

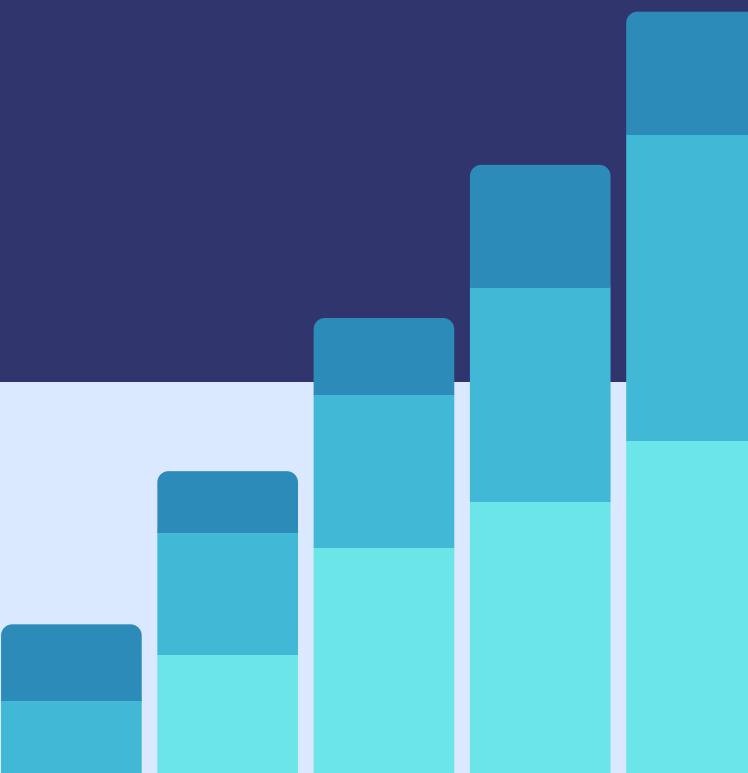
**MINICURSO**  
**BÁSICO DE**  
  
**PETCC**

**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

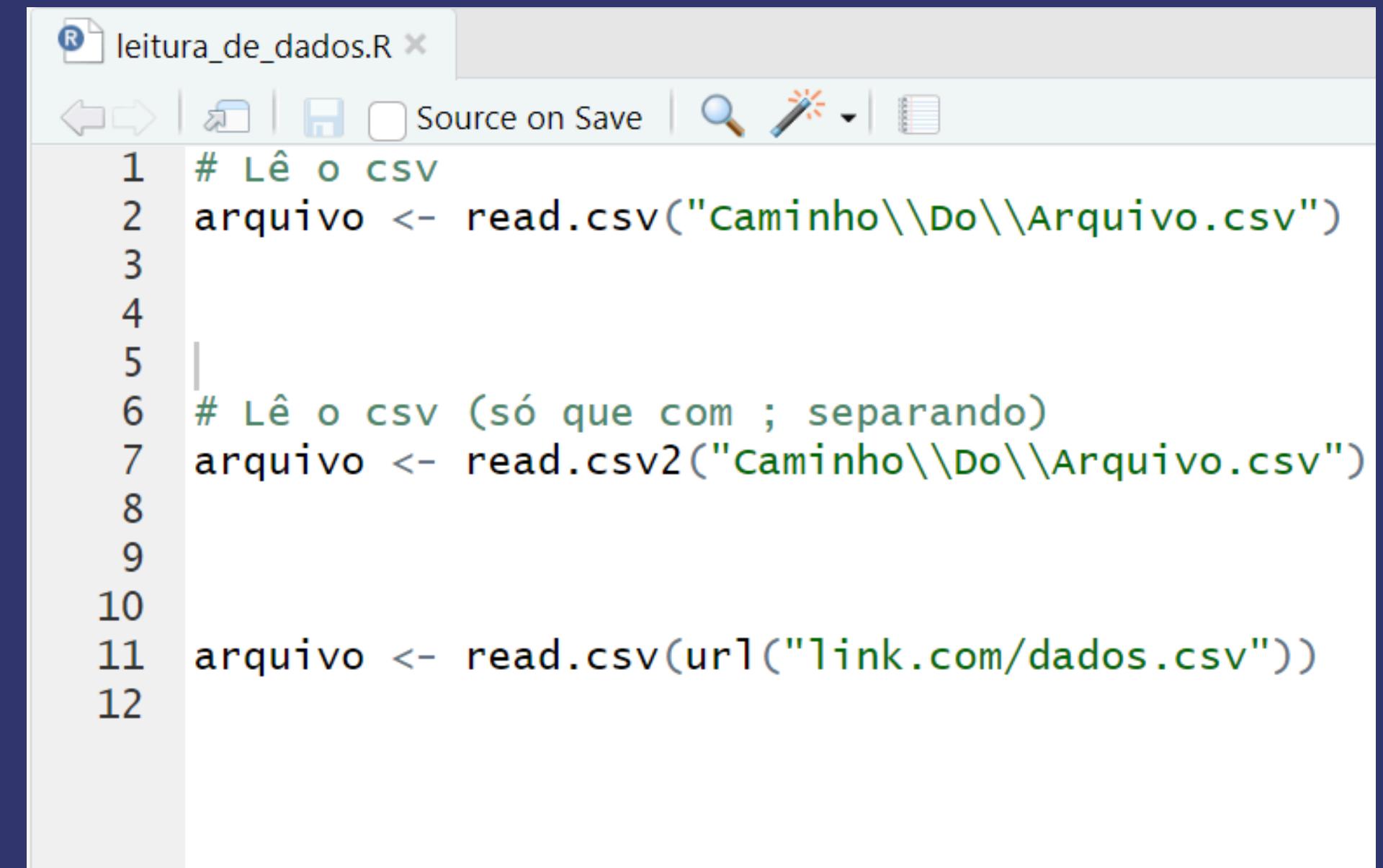
O formato CSV (Comma-Separated Values) é um dos mais simples e amplamente utilizados para armazenar dados tabulares. Nele, os valores de cada linha são separados por vírgulas, representando colunas de uma tabela, e cada linha corresponde a um registro.

# 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”

```
> mean(df$Attendance)
[1] 79.97745
> mean(df$Attendance [which(df$Parental_Involvement=="Low")])
[1] 80.30516
> mean(df$Attendance [which(df$Parental_Involvement=="High")])
[1] 79.94811
```