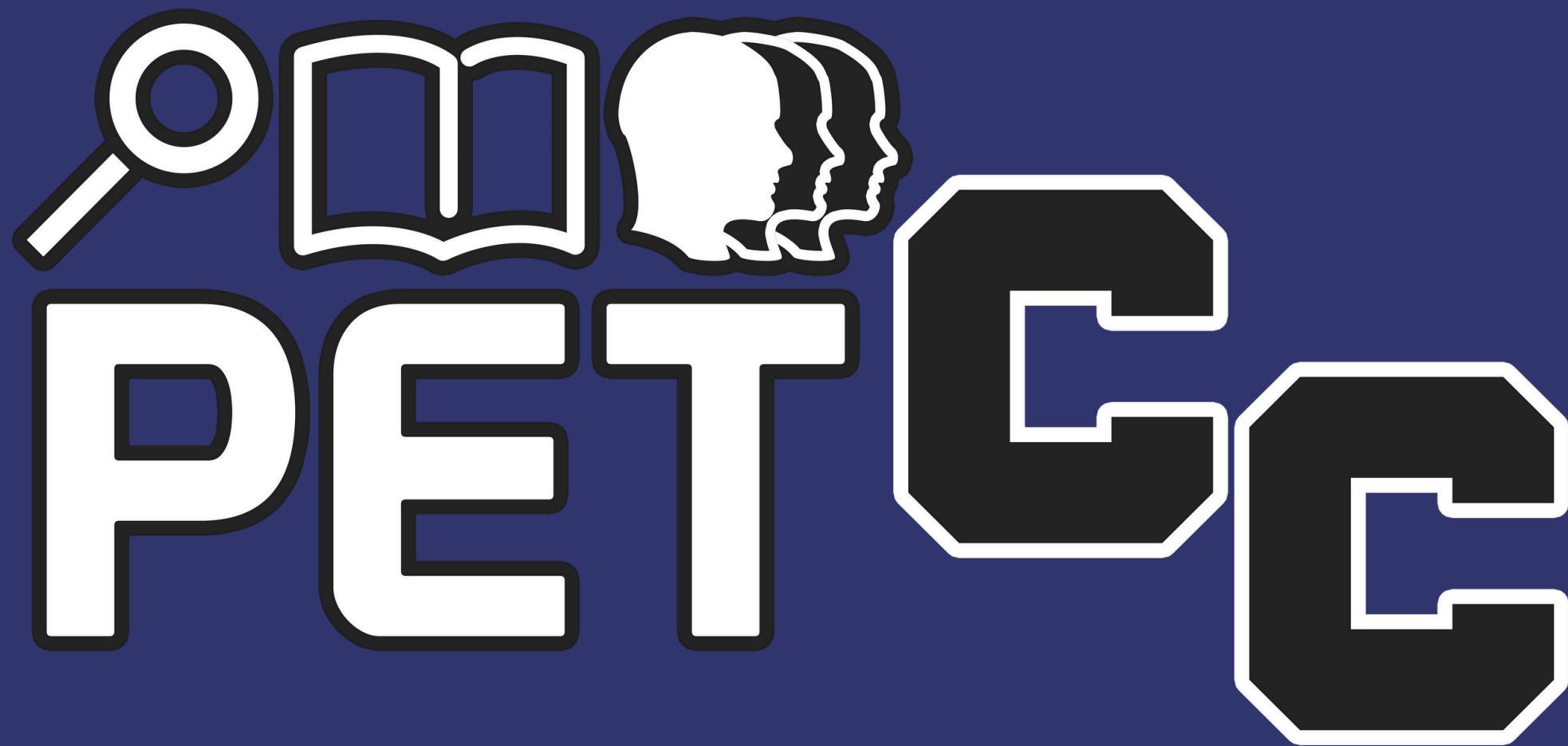


**MINICURSO**  
**BÁSICO DE**  
  
**PETCC**

**R**





# Apresentação da linguagem



# Manipulação de dados



# Análise Exploratória de Dados



# Visualização de Dados avançada

# Repositório do github do minicurso:

[github.com/gvheisler/minicurso-R](https://github.com/gvheisler/minicurso-R)

 minicurso-R Public

Unpin Unwatch 1 Fork 0 Star 1

main 1 Branch 0 Tags Go to file Add file Code

 jcaggens Update README.md dc9f21d · 15 hours ago 14 Commits

Aulas estrutura das pastas 3 weeks ago

Imagens estrutura das pastas 3 weeks ago

README.md Update README.md 15 hours ago

README

## Materiais do minicurso básico da linguagem R

*Minicurso oferecido pelo PET-CC da Universidade Federal de Santa Maria*





### Ementa

1. Introdução à Linguagem R
2. Manipulação de Dados
3. Análise Exploratória de Dados
4. Visualização de Dados

**About**

Repositório criado para servir como material de apoio no Minicurso de R

[www.ufsm.br/pet/ciencia-da-computacao](http://www.ufsm.br/pet/ciencia-da-computacao)

 r

 Readme

 Activity

 1 star

 1 watching

 0 forks

**Contributors** 3

 gvheisler Gabriel Vinícius Heisler

 jcaggens Josiane Aggens

 e-silveira Eduardo da Silveira

# Aula 01

Apresentação da linguagem



# O QUE É O R?

COMO SURGIU?

ONDE É USADO?

POR QUÊ USAR R?

# O que é o R?

- R é uma linguagem de programação focada em programação estatística e visualização de dados.
- É uma linguagem grátis e de código aberto. Grande parte das ferramentas utilizadas são bibliotecas feitas por usuários.
- É uma linguagem interpretada, não compilada (mostraremos o que isso significa).
- Uma das ferramentas mais utilizadas para programar em R é o RStudio.

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function Addins

Project: (None)

R Untitled1 x

Source on Save | Run | Source | Grid | Global Environment

1

1:1 (Top Level) R Script

Console Terminal Background Jobs

R 4.3.3 · ~/

```
R version 4.3.3 (2024-02-29 ucrt) -- "Angel Food Cake"
Copyright (C) 2024 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.
```

> |

Environment History Connections Tutorial

Import Dataset 95 MiB Grid

R Global Environment

Name	Type	Length	Size	Value
Environment is empty				

Files Plots Packages Help Viewer Presentation

Zoom Export

# Como surgiu?

Inspirada pela linguagem S, a linguagem R foi criada em 1993 para ensinar estatística introdutória na Universidade de Auckland. Apenas no ano de 2000 a versão 1.0 oficial foi lançada.

## Criadores

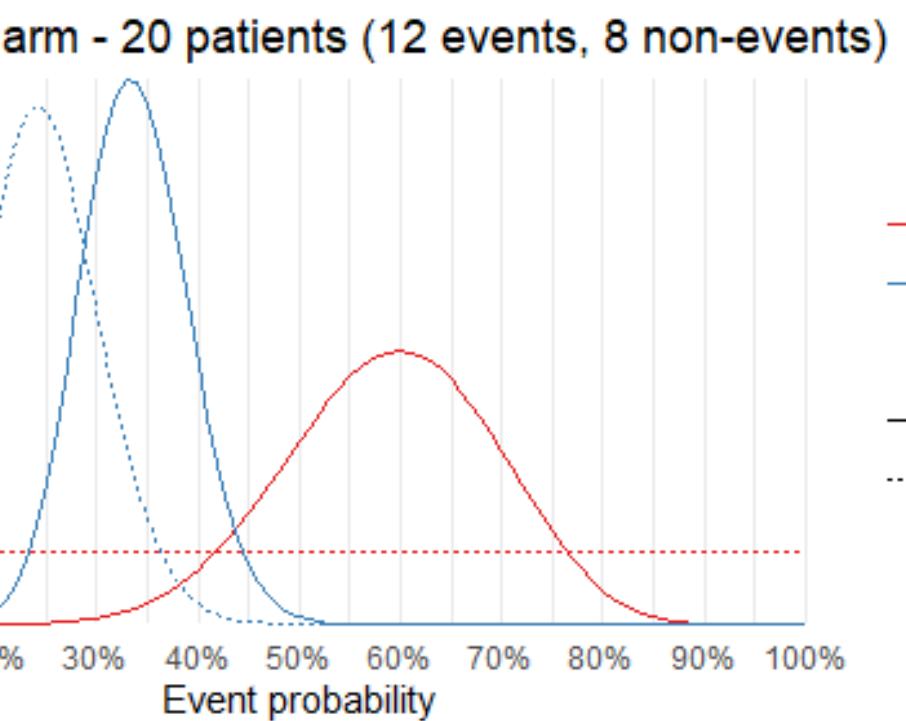
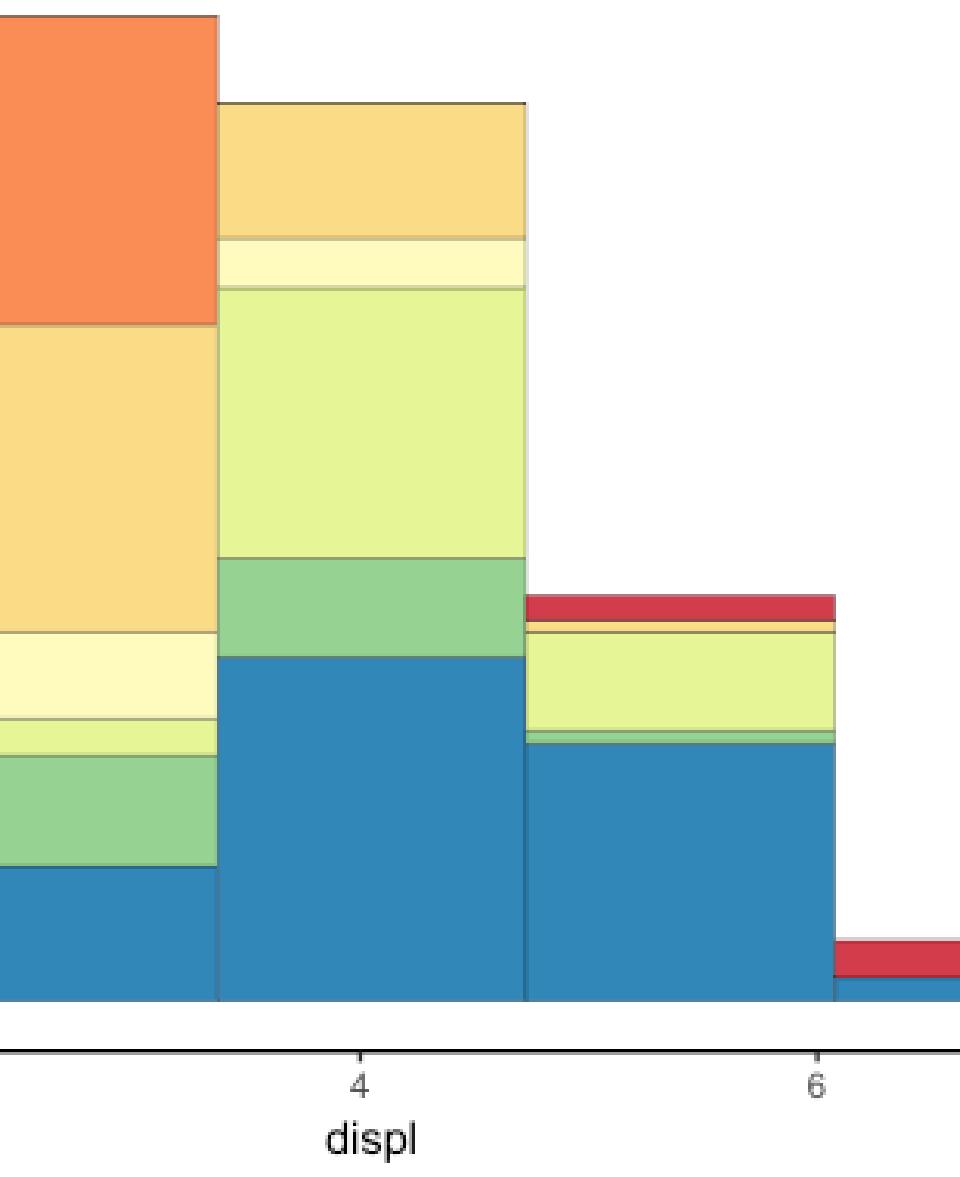


Ross  
Ihaka

Robert  
Gentleman

d Bins

ss Vehicle Classes



Westbound

Note: a route number can have several different trips, each with a different path. Only the most commonly-used path will be displayed on the map.

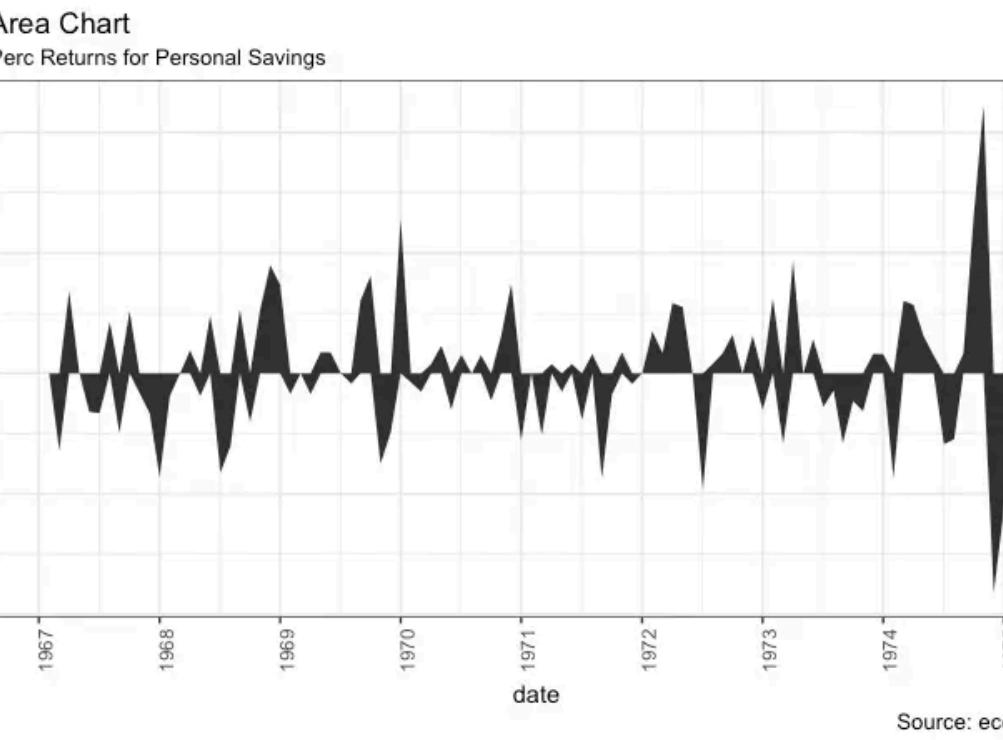
Zoom to fit buses

Refresh interval  
1 minute

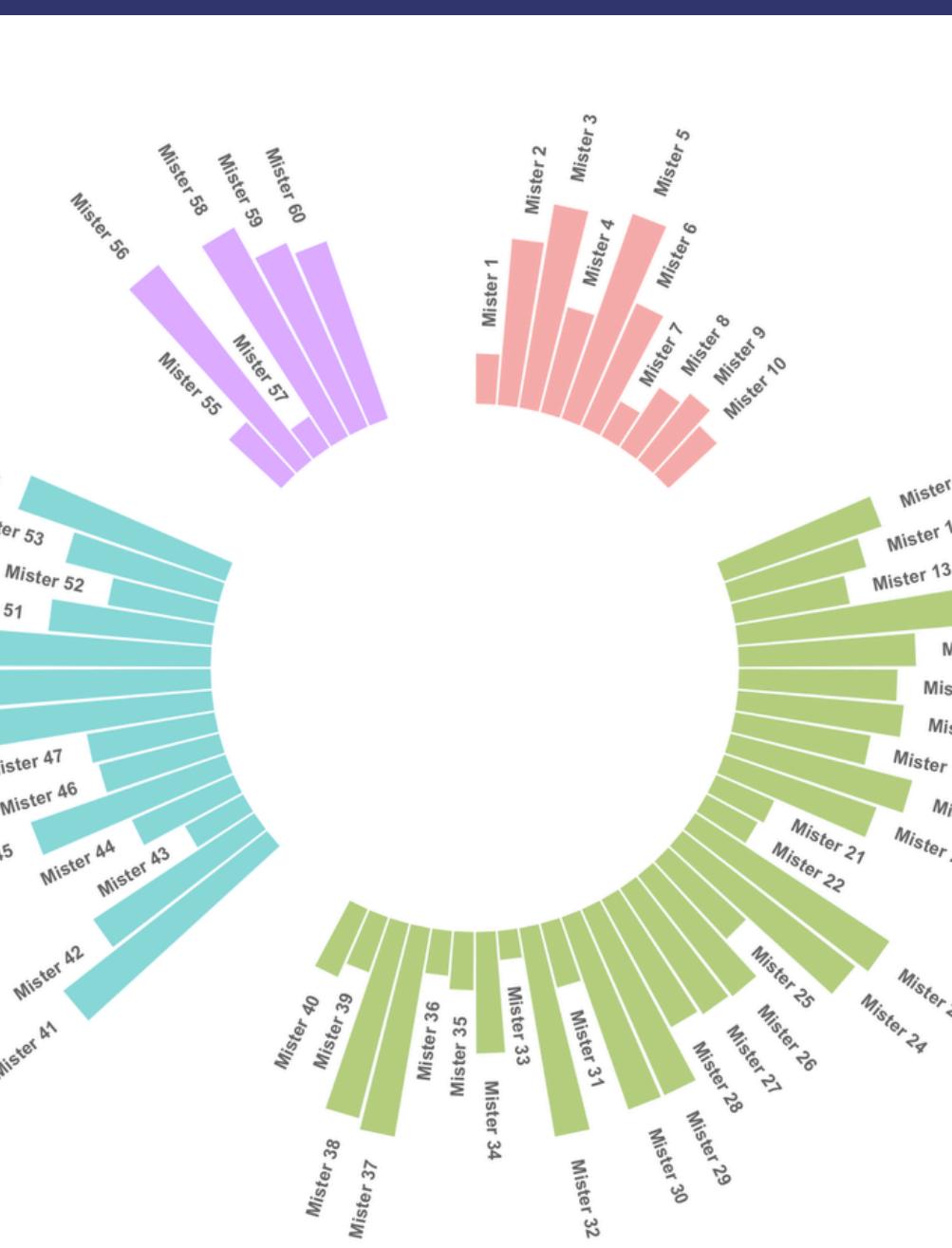
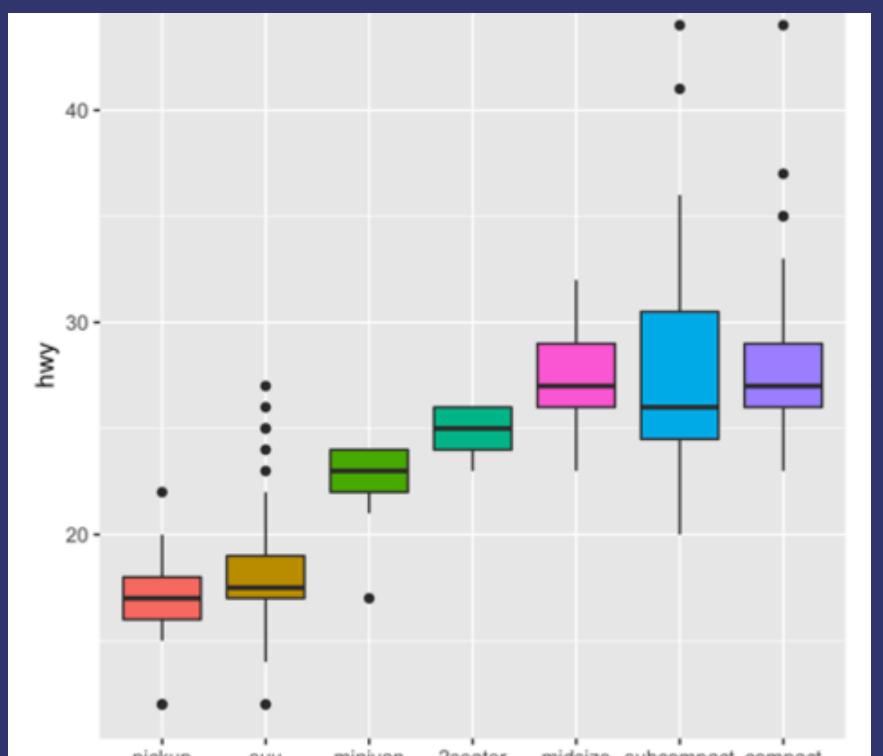
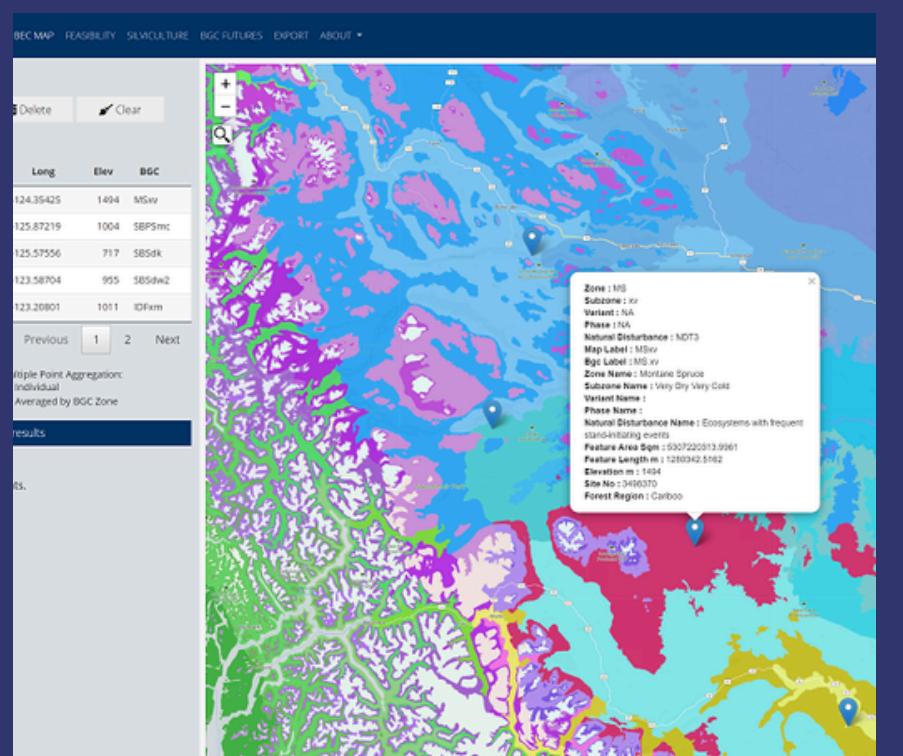
Data refreshed 0 seconds ago.

Refresh now

Source: esri.com



# Onde o R é utilizado?



# Por que usar R?

Linguagem  
amigável mesmo  
para não  
programadores

Grande  
comunidade,  
várias  
ferramentas

Integração com  
outras  
linguagens e com  
WEB

# INSTALANDO A FERRAMENTA

<https://posit.co/downloads/>



File Edit Code View Plots Session Build Debug Profile Tools Help

+ R Untitled1 x Go to file/function Addins

1

# Prática no RStudio

## 1 - Abra o RStudio

1:1 (Top Level) R Script

Console Terminal Background Jobs

R 4.3.3 · ~/

```
R version 4.3.3 (2024-02-29 ucrt) -- "Angel Food Cake"
Copyright (C) 2024 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.
```

Environment History Connections Tutorial

Import Dataset 95 MiB Grid

R Global Environment

Name	Type	Length	Size	Value
Environment is empty				

Files Plots Packages Help Viewer Presentation

Zoom Export

Untitled1 x

Source on Save | Run | Source | Grid | Environment | History | Connections | Tutorial | Import Dataset | 95 MiB | Global Environment | Name | Type | Length | Size | Value | Environment is empty |

1

Ambiente de escrita do código (salvável)

1:1 (Top Level) R Script

Console Terminal Background Jobs

R 4.3.3 · ~/

```
R version 4.3.3 (2024-02-29 ucrt) -- "Angel Food Cake"
Copyright (C) 2024 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.
```

## Terminal de execução

Variáveis carregadas no sistema

Files Plots Packages Help Viewer Presentation

Zoom Export

Environment is empty

## Gráfico, ajuda, arquivos

# Operadores

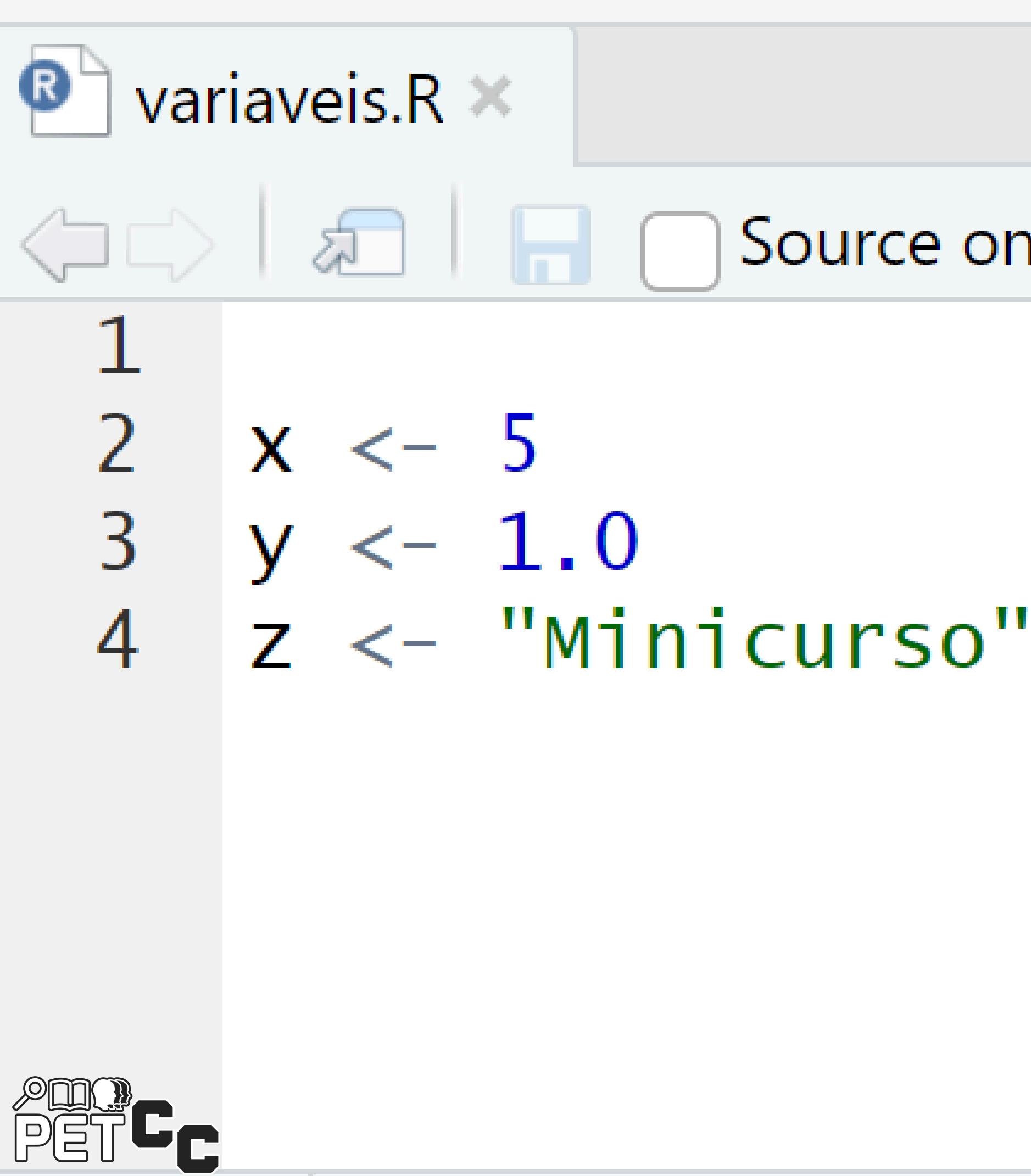
- O operador `<-` serve para designar um valor para uma variável
- Ou seja:
  - `x <- 5`
  - A variável `x` recebe o valor 5

# Operadores

- O operador + serve para somar
- Ou seja:
  - $x \leftarrow x + 5$
  - A variável  $x$  recebe o valor  
dela mesma + 5

# Operadores

- Os operadores básicos são
  - + (adição)
  - - (subtração)
  - \* (multiplicação)
  - / (divisão)
  - entre outros



```
R variaveis.R ×
← → | ↕ | H | Source on
1
2 x <- 5
3 y <- 1.0
4 z <- "Minicurso"
```

# Declaração de variáveis

Diferente, por exemplo, da linguagem C, no R não é necessário declarar uma variável com o tipo.

**CTRL + Enter para executar uma linha**

The screenshot shows the RStudio interface. In the top-left, a script file named 'variaveis.R' is open, containing the following code:

```
1 x <- 5
2 y <- 1.0
3 z <- "Minicurso"
```

In the top-right, the Environment pane shows the Global Environment with the message "Environment is empty".

At the bottom, the Console pane shows the R prompt: > |

The screenshot shows the RStudio interface with the following components:

- Code Editor:** Displays the script file `variaveis.R` containing the following code:

```
1 x <- 5
2 y <- 1.0
3 z <- "Minicurso"
```
- Environment View:** Shows the global environment with the following objects:

Name	Type	Length	Size	Value
x	numerical	1	56 B	5
y	numerical	1	56 B	1
z	character	1	120 B	"Minicurso"
- Console View:** Displays the R session history:

```
R 4.3.3 · ~/minicurso> x <- 5
R 4.3.3 · ~/minicurso> y <- 1.0
R 4.3.3 · ~/minicurso> z <- "Minicurso"
R 4.3.3 · ~/minicurso>
```

**Section Header:** A large, bold, dark blue text overlay reads "As variáveis ficam salvas na memória".

The screenshot shows the RStudio interface with the following components:

- Code Editor:** Displays the script file `variaveis.R` containing the following R code:

```
1 x <- 5
2 y <- 1.0
3 z <- "Minicurso"
4 variavel <- x + y
5 print(variavel)
```
- Environment View:** Shows the global environment with the following objects and their values:

Name	Type	Length	Size	Value
variavel	numerical	1	56 B	6
x	numerical	1	56 B	5
y	numerical	1	56 B	1
z	character	1	120 B	"Minicurso"
- Console View:** Displays the R session history:

```
R 4.3.3 · ~/minicurso
> x <- 5
> y <- 1.0
> z <- "Minicurso"
> variavel <- x + y
> print(variavel)
[1] 6
> |
```

**Text Overlay:** A large, bold, dark blue text block in the center-left of the slide reads:

**Para printar algo usa-se  
print(nome\_da\_variavel)**

# Sua vez:

Faça com que uma variável x receba o cálculo:

$$12*7$$

faça com que uma variavel y receba o cálculo:

$$x + 15$$

printe o valor da variável y na tela.

# Tipos de dados

- Character
- Numeric
- Integer
- Complex
- Logical

# Outras características

- 1 Basicamente, todas variáveis são vetores
- 2 Os vetores são iniciados em 1
- 3 Existem loops, estruturas condicionais, etc

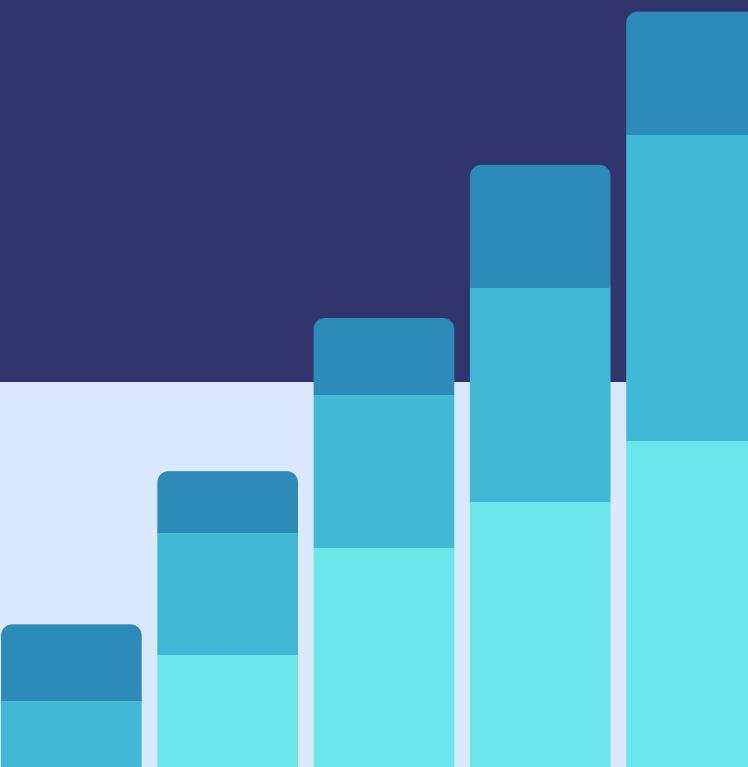
**MINICURSO**  
**BÁSICO DE**  
  
**PETCC**

**R**

# Aula 02

**Manipulação de dados**

# MANIPULAÇÃO DE DADOS



- Tipos de dados
- Formatos de dados
- Importar e Exportar
- Como usar e manipular esses dados
- Seleção e filtragem de dados
- Agrupamento e summarização

# O que são dados?

Um dado é qualquer informação que pode ser coletada, registrada e processada por um sistema. Ele pode ser numérico, textual, visual ou de outro tipo e serve como base para análises, decisões ou operações computacionais.

# O que são dados?

Os dados, por si só, são elementos brutos, mas quando organizados, processados e interpretados, podem se transformar em conhecimento útil. Eles podem estar armazenados em diversos formatos e são essenciais para praticamente todas as áreas, desde ciência e negócios até tecnologia e pesquisa.

# TIPOS DE DADOS

ESTRUTURADOS

NÃO  
ESTRUTURADOS

SEMI  
ESTRUTURADOS

# Dados estruturados

São organizados em formatos definidos, como tabelas com linhas e colunas (como arquivos CSV ou bancos de dados relacionais)

A	B	C	D	E
Last Name	Sales	Country	Quarter	
Smith	\$16,753.00	UK	Qtr 3	
Johnson	\$14,808.00	USA	Qtr 4	
Williams	\$10,644.00	UK	Qtr 2	
Jones	\$1,390.00	USA	Qtr 3	
Brown	\$4,865.00	USA	Qtr 4	
Williams	\$12,438.00	UK	Qtr 1	
Johnson	\$9,339.00	UK	Qtr 2	
Smith	\$18,919.00	USA	Qtr 3	
Jones	\$9,213.00	USA	Qtr 4	
Jones	\$7,433.00	UK	Qtr 1	
Brown	\$3,255.00	USA	Qtr 2	
Williams	\$14,867.00	USA	Qtr 3	
Williams	\$19,302.00	UK	Qtr 4	
Smith	\$9,698.00	USA	Qtr 1	

# Dados semi estruturados

Eles têm alguma organização, mas não seguem um modelo rígido, como JSON, XML ou até arquivos de logs.

Embora tenham uma estrutura, ela não é tão fixa quanto nos dados estruturados.

```
orders": [
  {
    "orderno": "748745375",
    "date": "June 30, 2088 1:54:23 AM",
    "trackingno": "TN0039291",
    "custid": "11045",
    "customer": [
      {
        "custid": "11045",
        "fname": "Sue",
        "lname": "Hatfield",
        "address": "1409 Silver St",
        "city": "Ashland",
        "state": "NE",
        "zip": "68003"
      }
    ]
}
```

# Dados não estruturados

São aqueles que não têm uma estrutura predefinida. Exemplos incluem imagens, vídeos, áudios, e textos sem formatação (como artigos ou postagens em redes sociais).



# Formato CSV

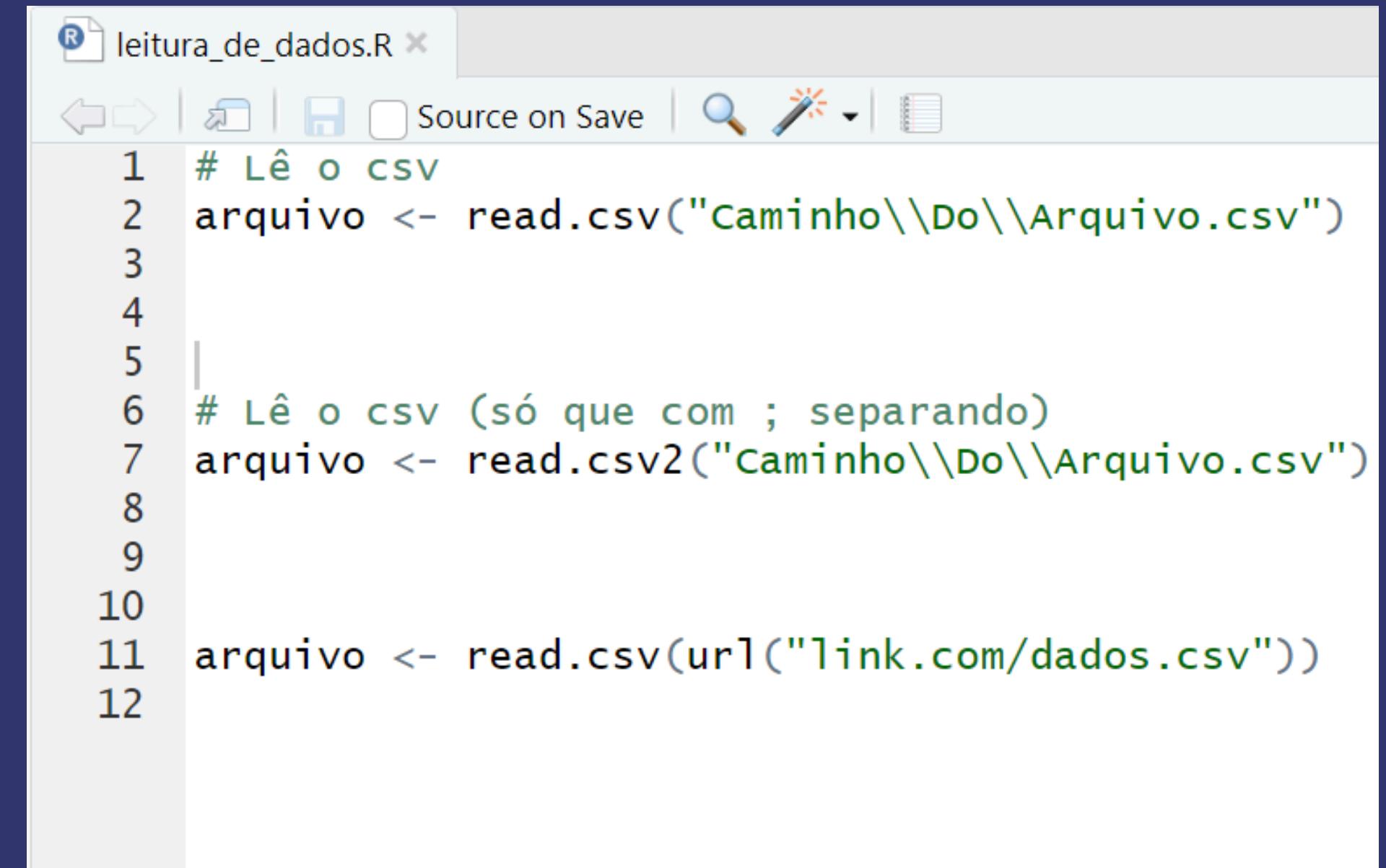
O formato CSV (Comma-Separated Values) é um dos mais simples e amplamente utilizados para armazenar dados tabulares. Nele, os valores de cada linha são separados por vírgulas, representando colunas de uma tabela, e cada linha corresponde a um registro.

# Formato CSV

Por ser um formato de texto simples, o CSV é fácil de criar, editar e compartilhar entre diferentes sistemas e plataformas. Apesar de não suportar recursos avançados como fórmulas ou metadados, sua simplicidade o torna ideal para manipulação de grandes volumes de dados e para importação/exportação entre softwares.

# Importação de dados em R

Para importar dados em R,  
geralmente usa-se o formato CSV.



The screenshot shows an RStudio interface with a code editor window titled "leitura\_de\_dados.R". The window contains three examples of R code for reading CSV files:

```
1 # Lê o csv
2 arquivo <- read.csv("caminho\\Do\\Arquivo.csv")
3
4
5
6 # Lê o csv (só que com ; separando)
7 arquivo <- read.csv2("caminho\\Do\\Arquivo.csv")
8
9
10
11 arquivo <- read.csv(url("link.com/dados.csv"))
12
```

# Exemplo

	Nome	Idade	Salario	Departamento
1	Ana	28	3500	Marketing
2	Bruno	34	4800	TI
3	Carla	29	4000	Financeiro
4	Daniel	42	5200	TI
5	Eduarda	31	4500	Marketing
6	Felipe	25	3200	Financeiro
7	Gustavo	38	5000	TI
8	Helena	26	3700	Marketing

<https://gvheisler.github.io/ds/departamentos.csv>

Ao ler um csv pelo comando `read.csv` ele automaticamente é salvo no formato dataframe  
Este formato é bastante parecido, por exemplo, com uma planilha do excel  
No RStudio, dataframes podem ser visualizados facilmente através do comando `view(df)`, ou da interface

	Nome	Idade	Salario	Departamento
1	Ana	28	3500	Marketing
2	Bruno	34	4800	TI
3	Carla	29	4000	Financeiro
4	Daniel	42	5200	TI
5	Eduarda	31	4500	Marketing
6	Felipe	25	3200	Financeiro
7	Gustavo	38	5000	TI
8	Helena	26	3700	Marketing

<https://gvheisler.github.io/ds/departamentos.csv>

# Exemplo

Podemos acessar o conteúdo do dataframe de formas diferentes

- Pelo índice da linha/coluna:
  - `df[1,1]` acessa a primeira linha e primeira coluna
  - `df[,1]` acessa toda a primeira coluna
  - `df[1,]` acessa toda a primeira linha
  - `df$Nome` acessa toda a coluna “Nome”

	Nome	Idade	Salario	Departamento
1	Ana	28	3500	Marketing
2	Bruno	34	4800	TI
3	Carla	29	4000	Financeiro
4	Daniel	42	5200	TI
5	Eduarda	31	4500	Marketing
6	Felipe	25	3200	Financeiro
7	Gustavo	38	5000	TI
8	Helena	26	3700	Marketing

<https://gvheisler.github.io/ds/departamentos.csv>

# Exemplo

Podemos colocar mais valores caso o objetivo seja acessar várias colunas, ou colunas específicas

- `df[,c(1,3)]` cria um novo dataframe apenas com as colunas 1 e 3
- `df[which(df$Departamento=="TI"),]` retornará um dataset apenas com as linhas onde o departamento é TI

	Nome	Idade	Salario	Departamento
2	Bruno	34	4800	TI
4	Daniel	42	5200	TI
7	Gustavo	38	5000	TI

# Exemplo

	Nome	Idade	Salario	Departamento
1	Ana	28	3500	Marketing
2	Bruno	34	4800	TI
3	Carla	29	4000	Financeiro
4	Daniel	42	5200	TI
5	Eduarda	31	4500	Marketing
6	Felipe	25	3200	Financeiro
7	Gustavo	38	5000	TI
8	Helena	26	3700	Marketing

`df[c(2:5),]` retorna um dataset com as linhas 2, 3, 4, 5

`df$Salario <- df$Salario * 2` dobra o salário de todo mundo

`mean(df$Salario)` retorna a média de todos os salários

<https://gvheisler.github.io/ds/departamentos.csv>

```
mean(df[which(df$Departamento=="TI"&df$Idade>35), "Salario"])
mean(df$Salario[which(df$Departamento=="TI"&df$Idade>35)])
```

```
> mean(df[which(df$Departamento=="TI"&df$Idade>35), "salario"])
[1] 5100
> mean(df$salario[which(df$Departamento=="TI"&df$Idade>35)])
[1] 5100
```

# Prática

Leia um dataset a partir da url

<https://gvheisler.github.io/ds/alunos.csv>

Calcule a média de presenças (coluna Attendance) de todos os alunos

Depois, calcule a média de presenças (coluna Attendance) dos alunos cujo envolvimento parental (coluna Parental\_involvement) é “Low”, e depois “High”