

Boas-vindas ao Curso Básico de R do BMClima!

Este curso é a porta de entrada para o mundo da análise de dados em saúde e clima. O objetivo é preparar você para analisar dados e dominar a linguagem R para análise de dados.



by Priscilla Normando

Introdução ao R

Linguagem de Programação

O R é uma linguagem de programação poderosa e flexível, ideal para análise estatística e visualização de dados.

Ferramentas Complementares

O R é utilizado em conjunto com outras ferramentas, como RStudio, Google Colab, Git e GitHub, que facilitam o desenvolvimento e a colaboração.

Vantagens

O R é gratuito, de código aberto e possui uma comunidade ativa, com uma ampla gama de pacotes e ferramentas disponíveis.

Aplicações

O R é utilizado em diversas áreas, como saúde, meio ambiente, economia e marketing, para análise de dados complexos e geração de insights.



Instalando as Ferramentas

R

Baixe o R do CRAN (Comprehensive R Archive Network) e instale seguindo as instruções para o seu sistema operacional.

1

Git

Baixe o Git do site oficial e instale para controlar versões do código e facilitar o compartilhamento de projetos.

2

3

Google Colab

Crie uma conta no Google Colab (opcional) para utilizar o R em um ambiente de nuvem, ideal para colaboração e projetos online.

4

5

6

RStudio

Baixe o RStudio do site oficial e instale após instalar o R. O RStudio é um ambiente de desenvolvimento integrado (IDE) que facilita o uso do R.

GitHub

Crie uma conta no GitHub para hospedar seus projetos, colaborar com outros desenvolvedores e compartilhar código.

Ferramentas Adicionais

Explore pacotes adicionais, como Latex, webshot e bookdown, que podem ser úteis para gerar relatórios, gráficos e livros.

Primeiros Passos na Programação

1 Conceitos Básicos

Aprenda sobre variáveis, constantes, operadores, tipos de dados e estruturas de dados, como vetores, matrizes, listas e dataframes.

2 Expressões e Funções

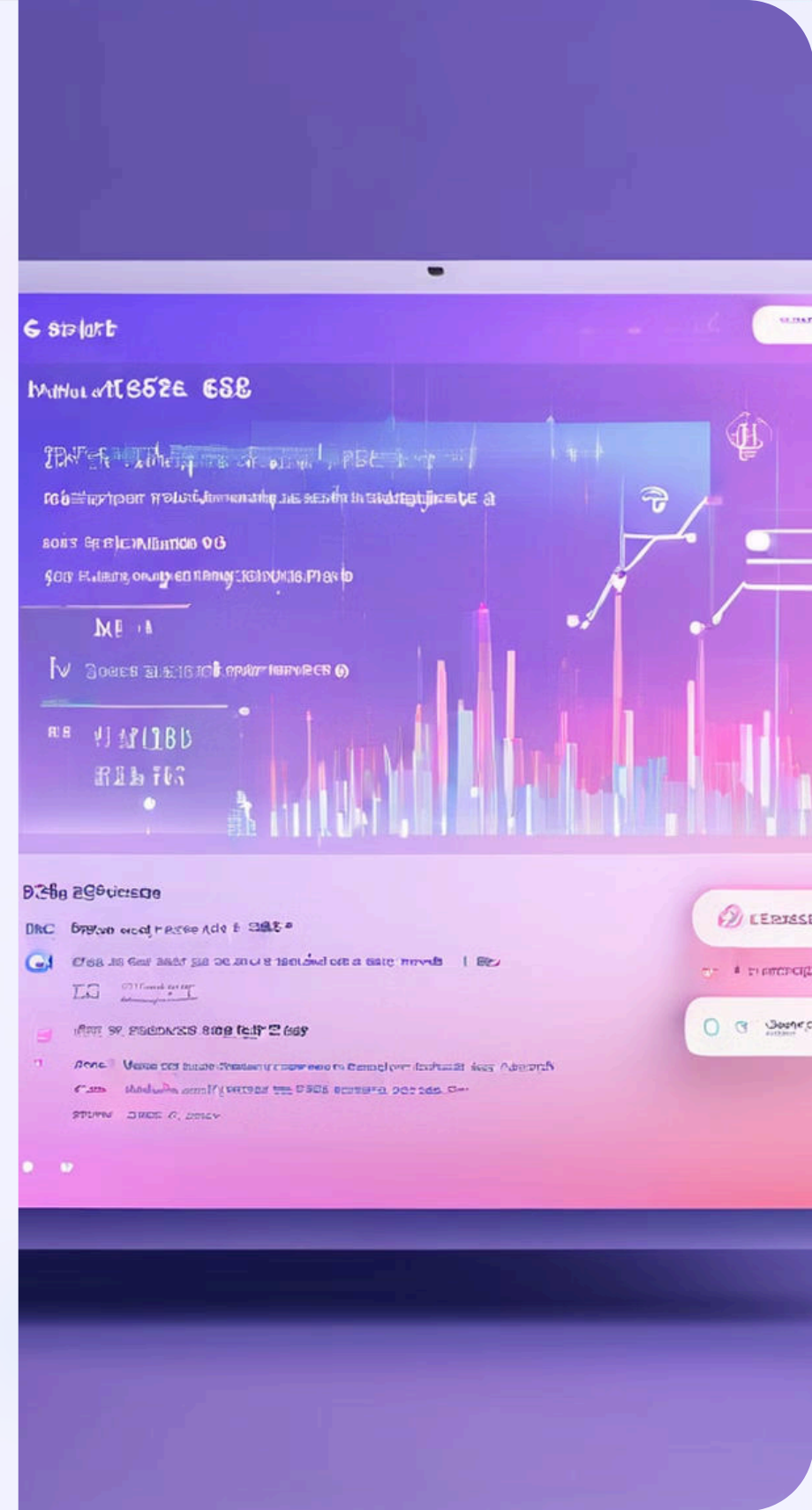
Entenda como funcionam expressões, fórmulas e funções, que são blocos de código reutilizáveis que facilitam a programação em R.

3 Organização do Código

Crie uma pasta de projeto, utilize a ferramenta "Projetos" do RStudio e desative as opções .RData e .Rhistory para manter seu trabalho organizado e evitar problemas.

4 Primeiro Script

Crie um arquivo chamado primeiroScript.R, salve-o e copie e cole o código fornecido para realizar cálculos simples e entender a interface do RStudio.



Funções e Pacotes

Funções

As funções são blocos de código reutilizáveis que executam tarefas específicas, como `print()`, `c()`, `as.Date()`.

Objetos

Os objetos são nomes que guardam valores, como variáveis, constantes, etc. Utilize nomes descritivos e siga as regras de nomeação para manter seu código organizado.

Pacotes

Os pacotes são conjuntos de funções que estendem as capacidades do R. Utilize `install.packages("nome_do_pacote")` para instalar e `library("nome_do_pacote")` para carregar os pacotes.

Analizando Dados do SIM

1

Instalação e Carregamento

Instale e carregue os pacotes necessários para o projeto, como remotes, microdatasus, tidyverse e readr.

2

Configuração da Pasta

Utilize as funções `getwd()` e `setwd()` para definir a pasta de trabalho e organizar os arquivos do projeto.

3

Download dos Dados

Baixe os dados do SIM (Sistema de Informação de Mortalidade) utilizando o pacote `microdatasus`.

4

Pré-processamento

Utilize a função `process_sim()` para pré-processar os dados do SIM, limpando e organizando as informações.

5

Inspeção e Análise

Utilize as funções `head()`, `summary()` e `str()` para inspecionar a estrutura dos dados e realizar análises estatísticas básicas.

Operador Pipe e Entrada/Saída de Dados



Operador Pipe

O operador pipe (`%>%`) encadeia funções de forma mais legível e eficiente. Em vez de criar variáveis intermediárias, você pode conectar funções diretamente, como `dados %>% funcao1() %>% funcao2() %>% funcao3()`.



Entrada de Dados

Utilize funções como `readline()` para ler dados do teclado, `read.csv()` e `read.table()` para ler arquivos, e pacotes como DBI e RMySQL para acessar bancos de dados.



Saída de Dados

Exiba os resultados no console com `print()` e `cat()`, salve em arquivos com `write.csv()` e `write.table()`, ou utilize funções de plotagem para gerar gráficos.



Manipulação de Strings

Utilize funções para concatenar, extrair, substituir, converter, remover espaços, dividir, verificar padrões, formatar e manipular strings.



Estruturas de Controle e Loops

Estruturas de Decisão

if, else, ifelse

switch

case_when

Loops

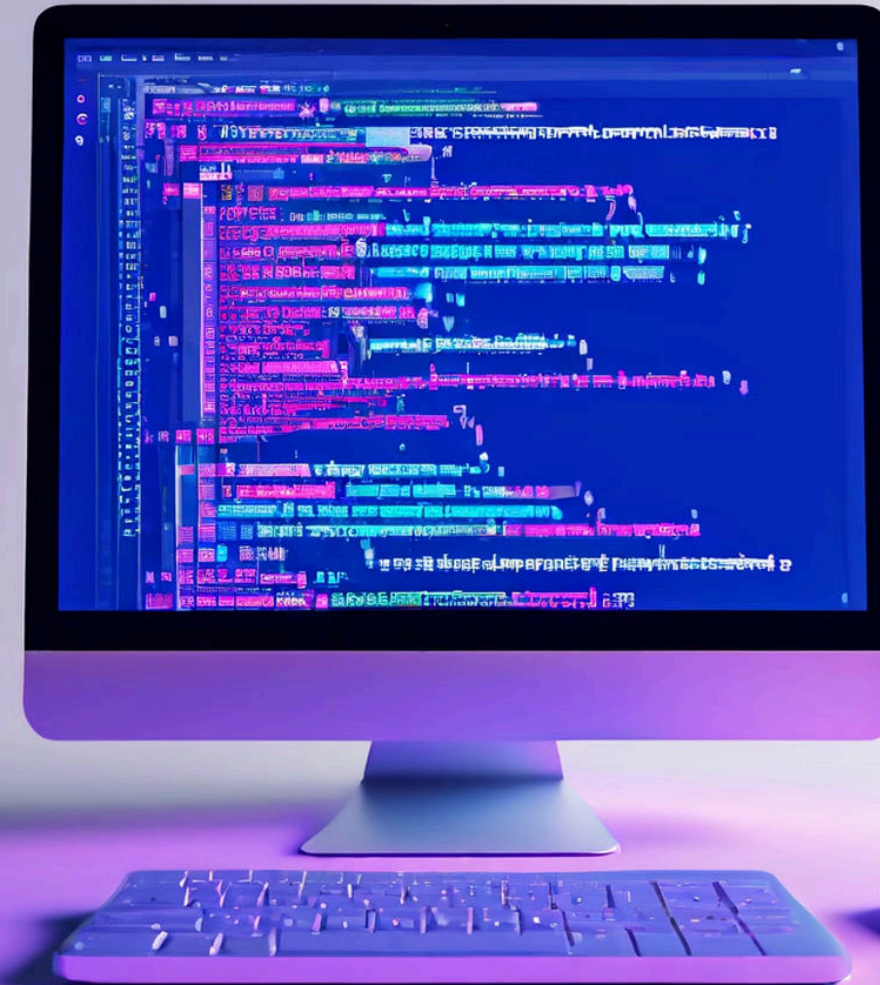
for

while

repeat

lapply(), sapply()

apply(), tapply(), mapply()



Pipeline de Dados

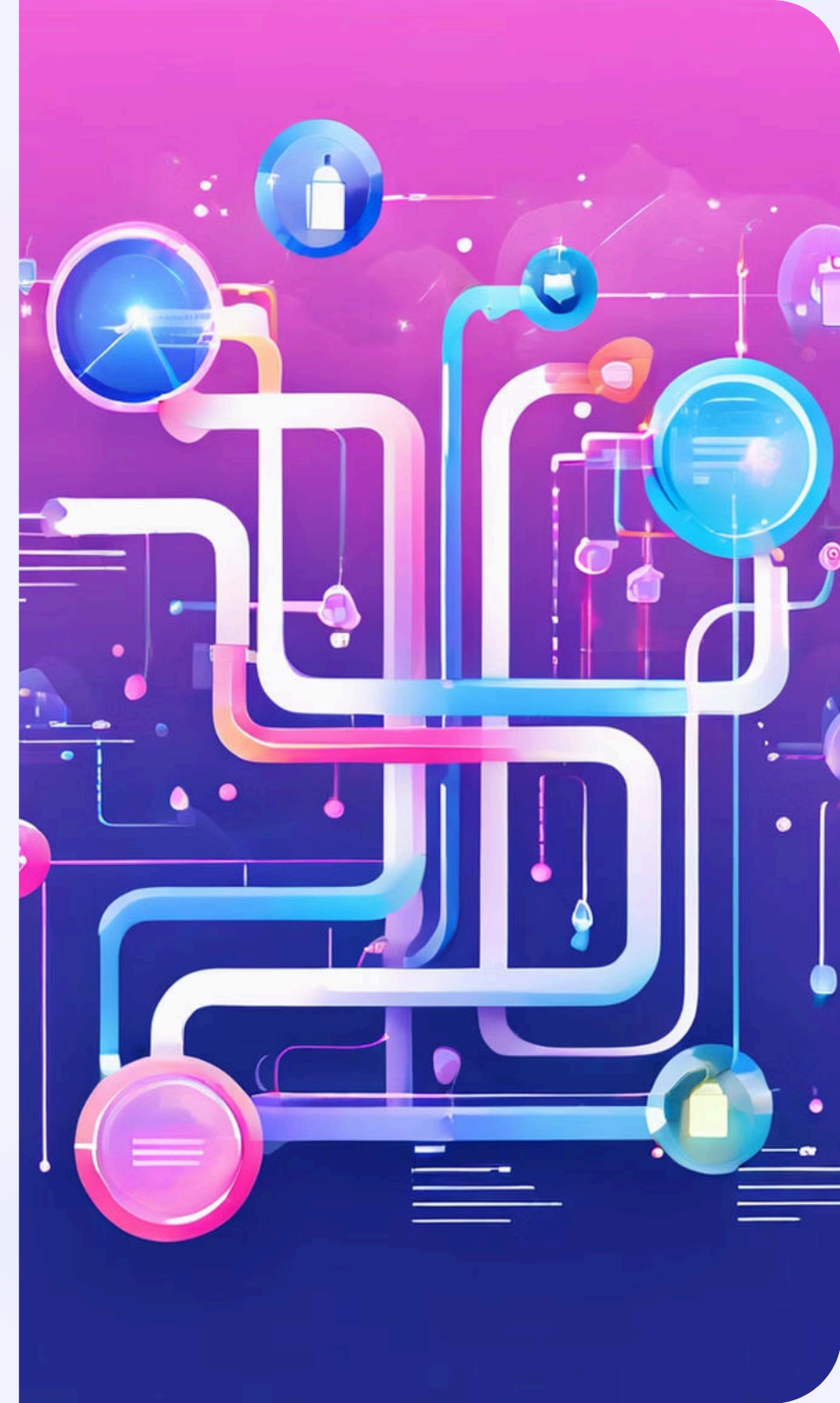
- 1 Coleta**

Extraia dados de diversas fontes, como bancos de dados, APIs, arquivos CSV, etc. Utilizando o pacote microdatasus, você pode coletar dados do DataSUS.
- 2 Limpeza**

Remova dados duplicados, inconsistentes ou incorretos. Padronize os formatos para garantir a integridade dos dados.
- 3 Transformação**

Converta os dados em um formato adequado para análise. Agrupe, filtre e calcule novas métricas para obter insights relevantes.
- 4 Armazenamento**

Carregue os dados processados em um local de destino, como um banco de dados, data warehouse ou data lake.





Projeto Final

Utilize o script fornecido como inspiração para construir um script em R que solucione o cenário proposto. Inclua as cinco etapas de uma pipeline de dados, coletando e processando dados do DataSUS com o pacote microdatasus.