



Laboratório de Biologia Computacional e Molecular

Centro de Biotecnologia da UFRGS  
Universidade Federal do Rio Grande do Sul



# R para Ciências da Vida (BCM13065) Aula 7

PPGBCM - UFRGS

Diego Bonatto  
2024/2

# Por que vetorizar comandos?

- Vetorização em R é uma técnica fundamental que permite realizar operações em todos os elementos de um vetor, matriz ou data frame de uma só vez, sem a necessidade de utilizar laços “for” ou “while”.
- Essa abordagem é extremamente eficiente e é uma das razões pelas quais o R é tão popular para análises de dados.

# Tidyverse

- O tidyverse é uma coleção de pacotes R projetados para trabalhar de forma integrada, proporcionando uma abordagem consistente e eficiente para a manipulação de dados.
- Esses pacotes são conhecidos por sua sintaxe intuitiva e por facilitarem a limpeza, transformação, visualização e modelagem de dados



## R packages for data science

The tidyverse is an opinionated **collection of R packages** designed for data science. All packages share an underlying design philosophy, grammar, and data structures.

Install the complete tidyverse with:

```
install.packages("tidyverse")
```

## Learn the tidyverse

See how the tidyverse makes data science faster,

# Tibble

- Um tibble é um tipo de data frame moderno, introduzido pelo pacote “tibble”. É uma estrutura de dados tabular que oferece várias vantagens em relação aos data frames tradicionais:
  - Impressão mais limpa: Os tibbles são impressos de forma mais compacta e informativa.
  - Manipulação mais fácil: O dplyr é otimizado para trabalhar com tibbles.
  - Integração com outros pacotes: Tibbles são compatíveis com uma ampla gama de pacotes, incluindo ggplot2, purrr, e outros.

# Tibble

country	year	cases	population
Afghanistan	1999	3775	19987071
Afghanistan	2000	2666	20595360
Brazil	1999	37737	172006362
Brazil	2000	80488	174504898
China	1999	212258	1272915272
China	2000	213766	128042583

variables

country	year	cases	population
Afghanistan	1999	3775	19987071
Afghanistan	2000	2666	20595360
Brazil	1999	37737	172006362
Brazil	2000	80488	174504898
China	1999	212258	1272915272
China	2000	213766	128042583

observations

country	year	cases	population
Afghanistan	1999	3775	19987071
Afghanistan	2000	2666	20595360
Brazil	1999	37737	172006362
Brazil	2000	80488	174504898
China	1999	212258	1272915272
China	2000	213766	128042583

values

Dataframe:

```
> print(df)
```

	coluna_numerica	coluna_caracter	coluna_logica
1	1	A	TRUE
2	2	B	FALSE
3	3	C	TRUE
4	4	D	FALSE
5	5	E	TRUE

Tibble:

```
> print(tibble_df)
```

```
# A tibble: 5 × 3
```

	coluna_numerica	coluna_caracter	coluna_logica
	<i>&lt;int&gt;</i>	<i>&lt;chr&gt;</i>	<i>&lt;lgl&gt;</i>
1	1	A	TRUE
2	2	B	FALSE
3	3	C	TRUE
4	4	D	FALSE
5	5	E	TRUE



# dplyr

## Overview

dplyr is a grammar of data manipulation, providing a consistent set of verbs that help you solve the most common data manipulation challenges:

- `mutate()` adds new variables that are functions of existing variables
- `select()` picks variables based on their names.
- `filter()` picks cases based on their values.
- `summarise()` reduces multiple values down to a single summary.
- `arrange()` changes the ordering of the rows.

These all combine naturally with `group_by()` which allows you to perform any operation “by group”. You can learn more about them in `vignette("dplyr")`. As well as these single-table verbs, dplyr also provides a variety of two-table verbs, which you can learn about in `vignette("two-table")`.

If you are new to dplyr, the best place to start is the [data transformation chapter](#) in R for Data Science.

## Quando usar dplyr?

- Projetos onde a legibilidade e a simplicidade são prioritárias.
- Datasets pequenos a médios (<1 GB).
- Trabalhos em equipe ou com colaboradores que têm pouca experiência com R.
- Quando já se está utilizando outras ferramentas do tidyverse.



# data.table



CRAN **Not OK** R-CMD-check **failing** codecov **99%** pipeline **passed** downloads **1M/month**  
 CRAN usage **1594** BioC usage **324** indirect usage **3997**

`data.table` provides a high-performance version of [base R's data.frame](#) with syntax and feature enhancements for ease of use, convenience and programming speed.

## Why data.table?

- concise syntax: fast to type, fast to read
- fast speed
- memory efficient
- careful API lifecycle management
- community
- feature rich

## Features

- fast and friendly delimited **file reader**: `?fread`, see also [convenience features for small data](#)
- fast and feature rich delimited **file writer**: `?fwrite`
- low-level **parallelism**: many common operations are internally parallelized to use multiple CPU threads
- fast and scalable aggregations; e.g. 100GB in RAM (see [benchmarks](#) on up to **two billion rows**)
- fast and feature rich joins: **ordered joins** (e.g. rolling forwards, backwards, nearest and limited staleness), **overlapping range joins** (similar to `IRanges::findOverlaps`), **non-equi joins** (i.e. joins using operators `>`, `>=`, `<`, `<=`), **aggregate on join** ( `by=.EACHI` ), **update on join**
- fast add/update/delete columns **by reference** by group using no copies at all
- fast and feature rich **reshaping** data: `?dcast` (*pivot/wider/spread*) and `?melt` (*unpivot/longer/gather*)
- **any R function from any R package** can be used in queries not just the subset of functions made available by a database backend, also columns of type `list` are supported
- has **no dependencies** at all other than base R itself, for simpler production/maintenance

## Quando usar data.table?

- Manipulação de grandes datasets (>1 GB), especialmente em sistemas com menos memória disponível.
- Quando o desempenho (tempo de execução) é crítico.
- Cenários em que múltiplas manipulações complexas e eficientes precisam ser realizadas em uma única etapa.
- Projetos que requerem controle detalhado sobre memória e desempenho.