

# Análise dos Dados com R

Projetos em R

James R. Hunter, PhD  
Retrovirologia, EPM, UNIFESP

2024-11-05



# Semana Passada

- Falamos das variáveis e estruturas de variáveis
  - `vector`, `data.frame`, `tibble`

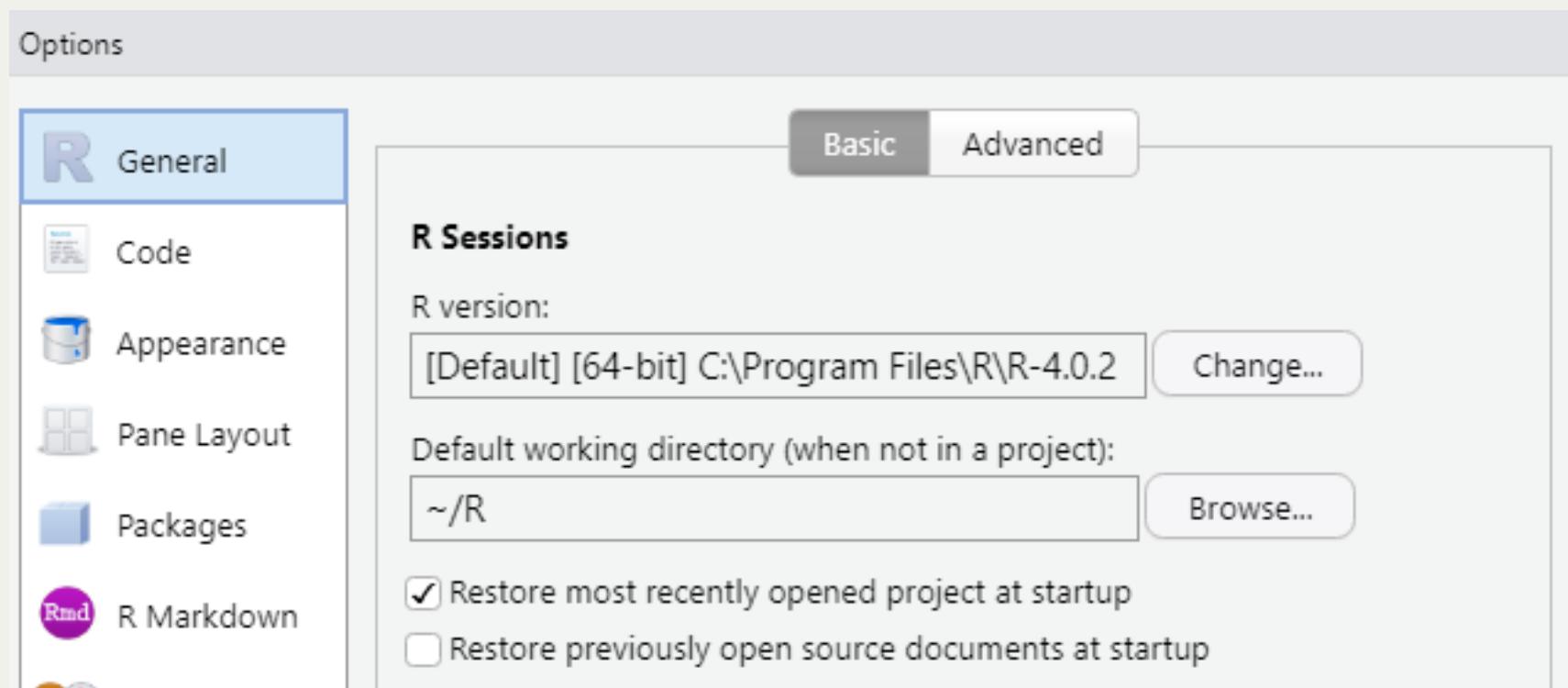
# Hoje

- Falamos de como usar esses objetos
  - Como combinar em *frases* que executam comandos e análises
  - Loops
  - Declarações condicionais (*if ... then ... else*)

# Projetos em R e RStudio

# Pastas Ativas

- Quando você inicia RStudio
- Programa usa a pasta que você definiu em “Tools/Global Options”



# setwd( ) & getwd( )

- Para criar uma nova pasta ativa (*working directory*), precisa usar o comando `setwd()`
  - Com o **nome completo do caminho** em aspas como argumento
- Exemplo - minha pesquisa da tese de doutorado sobre HIVAIDS
  - `setwd("C:/Users/james/OneDrive/Documents/HIVAIDS/Drug_Holiday")`
- Função complementária `getwd()`
  - Sem argumento
  - Mostra o que é a pasta ativa

```
1 getwd()
```

```
[1] "/Users/jameshunter/Documents/MAD/Bioinformatica Aplicada 2024/aulas_2024/class_notes"
```

# Porque Precisamos Projetos?

- Vai trabalhar em projetos diferentes
  - Bases de dados diferentes
  - *Scripts* (programas) diferentes
  - Documentos diferentes
- Jogando todos seus arquivos de todos os projetos em uma pasta R vai enloquecer você
  - Muitos projetos podem ter acima de 500 arquivos
- Planejamento do fluxo do projeto implica organização
- Construir um projeto de análise de dados em R baseado num espaço de trabalho distinto e reservado

# Projetos em RStudio – Trabalho Feito Melhor

- Tudo mundo tem uma estrutura diferente das pastas nos seus computadores
  - Tenho projetos gerais de computação na pasta R
  - Todos meus projetos sobre HIV ficam em minha pasta HIVAIDS
  - Todas as versões desta matéria ficam na minha pasta MAD
- O que é a estrutura das suas pastas?
- Projetos de R criam sua pasta ativa própria
- Se for na rede de trabalho seria a mesma estrutura para tudo mundo no grupo
- Não tem mais os caminhos de pasta quilométricos
- Que implica digitando eles errados

# The Tidyverse Warning/Threat!

A Equipe Tidyverse ameaça:

Se a primeira linha de um script de R é:

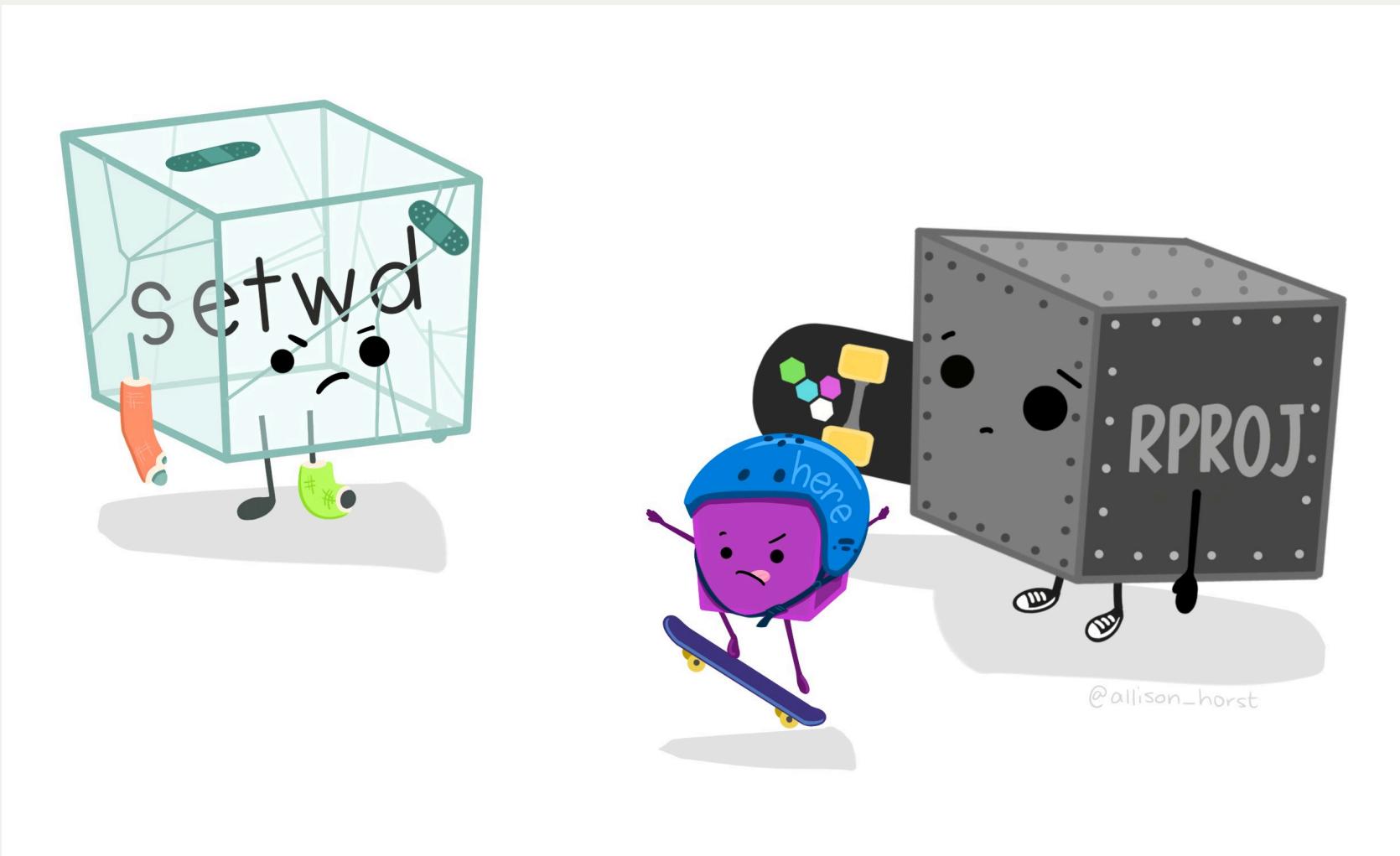
```
setwd("C:\Users\jenny\path\that\only\I\have")
```

Alguém da equipe virá para seu escritório e **incendiar o seu computador**

**Você foi avisado!!!**

A melhor maneira a evitar tal consequência é criar um novo projeto de RStudio para cada nova atividade.

# Projetos



Source: Alison Horst, <https://github.com/allisonhorst/stats->

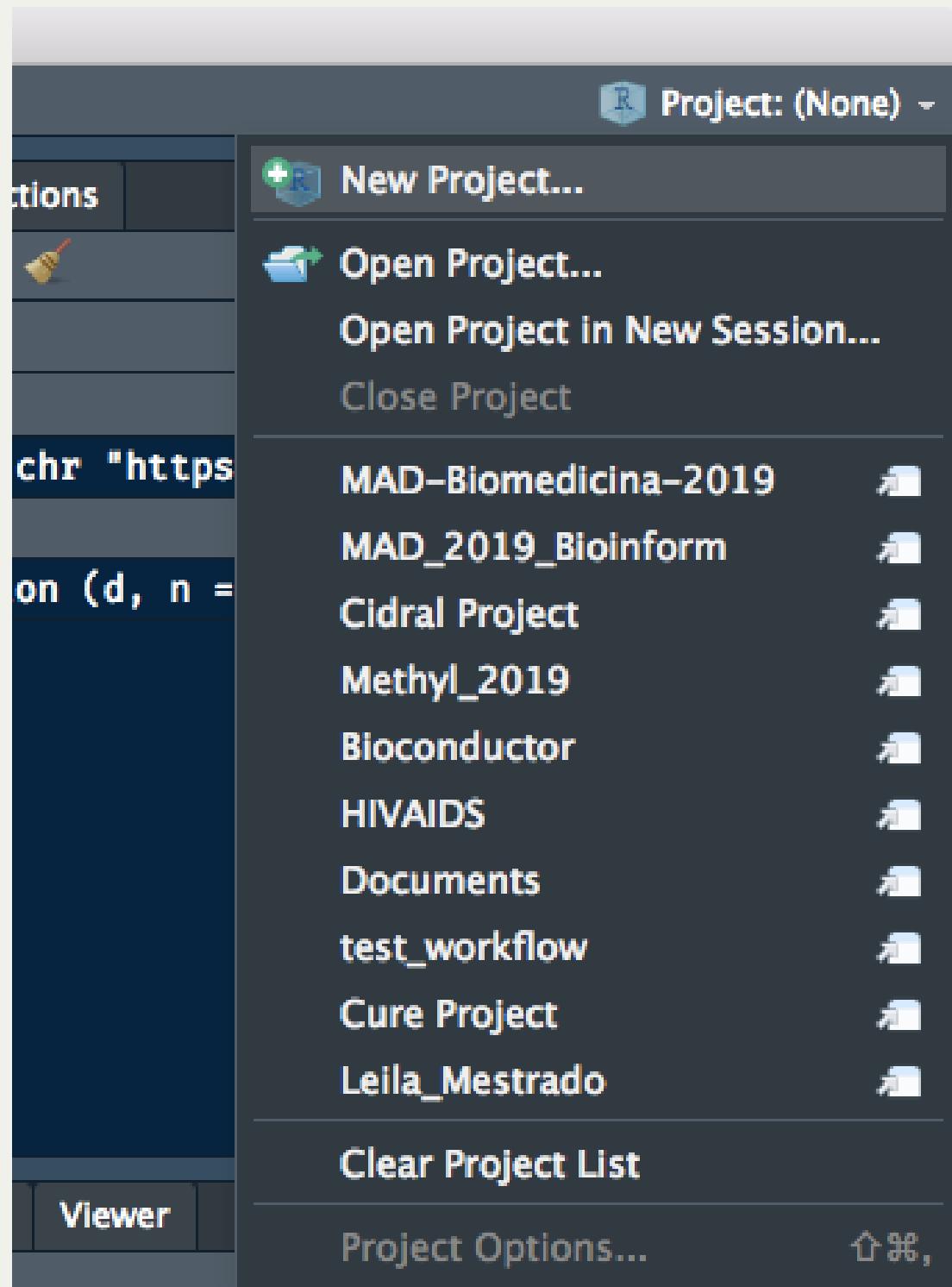
# Como Criar um Projeto de RStudio

- Procure o texto  
**Project: (None)**
- No topo da tela  
acima de aba de  
*Environment*
- Até agora, não tem um projeto **ativo**



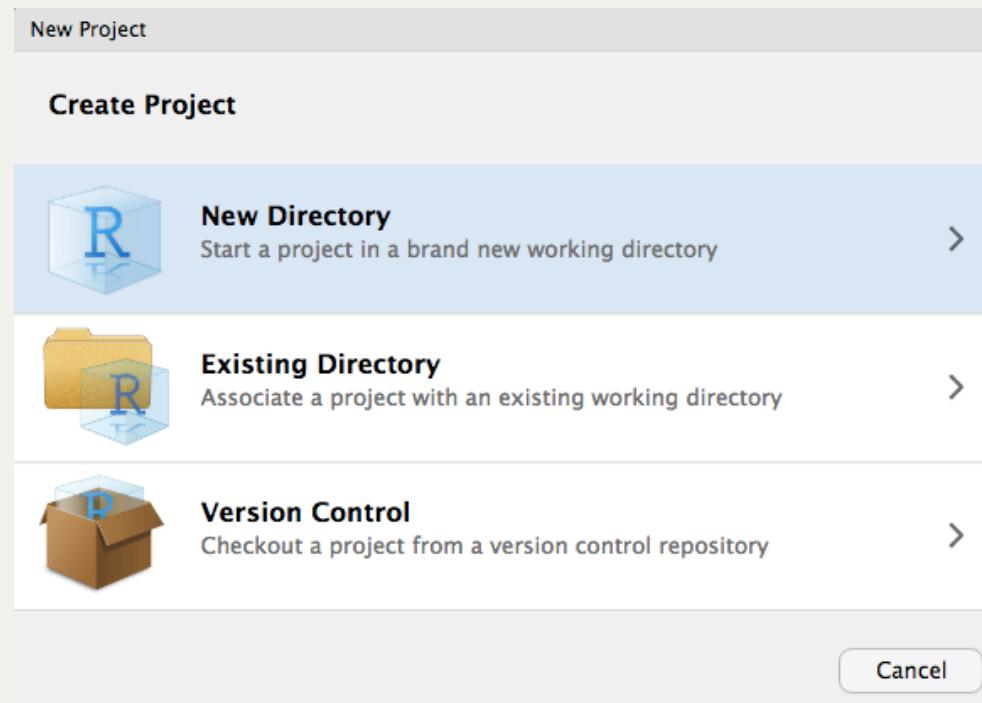
# Criar Novo Projeto

- Clique na seta a direta do “(None)”
- Menu suspenso abrirá
- Clique opção New Project ...



# Criar Janela do Projeto

- Opções para escolher uma pasta nova ou existente
- Opção **Version Control** mais avançada
- Neste caso, escolhe **New Directory**

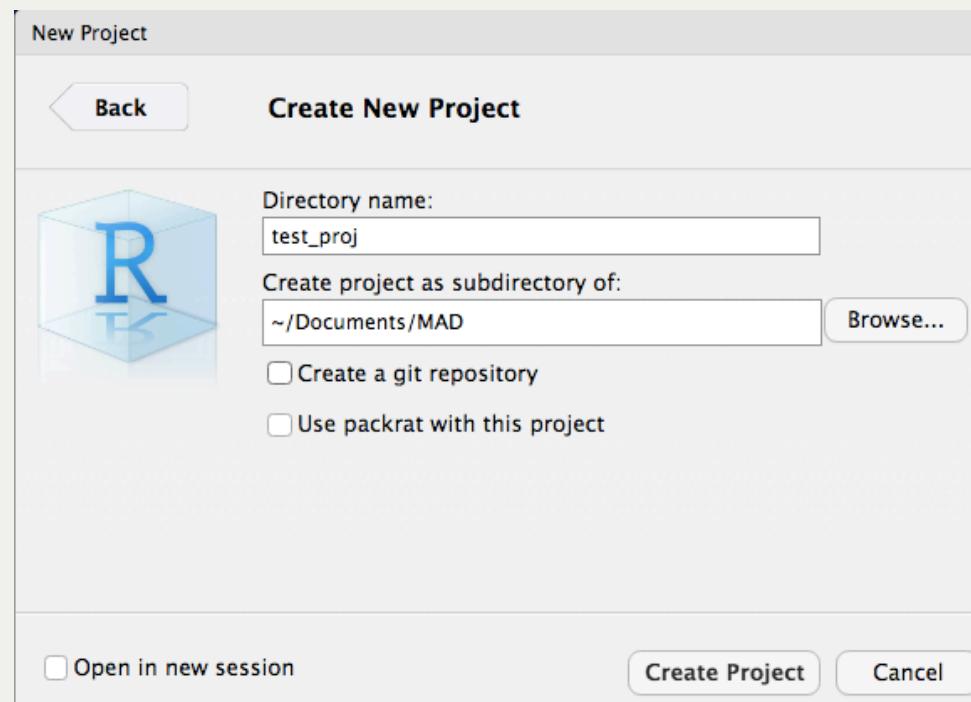


# Tipos de Projetos Novos

- RStudio tem muitos tipos de projetos
- Por enquanto, queremos um *New Project* simples: clique lá

# Tela de Criação do Projeto Final

- Dê um nome e um locale no disco
- Clique o botão **Create Project**
- Por enquanto, esqueça as caixas de **git** e **renv**



# Eis Seu Projeto

The screenshot shows the RStudio interface with the following details:

- Console Pane:** Displays the R startup message and the command prompt > |.
- Environment Pane:** Shows the global environment with two entries:
  - r: Named chr "https://cran.r-project.org"
  - ht: function (d, n = 6)
- Files Pane:** Shows the project structure:
  - New Folder
  - Delete
  - Rename
  - More
  - Home > Documents > MAD > test\_proj
  - test\_proj.Rproj (205 B, Aug 1, 2019, 10:46 AM)

# Sub-Pastas

- Projetos podem tornar muito grande, **MUITO GRANDE**
  - Projeto de tese project: 1.978 arquivos, 383 sub-pastas, 425 MB
- Sub-pastas para arquivos de tipos diferentes
  - Parece a criação de um projeto
  - Ajuda organizar os arquivos
- Sempre guardar dados brutos longe de qualquer outro tipo de arquivo de dados
  - Tratados, sub-conjuntos, etc.
  - Você vai precisar eles de novo!

# Minha Estrutura das Pastas Preferida

- Sub-pastas
  - Dados
  - Dados brutos
  - Documentos
  - Gráficos
  - Programas/Scripts
  - Slides

# Here



Source: Alison Horst, <https://github.com/allisonhorst/stats->

# Problema

- Vários projetos e várias pastas podem ter caminhos (*paths*) muito diferentes
- Caminho para chegar no `test_proj` no meu velho laptop Windows:
  - `C:/Users/james/OneDrive/Documents/MAD/test_proj`
- Caminho para chegar no `test_proj` no meu novo Mac Air:
  - `/Users/jameshunter/Documents/MAD/test_proj`
- Se eu quis utilizar meu caminho para ir para um sub-pasta `graphics` nas duas maquinas
  - **Seria um desastre**

# Solução: `here::here()`

- Pacote `here` faz uma coisa só
  - Mostra para você o caminho completo para sua pasta ativa
  - Numa forma simples: não tem argumentos

```
1 here::here()
```

```
[1] "/Users/jameshunter/Documents/MAD/Bioinformatica Aplicada 2024/aulas_2024"
```

# Ir à Sub-Pasta **graphics** do Projeto

- Give the function the argument "graphics"
- Thereafter, can use variable **gr** to refer to full path of location

```
1 gr <- here::here("graphics")
2 gr
[1] "/Users/jameshunter/Documents/MAD/Bioinformatica Aplicada 2024/aulas_2024/graphics"
```

# Fine Parte 1