**Para saber mais: instalação do R, R Studio e importação do projeto**

**P**ara você que se sente mais confortável em trabalhar com os dados em sua máquina ou gostaria de realizar muitos testes que podem consumir mais dados que os disponíveis de maneira gratuita pelo Posit Cloud, é possível seguir, de maneira **OPCIONAL**, o seguinte passo a passo para fazer o seu projeto pelo **R Studio** na versão desktop.

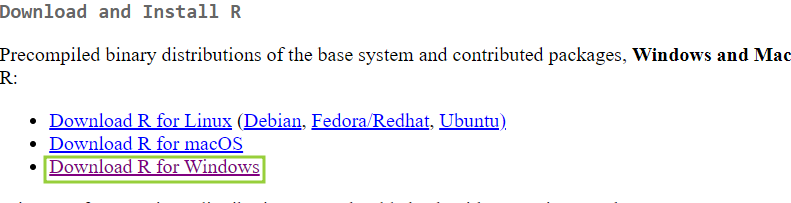
### Instalação do R, R Studio e importação do projeto

Para conseguir utilizar a linguagem R no seu computador e criar ou executar projetos em sua máquina, o primeiro passo é a instalação do **R** e do **R Studio**.

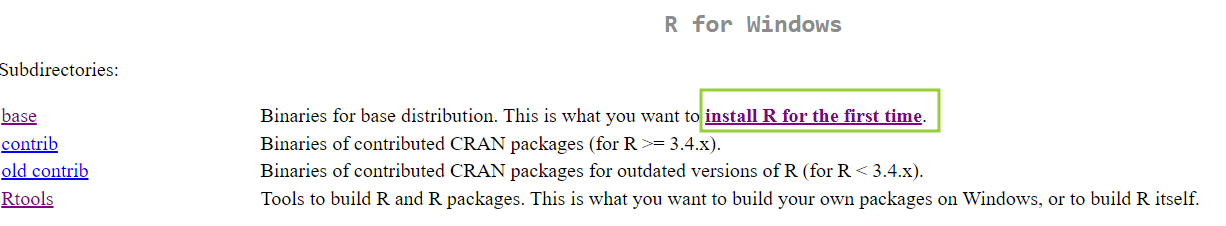
O R é o pacote com a linguagem em si, enquanto o R Studio é um ambiente de desenvolvimento integrado (IDE) que facilita a escrita, execução e organização de código R. Vamos seguir o passo a passo para instalar essas ferramentas e importar o nosso projeto.

#### Instalação do R

Acesse o [site oficial do R](https://cran.r-project.org/) e escolha o sistema operacional (SO) que fará o download do pacote. Vamos apresentar o processo com o SO do Windows.



Após escolher o SO, se estiver baixando o R pela primeira vez clique no botão **install R for the first time** na página que abriu.



Agora, basta clicar em **Download R for Windows** para baixar o instalador.

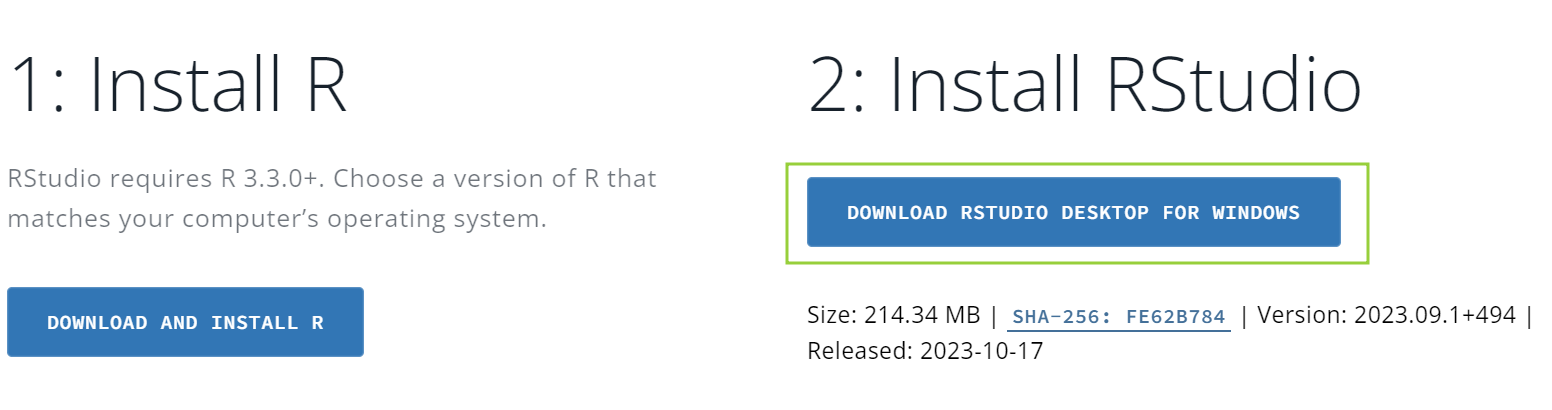


Em sequência, execute o instalador e siga as instruções para instalar o R na sua máquina.

Pronto! Após essa 1ª etapa, vamos para a instalação do R Studio.

#### Instalação do R Studio

Acesse o [site oficial do R Studio](https://posit.co/download/rstudio-desktop/). No início da página, são apresentados os dois passos necessários para utilizar o R Studio: instalar o R (que já fizemos) e instalar o R Studio.

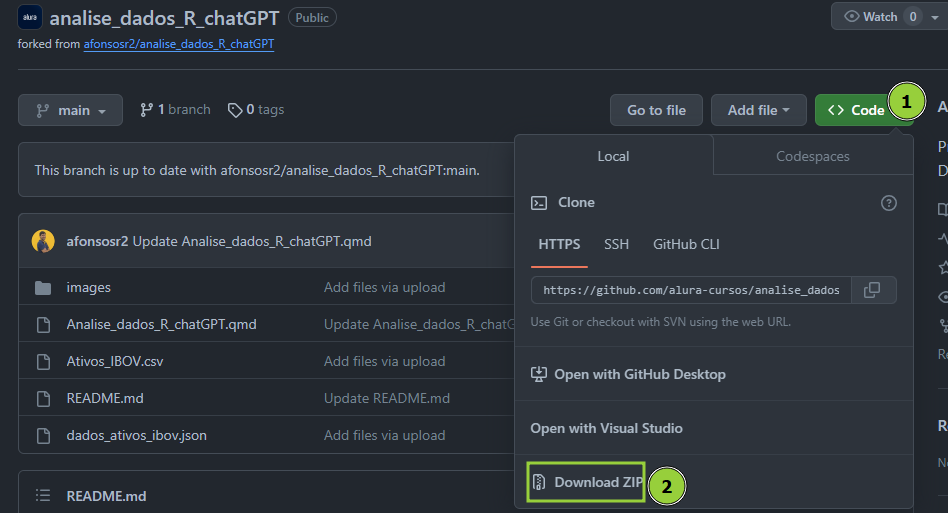


Vamos clicar em **Download Rstudio Desktop For Windows** para baixar o instalador no Windows. Note que ao descer mais na página do site oficial é possível observar diversos pacotes para diferentes SOs, caso seja o seu caso.

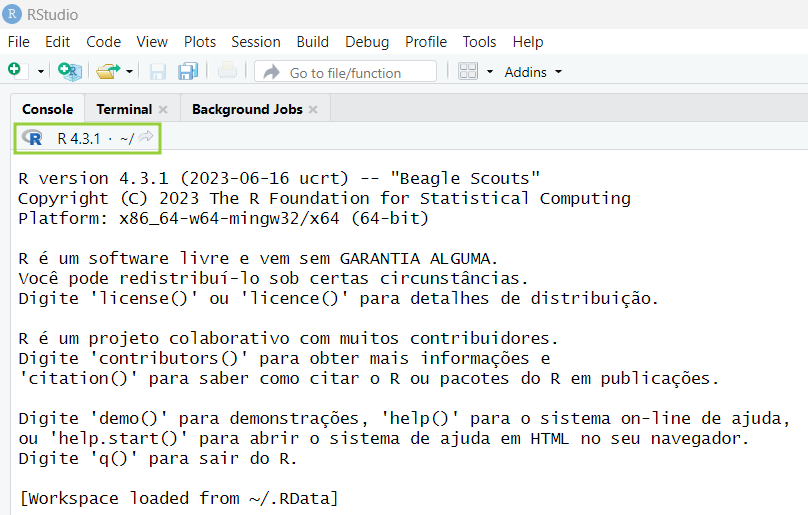
Em sequência, execute o instalador e siga as instruções para instalar o R Studio na sua máquina.

#### Importar o projeto

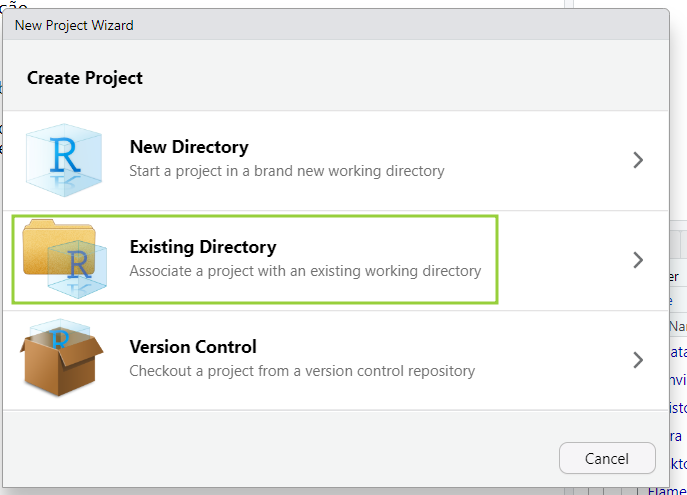
Após a instalação do R e do R Studio, vamos primeiro baixar o projeto do GitHub acessando este [link](https://github.com/alura-cursos/analise_dados_R_chatGPT). Para isto, vamos na opção **Code → Download ZIP** para baixar o projeto na máquina. Após baixado, descompacte a pasta e deixe no local que deseja para utilizar no projeto.



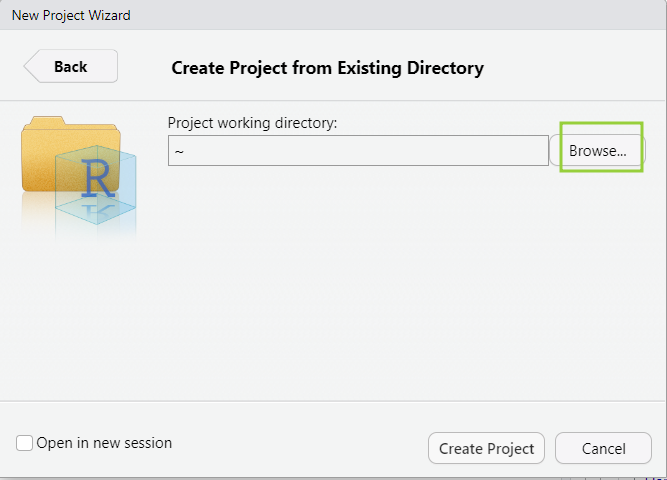
Agora, vamos abrir o RStudio que deve detectar automaticamente a instalação do R. Na parte superior esquerda do programa, você verá a versão do R que está sendo usada.



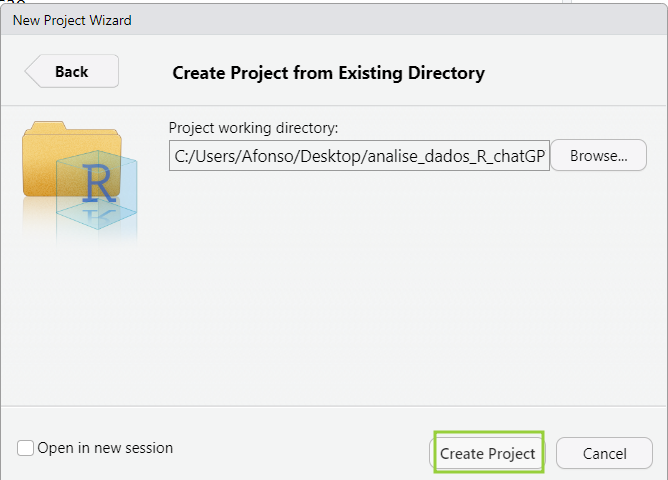
Vamos então criar o nosso projeto. Vá ao menu **File** (Arquivo) e selecione **New Project** (Novo Projeto). Em seguida, escolha a opção **Existing Directory** (Diretório Existente).



Este é o local onde seus scripts, dados e arquivos do projeto serão armazenados. Clique em **Browse** para escolher o nosso diretório de trabalho, ou seja, a nossa pasta raiz do projeto.



Navegue até a pasta que importamos os dados do GitHub e, por fim, clique em **Create Project**.



Pronto! Agora nosso projeto está configurado em nossa máquina e você pode executá-lo também seguindo todos os processos que aprendemos até aqui.

**Para saber mais: Quarto Notebook**

O [**Quarto Notebook**](https://www.quarto.org/) do R é uma ferramenta de autoria de documentos dinâmicos que combina texto formatado com código R e outros idiomas. Ele é baseado no R Markdown, mas oferece uma série de novos recursos e funcionalidades.

O Quarto Notebook do R pode ser usado para criar uma variedade de documentos, incluindo:

* Relatórios científicos
* Documentação técnica
* Apresentações
* Blogs
* Sites

Ele é especialmente útil para a criação de documentos que envolvem cálculos, análises de dados ou visualização de dados.

## Principais recursos do Quarto Notebook do R

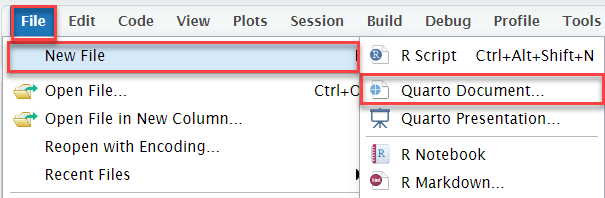
O Quarto Notebook do R oferece uma série de novos recursos e funcionalidades que o tornam uma ferramenta poderosa para a autoria de documentos dinâmicos. Alguns dos principais recursos incluem:

* **Suporte a múltiplos idiomas**: O Quarto Notebook do R suporta múltiplos idiomas, incluindo R, Python, Julia e JavaScript. Isso permite que você use a linguagem de programação que melhor se adapta às suas necessidades.
* **Suporte a notebooks**: O Quarto Notebook do R oferece suporte a notebooks, que são documentos que combinam texto formatado com código. Isso facilita a organização e a colaboração em projetos.
* **Suporte a gráficos interativos**: O Quarto Notebook do R oferece suporte a gráficos interativos, que permitem que os leitores explorem os dados de forma mais profunda.
* **Suporte a publicação**: O Quarto Notebook do R oferece suporte a uma variedade de formatos de publicação, incluindo PDF, HTML, Word e Markdown.

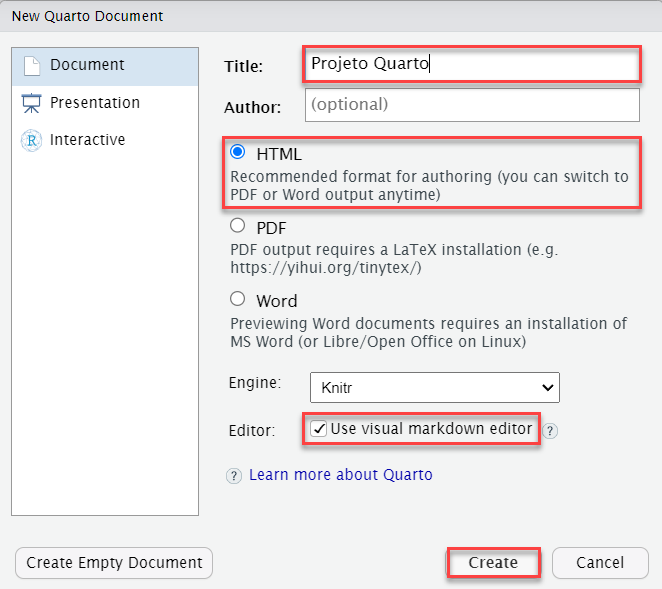
### Como usar o Quarto Notebook do R

O Quarto Notebook do R pode ser usado com uma variedade de editores de texto, incluindo RStudio, VS Code e JupyterLab.

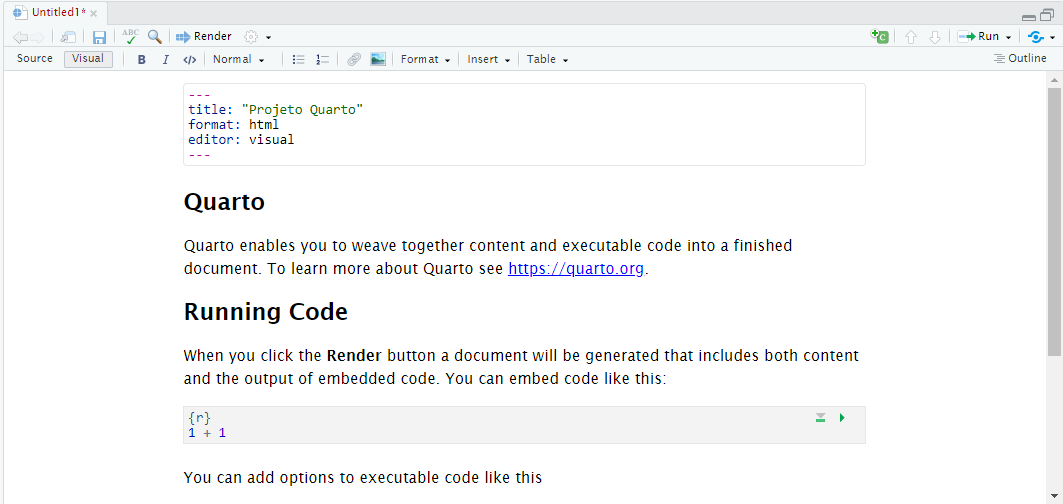
Você pode criar um novo notebook Quarto no Posit Cloud através do menu **File** no canto superior esquerdo. Nele, siga a opção **New File** → **Quarto Document**.



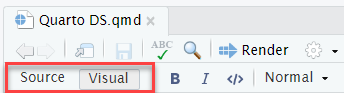
Em seguida, adicione o nome do seu documento e outras opções:



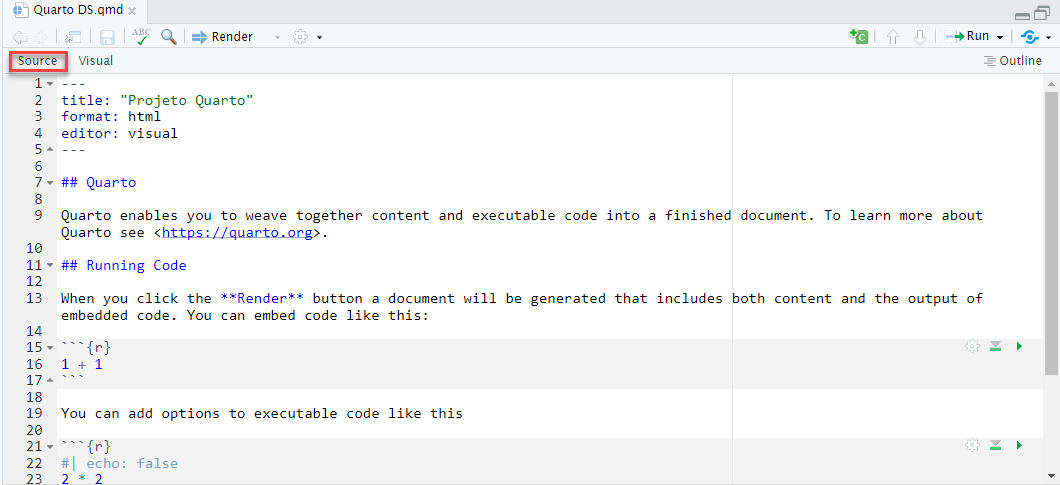
O documento Quarto inicial criado será o seguinte:



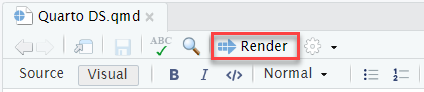
O notebook possui as opções de exibição e edição Visual, a qual já estava aberta por padrão, e a opção Source. Você pode trocar o modo no canto superior esquerdo, na barra de ferramentas do arquivo:



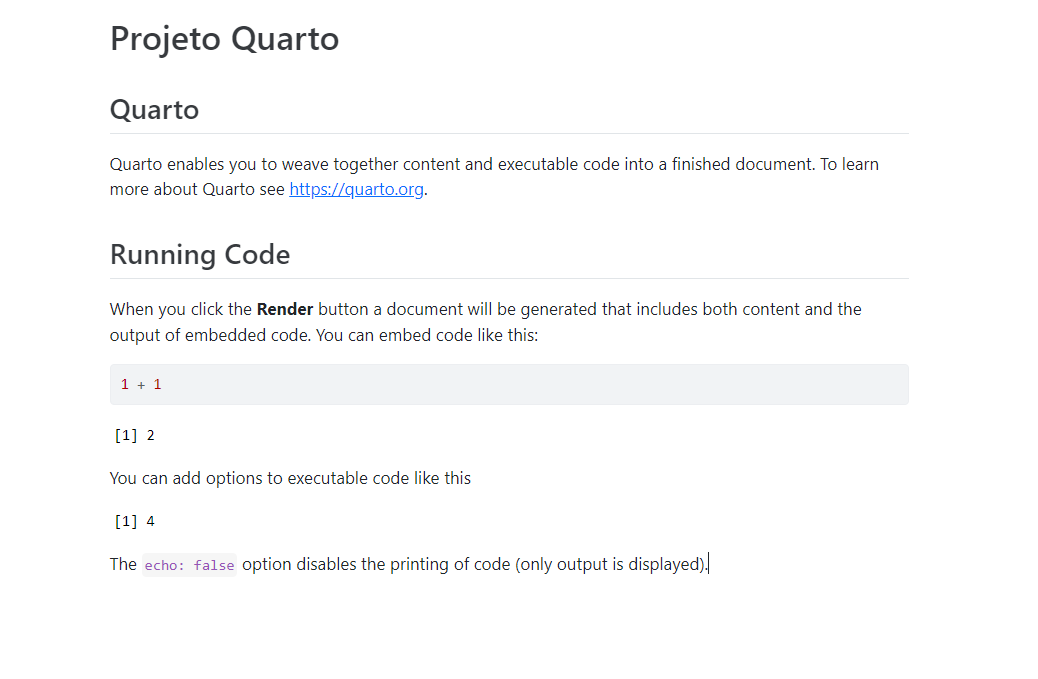
Enquanto a opção **Visual** permite que você tenha um arquivo visualmente agradável e possa formatar o texto através da barra de ferramentas, a opção **Source** fornece o controle total da formatação, mostrando todas as tags markdown:



Além das opções de edição do documento, você pode exibir o seu projeto em outra aba. Para renderizar um documento, clique no botão **Render** na barra de ferramentas do arquivo.



Como resultado, uma nova aba será aberta com o seu projeto renderizado:



O Quarto Notebook do R é uma ferramenta poderosa para a autoria de documentos dinâmicos. Ele oferece uma série de novos recursos e funcionalidades que o tornam uma escolha ideal para uma variedade de aplicações.

Com isso, você consegue dar os seus primeiros passos com o Quarto Notebook e começar a elaborar os seus projetos. Caso você queira explorar as funcionalidades do Quarto, acesse o capítulo [34 - Quarto is a Next-Generation RMarkdown](https://bookdown.org/pdr_higgins/rmrwr/quarto-is-a-next-generation-rmarkdown.html) do livro [Reproducible Medical Research with R](https://bookdown.org/pdr_higgins/rmrwr/" \t "_blank). O livro está em inglês, mas você pode utilizar o tradutor do seu navegador para ler em português.

**Para saber mais: tipos de dados**

Em linguagens de programação como o R, compreender os tipos de dados é essencial para manipular informações de maneira eficaz. Os tipos de dados definem a natureza das variáveis e influenciam as operações que podem ser realizadas sobre elas.

No R, há vários tipos de dados, cada um projetado para lidar com diferentes tipos de informações.

Vamos explorar alguns desses tipos de dados antes de discutir como verificar e entender esses tipos utilizando a função class().

* **Numérico (numeric/float):**
  + Este tipo de dado abrange números reais ou inteiros e é fundamental para operações matemáticas. Por exemplo, a variável x <- 5.2 é do tipo numérico.
* **Inteiro (integer):**
  + Semelhante ao tipo numérico, mas restrito a valores inteiros. O tipo integer é utilizado quando os valores podem ser apenas números inteiros, como y <- 10L (o sufixo "L" indica um inteiro).
* **Caracteres (character):**
  + Representa sequências de caracteres, como palavras ou frases. Variáveis desse tipo são frequentemente usadas para armazenar informações textuais, como z <- "Olá, mundo".
* **Lógico (logical):**
  + Variáveis lógicas representam valores verdadeiros ou falsos (TRUE ou FALSE). Essas são essenciais para expressar condições lógicas em operações e testes, como w <- TRUE.
* **Data (Date):**
  + Para lidar com informações de data, o tipo Date é utilizado. Isso permite operações específicas de data e hora. Por exemplo, data\_atual <- as.Date("2024-01-24").

A compreensão desses tipos de dados é crucial para escrever código eficiente e preciso em R. Agora, vamos explorar como podemos utilizar a função class() para identificar o tipo de dado associado a uma variável em nosso código.

nota <- 5.2

idade <- 20

frase <- "Olá, mundo"

cadastrado <- **TRUE**

data\_atual <- as.Date("2024-01-24")

# Verificando os tipos de dados

cat("nota é do tipo:", class(nota), "\n")

cat("idade é do tipo:", class(idade), "\n")

cat("frase é do tipo:", class(frase), "\n")

cat("cadastrado é do tipo:", class(cadastrado), "\n")

cat("data\_atual é do tipo:", class(data\_atual), "\n")

A função cat() exibe as informações de forma personalizada, combinando os elementos.

Compreender os tipos de dados em R é essencial para desenvolver análises estatísticas e manipulações eficientes. A diversidade de tipos, como numérico, inteiro, caractere, lógico e data, oferece flexibilidade na representação de informações. A função class() fornece uma visão rápida e informativa dos tipos de dados em R, facilitando o entendimento e a manipulação adequada das variáveis em seus projetos e análises.

**Para saber mais: exibindo com Print, Cat e Paste**

Quando estamos trabalhando com R, é comum simplesmente digitar o nome de uma variável ao final de uma célula de código para exibir seu conteúdo. Mas vamos dar uma olhada mais profunda nisso e entender como isso realmente funciona, com a ajuda das funções print() e cat(). Além disso, a função paste() pode ser uma ferramenta valiosa para personalizar a exibição de informações.

### Mostrando Dados do Jeito Padrão em R

Quando você digita o nome de uma variável e executa o código, o R usa automaticamente a função print() para exibir o conteúdo dessa variável. Essa abordagem é rápida e conveniente, mas às vezes queremos mais controle sobre como as coisas são apresentadas.

### print() - Mais Controle sobre a Exibição

A função print() oferece esse controle adicional. Vamos ver um exemplo:

# Vamos usar a função print()

idade <- 25

print("A idade é:")

print(idade)

Com print(), podemos adicionar uma mensagem personalizada, como "A idade é:", antes de mostrar o valor da variável. Isso dá um toque mais elaborado à exibição dos dados.

### cat() - Concatenando Informações

Agora, a função cat() nos permite juntar vários elementos em uma única saída. Veja um exemplo:

# Vamos usar a função cat()

nome <- "Maria"

idade <- 30

cat("Nome:", nome, "\n")

cat("Idade:", idade, "anos\n")

Com a função cat(), podemos criar uma saída personalizada com informações sobre o nome e a idade, cada uma em uma linha separada.

### paste() - Personalizando a Exibição

Além das funções mencionadas, a função paste() é uma ferramenta valiosa para personalizar a exibição de informações. Permite a combinação de strings e valores de variáveis, facilitando a criação de mensagens mais detalhadas. Um exemplo prático é a seguinte modificação no código original:

# Utilizando a função paste()

idade <- 25

paste("A idade é:", idade)

Com paste(), é possível criar mensagens mais elaboradas, incorporando variáveis diretamente na string.

Embora a exibição padrão em R seja conveniente, as funções print(), cat(), e paste() oferecem um controle mais refinado sobre a apresentação dos dados. Com essas ferramentas, podemos personalizar a exibição e tornar nossas saídas mais claras e informativas. É uma maneira simples de aprimorar a comunicação e a compreensão dos resultados em nossos projetos em R.