

# Aula 1

Renato Rodrigues Silva

Universidade Federal de Goiás

( 2020-12-15)

# Bem-vindos!!!

- O curso, Introdução ao software R: Aplicações na área da saúde, objetiva capacitar os funcionários e gestores da Secretaria da Saúde do Estado de Goiás para que eles possam desenvolver melhor as suas atividades.
- O curso também será aberto para outros profissionais liberais da área da saúde, incluindo alguns docentes, discentes e pesquisadores da Escola de Medicina da Universidade Federal de Goiás.

# Programação:

Dia 1:

- Introdução ao GitHub / Comentários sobre sistemas operacionais;
- Introdução ao software R;
- Introdução ao ambiente de desenvolvimento integrado RStudio.

Dia 2:

- Manipulação / Visualização de dados.
- Medidas Resumo com software R.

Obs: O cronograma pode mudar de acordo com a necessidades dos participantes.

Dia 3:

- Intervalos de Confiança
- Testes de Comparação de Médias
- Análise de Variância com 1 fator
- Análise de Correlação
- Testes Não Paramétricos para comparar dois grupos
- Testes Não Paramétricos para comparar mais de dois grupos

Dia 4:

- Estudo de Caso

# Introdução ao GitHub

## Git

- Git é um sistema de **controle de versão** desenvolvido por Linus Torvalds (HOSTINGER, 2015?).

## Controle de versão

- Controle de versão permite o acompanhamento de mudanças feitas no código-fonte.
- Também registra quem efetuou a mudança e possibilita a restauração do código removido ou modificado.
- Não há códigos sobreescritos uma vez que o Git salva múltiplas cópias no repositório (HOSTINGER, 2015?).

# Introdução ao GitHub

## Git versus GitHub

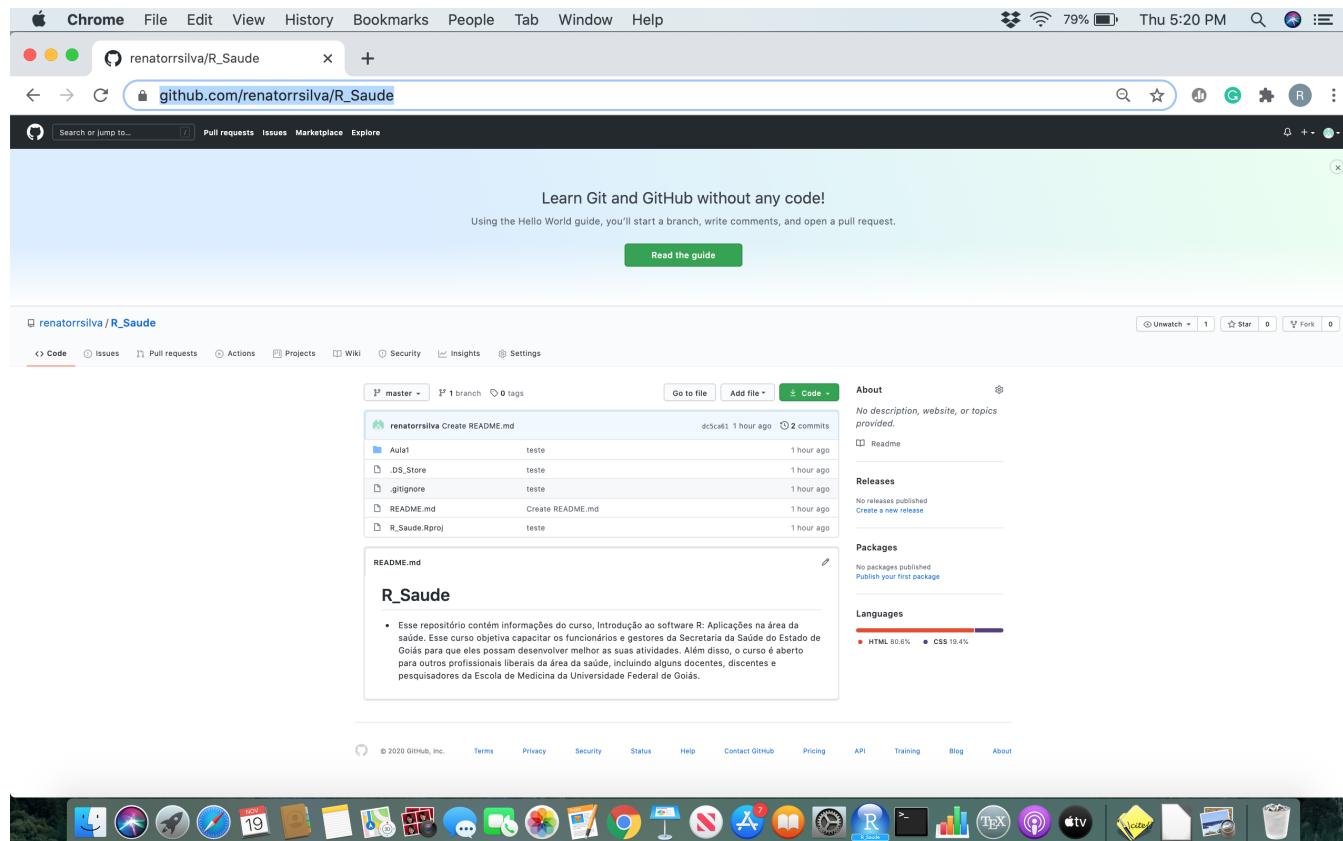
- Simplificando, Git é um sistema de controle de versão que permite gerenciar e controlar o histórico do código-fonte (DEVMOUNTAIN BLOG, 2015?)
- GitHub é um serviço de hospedagem baseado em nuvem que permite gerenciar repositórios Git. Se você tiver projetos de código aberto que usam Git, o GitHub foi projetado para ajudá-lo a gerenciá-los melhor (DEVMOUNTAIN BLOG, 2015?).

## Informações Adicionais

- Nesse curso, vamos focar em ensinar como baixar arquivos a partir do GitHub.
- Maiores detalhes de como usar o GitHub foge do escopo desse curso, mas quem tiver interesse é só [clicar aqui](#).

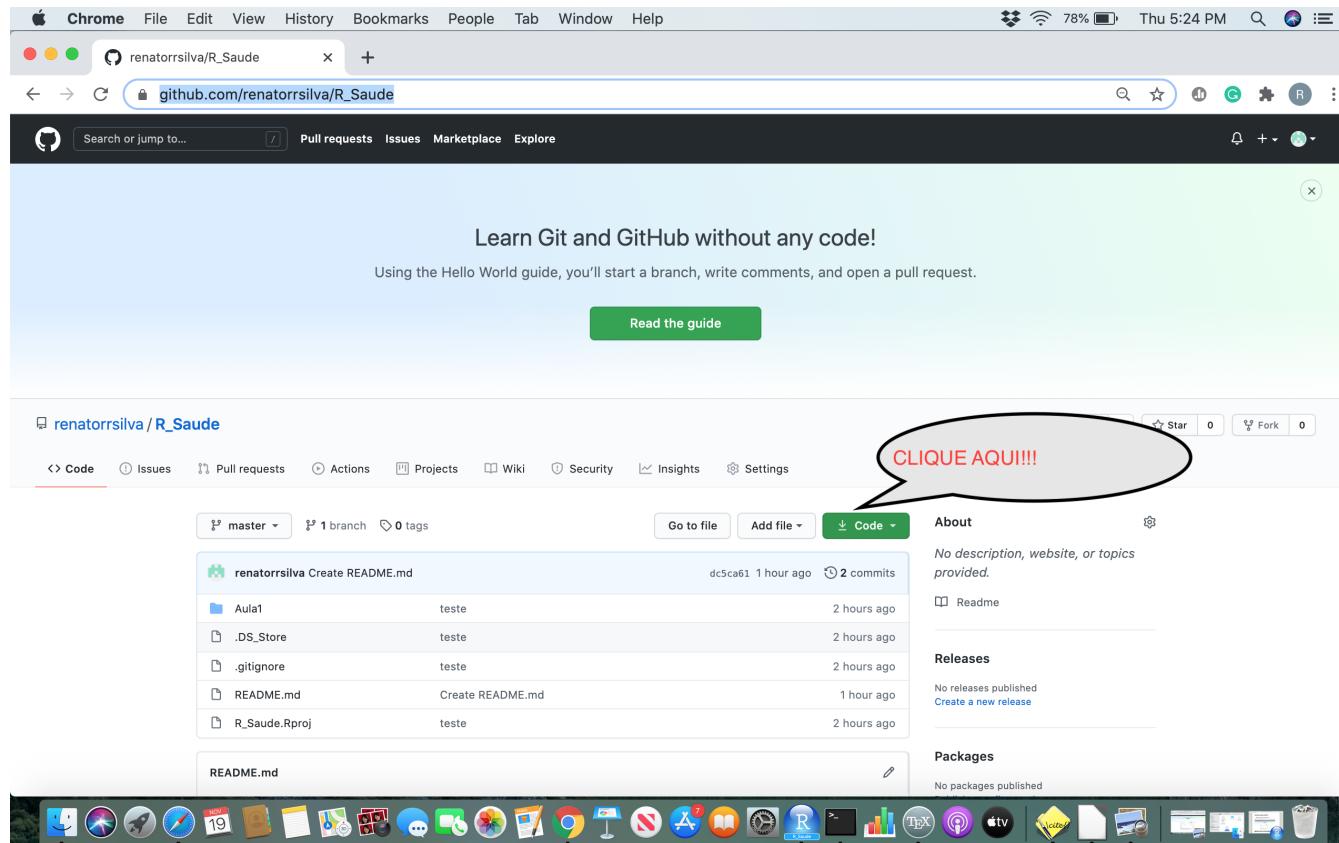
# Como baixar arquivos do GitHub

- Digite no seu web browser o seguinte endereço:  
[https://github.com/renatorrsilva/R\\_Saude](https://github.com/renatorrsilva/R_Saude):



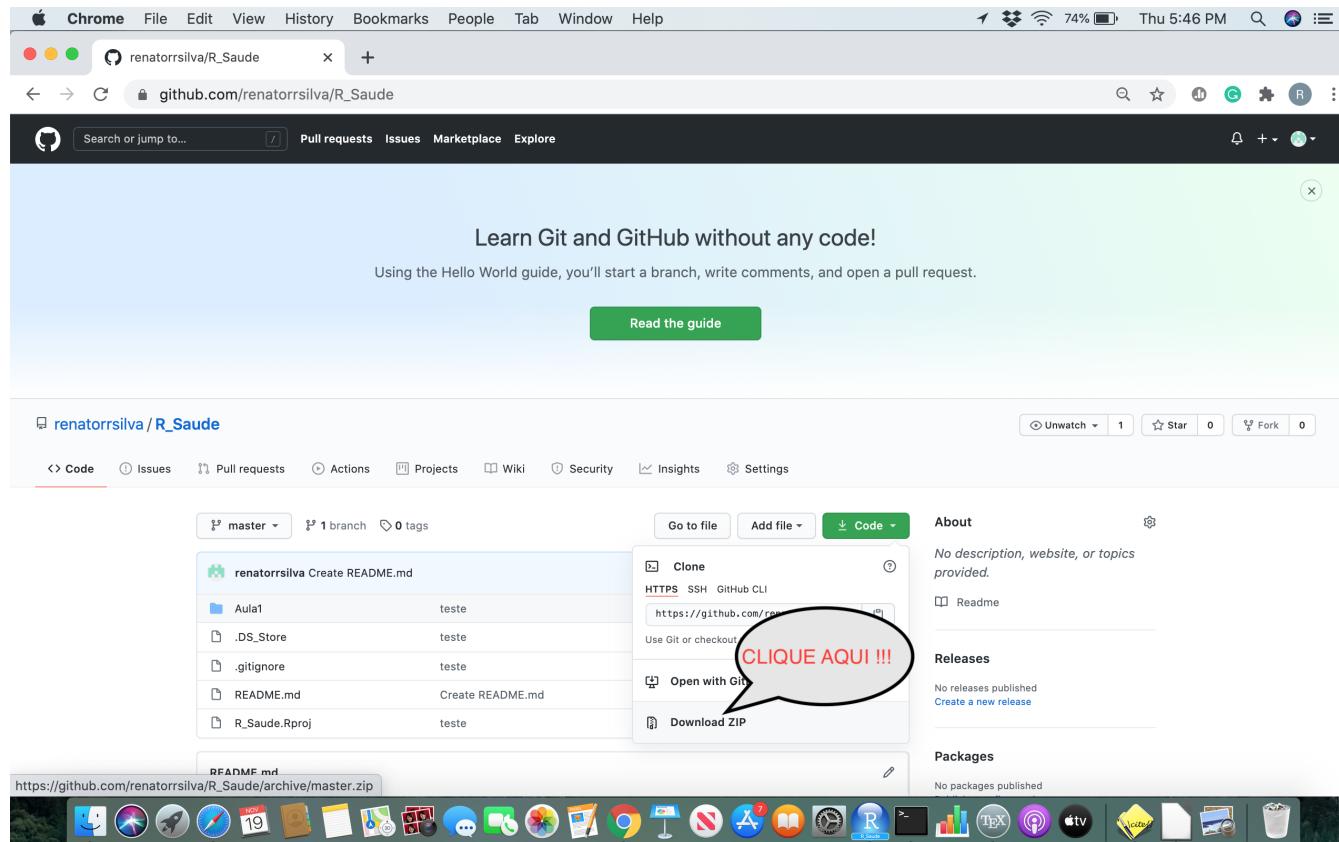
# Como baixar arquivos do GitHub

- Clique na janela Code em verde



# Como baixar arquivos do GitHub

- Clique no botão Downloads.ZIP



# Breve resumo sobre sistemas operacionais

## O que é sistema operacional

- "É um software ou conjunto de softwares cuja função é administrar os recursos de um sistema, estabelecendo a **interface entre o computador e o usuário**" (TECNOBLOG, 2020?).
- Como exemplo temos: Microsoft Windows, Mac OS X e Linux.
- Veremos que ao longo do curso, alguns aspectos relacionados a instalação de softwares pode ser diferente para cada sistema operacional. No entanto, as linhas de comando do software R são as mesmas, independendo do sistema operacional a ser usado.
- Maiores detalhes sobre sistema operacional, [clique aqui](#) (GCF APRENDE LIVRE, 2020?).

# Breve resumo sobre sistemas operacionais

## Caminho

Um caminho, do inglês path, é a forma geral do nome de um arquivo ou diretório, especifica uma localização única em um sistema de arquivos (WIKIPEDIA, 2000?).

Exemplos:

- Nome do diretório onde está localizado o arquivo dessa aula.

/Users/renatorodriguessilva/Dropbox/UFG/extensao/R\_Saude/Aula1.

# Caminho absoluto

- Um caminho absoluto aponta para a mesma localização em um sistema de arquivos, independentemente do diretório de trabalho atual (WIKIPEDIA, 2000?). Para fazer isso, é necessário incluir o diretório raiz (WIKIPEDIA, 2000?).

# Caminho relativo

- Um caminho relativo inicia a partir de algum diretório de trabalho fornecido, evitando a necessidade de fornecer o caminho absoluto completo (WIKIPEDIA, 2000?).
- Um nome de arquivo pode ser considerado como um caminho relativo baseado no diretório de trabalho atual (WIKIPEDIA, 2000?).

## IMPORTANTE !!!

- Se o diretório de trabalho não for o diretório pai do arquivo (onde o arquivo está localizado), resultará em um erro de arquivo não encontrado se o arquivo for endereçado pelo seu nome (WIKIPEDIA, 2000?).

# Breve resumo sobre sistemas operacionais

## Caminhos no Windows (DOS tradicional)

Um caminho DOS é composto de:

- Volume ou letra da unidade seguido pelo separador : (dois pontos),
- Um nome de diretório, o caractere separador de diretório \(\backslash\backslash\) (barra), separa subdiretórios dentro de uma hierarquia aninhada de diretório.
- Um nome de arquivo opcional. o caractere separador de diretório \(\backslash\backslash\) (barra) separa o caminho do arquivo e o nome do arquivo.

# Breve resumo sobre sistemas operacionais

Exemplos:

a. C:\Documents\Newsletters\Summer2018.pdf,

- Esse é um caminho de arquivo absoluto da raiz da unidade C:

b. 2018\January.xlsx

- Um caminho relativo para um arquivo em um subdiretório do diretório atual.

## IMPORTANTE !!!

- Caminhos declarados de forma incorreta é uma das maiores fontes de erros durante o aprendizado de softwares estatísticos, principalmente no que tange a leitura dos dados.
- Minha dica é: Aponte o software para um diretório, coloque seus arquivos neste diretório e sempre use apenas o nome do arquivo como caminho relativo.
- Para achar caminho de um arquivo no Windows, [clique aqui](#).

# Breve resumo sobre sistemas operacionais

## Caminhos no Linux / Mac OS X

- Caminho absoluto: Todo caminho absoluto inicia no diretório raiz (/), por exemplo: /home/aluno, a partir do qual, todos caminhos absolutos derivam, formando uma árvore de diretórios.
- A vantagem dos caminhos absolutos é poder identificar arquivos.

# Breve resumo sobre sistemas operacionais

## Caminhos no Linux / Mac OS X

- Caminho relativo: Todo caminho relativo não contém uma 'barra' no início. A referência é geralmente o diretório atual (pwd) onde o shell se encontra .
- Em caminhos relativos é possível utilizar outros atalhos como:
- ~ til, que referenciam o diretório pessoal. Por exemplo, ls ~/Downloads lista a pasta Downloads da pasta pessoal do usuário atual.
- . ponto, que significa o diretório atual. Por exemplo, ./run-app, executa o arquivo runapp que está localizado no diretório atual.
- .. dois-pontos, que significa o diretório pai. Por exemplo, cd .., muda para o diretório pai.
- Maiores detalhes, [clique aqui](#) (FONTANA & CASTRO, 2014).

# Breve resumo sobre sistemas operacionais

## Caminhos no Linux / Mac OS X

### Exemplos

/home/bohlke/programas/testes/arquivo1.blah

é um caminho absoluto para o arquivo. Pode-se alcançar o arquivo1.blah independendo do diretório atual.

programas/teste/arquivo1.blah

é um caminho relativo para o mesmo arquivo. Só se pode alcançá-lo se estivermos previamente em /home/bohlke.

Maiores informações sobre uso do terminal do Linux / Mac OS X [clique aqui](#).

# Introdução ao software R

## O que é R?

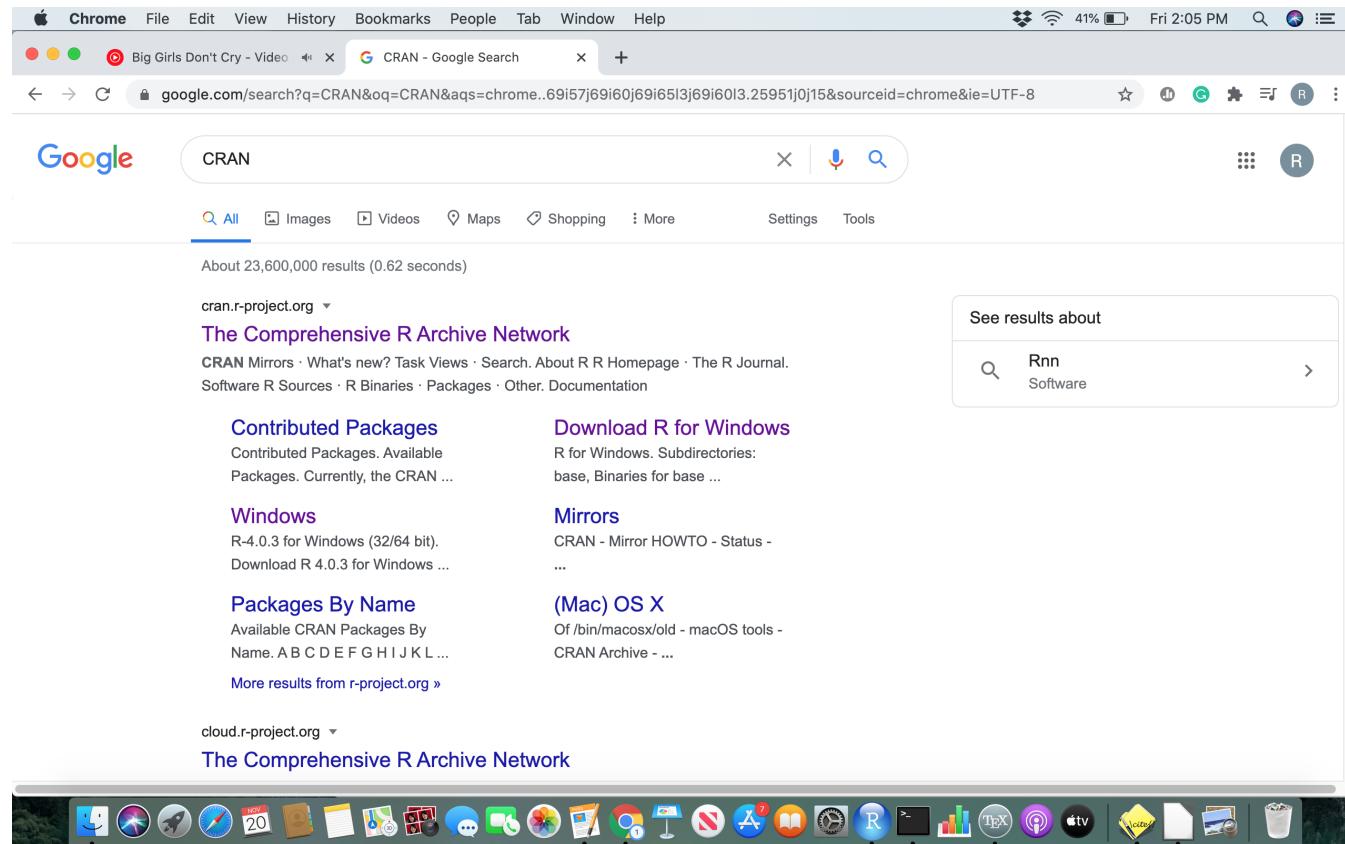
- R é um ambiente de software livre para computação estatística e gráficos. Compila e roda em uma ampla variedade de plataformas UNIX, Windows e MacOS (HORNIK 2020).

## O que é CRAN?

- A “Rede Comprehensive R Archive” (CRAN) é uma coleção de sites que carregam material idêntico, consistindo na (s) distribuição (ões) R, as extensões contribuídas, documentação para R e arquivos binários (HORNIK 2020).

# Instalação do software R no Windows

a. Digite no google a palavra CRAN



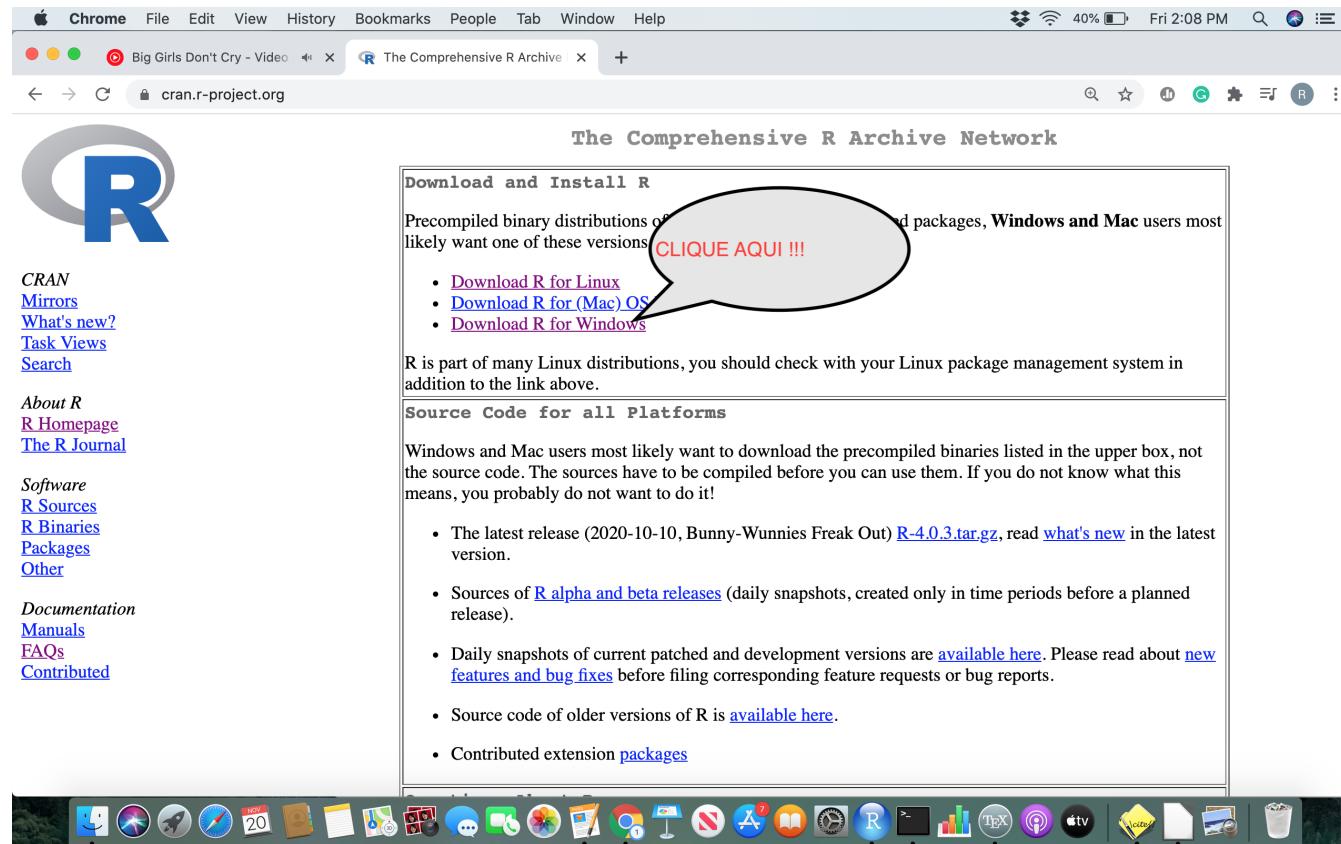
# Instalação do software R no Windows

b. Entre na página <https://cran.r-project.org/>

A screenshot of a Google search results page for "CRAN" in a Chrome browser. The search bar shows "Google CRAN". Below the search bar, there are filters for "All", "Images", "Videos", and "Maps". The search results include a snippet: "About 23,600,000 results (0.62 seconds)". A red speech bubble with the text "CLIQUE AQUI !!!" points to the first result, which is a link to "cran.r-project.org". The link is followed by the text "The Comprehensive R Archive Network". Below the link, there are several categories: "Contributed Packages", "Windows", "Packages By Name", and "Download R for Windows", "Mirrors", and "(Mac) OS X". On the right side, there is a sidebar titled "See results about" with a search bar containing "Rnn Software". The browser's toolbar at the bottom includes icons for various applications like Mail, Calendar, and Finder.

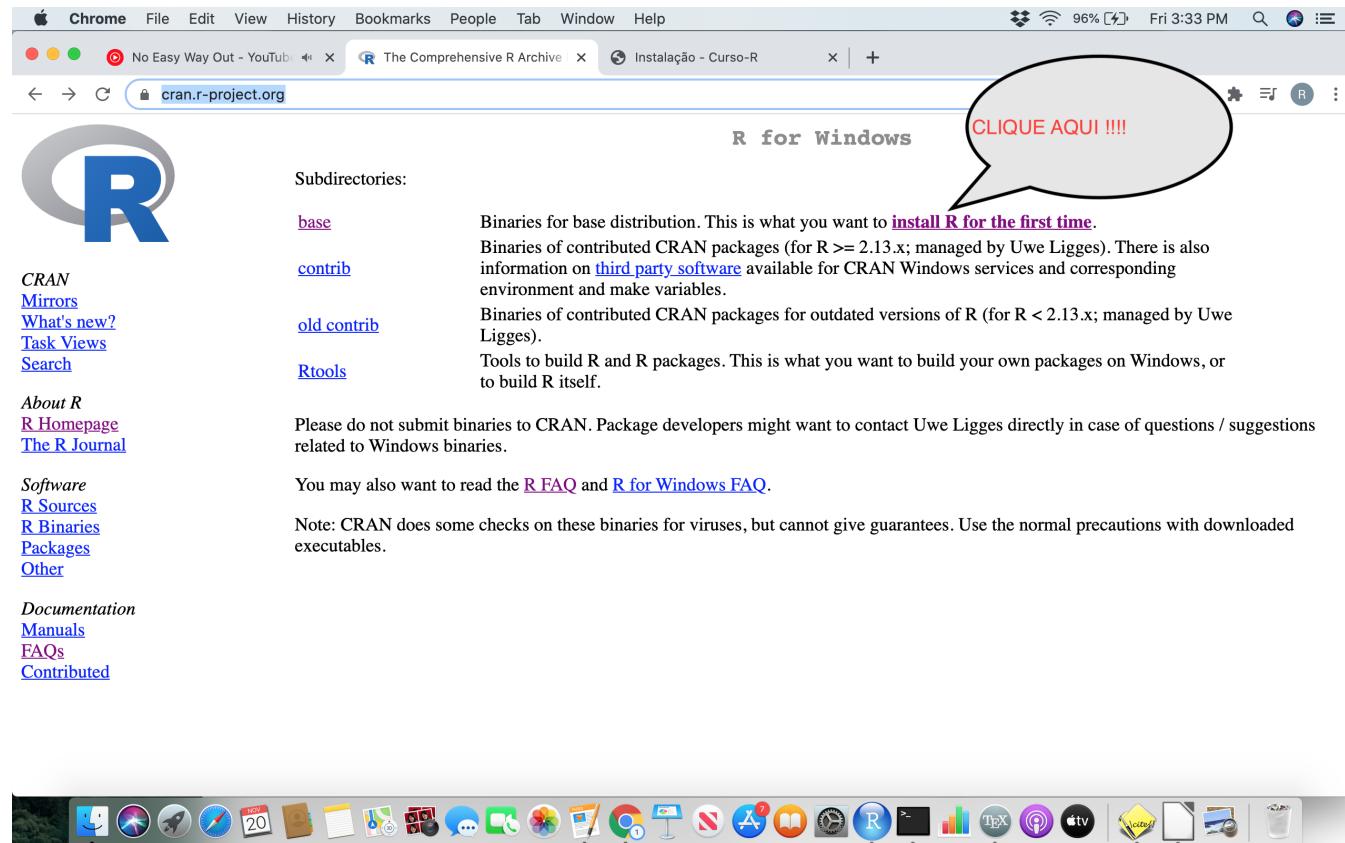
# Instalação do software R no Windows

## c. Clique em Download R for Windows



# Instalação do software R no Windows

## d. Clique em install R for the first time



# Instalação do software R no Windows

e. Clique em install Download R.4.0.3 for Windows e de

The screenshot shows a Chrome browser window with three tabs open:

- YouTube Music
- The Comprehensive R Archive
- Instalação - Curso-R

The main content area displays the CRAN website for R 4.0.3. A large blue "R" logo is on the left. The central text reads "Windows (32/64 bit)" with a red speech bubble containing "CLIQUE AQUI !!!" pointing to the "Download R 4.0.3 for Windows" link. Below this, there are links for "Installation and other instructions" and "New features in this version". A note about md5sum fingerprinting is present. A "Frequently asked questions" section lists links for running under Windows, updating packages, and choosing 32-bit vs 64-bit. A note for webmasters provides a stable redirect link. The bottom of the page shows a Mac OS X-style dock with various application icons.

# Instalação do software R no Windows

- Clique no arquivo duas vezes com o botão esquerdo. Ele pedirá para você selecionar a linguagem da instalação. Escolha um idioma e clique em “OK (CURSO R, 2018?)”.
- Em seguida, clique em “Avançar” até chegar na tela da imagem abaixo. Nessa etapa, você precisará escolher a pasta de instalação (CURSO R, 2018?).
- Se você escolher um local que não está dentro da pasta do seu usuário, você precisará de acesso de administrador. Se escolher uma pasta dentro do seu usuário (como na imagem abaixo), não precisará (CURSO R, 2018?).
- Continue clicando em “Avançar” e, ao fim da instalação, em “Concluir”.

# Instalação do software R no Mac OS X

a. Clique no arquivo que contém a versão mais recente do R em “Arquivos”.

The screenshot shows a Mac OS X desktop with a browser window open to the CRAN R package download page. The URL is [cran.r-project.org](https://cran.r-project.org). The page displays information about the latest release of R (4.0.3). A red speech bubble highlights the download link for the R-4.0.3.pkg file, which is described as being notarized and signed. The file size is approximately 8.5MB. The page also includes links for documentation, manuals, FAQs, and contributed packages, as well as news about the R.app Mac GUI and sources for the R.app GUI 1.73.

Please check the MD5 checksum of the downloaded image to ensure that it has not been tampered with or corrupted during the mirroring process. For example type  
openssl sha1  
in the Terminal and copy the output. Then compare it with the SHA1 checksum for the R-4.0.3.pkg image. On Mac OS X 10.7 and later you can also validate the signature with the command  
pkutil verifypkg R-4.0.3.pkg

**CLIQUE AQUI !!!**

**R 4.0.3 "Bunny-Wunnies Freak Out" released on 2020/10/10**

**Latest release:**

**R-4.0.3.pkg** (notarized and signed)  
SHA1:  
hash: 8402f586aeff1fdb12c6e34c73b286f87318fb1be  
(ca. 8.5MB)

**R 4.0.3** binary for macOS 10.13 (High Sierra) and higher, signed and notarized package. Contains R 4.0.3 framework, R.app GUI 1.73 in 64-bit for Intel Macs, Tcl/Tk 8.6.6 X11 libraries and Texinfo 6.7. The latter two components are optional and can be omitted when choosing "custom install", they are only needed if you want to use the `tcltk` R package or build package documentation from sources.

Note: the use of X11 (including `tcltk`) requires [XQuartz](#) to be installed since it is no longer part of OS X. Always re-install XQuartz when upgrading your macOS to a new major version.

**Important:** this release uses Xcode 10.1 and GNU Fortran 8.2. If you wish to compile R packages from sources, you will need to download and GNU Fortran 8.2 - see the [tools](#) directory.

**NEWS** (for Mac GUI)

News features and changes in the R.app Mac GUI

**Mac-GUI-1.73.tar.gz**  
SHA1:  
hash: 7f4b1d050757ce78545bdeb9d178a69d13046aa1

Sources for the R.app GUI 1.73 for Mac OS X. This file is only needed if you want to join the development of the GUI, it is not intended for regular users. Read the INSTALL file for further instructions.

Note: Previous R versions for El Capitan can be found in the [el-capitan/base](#) directory.

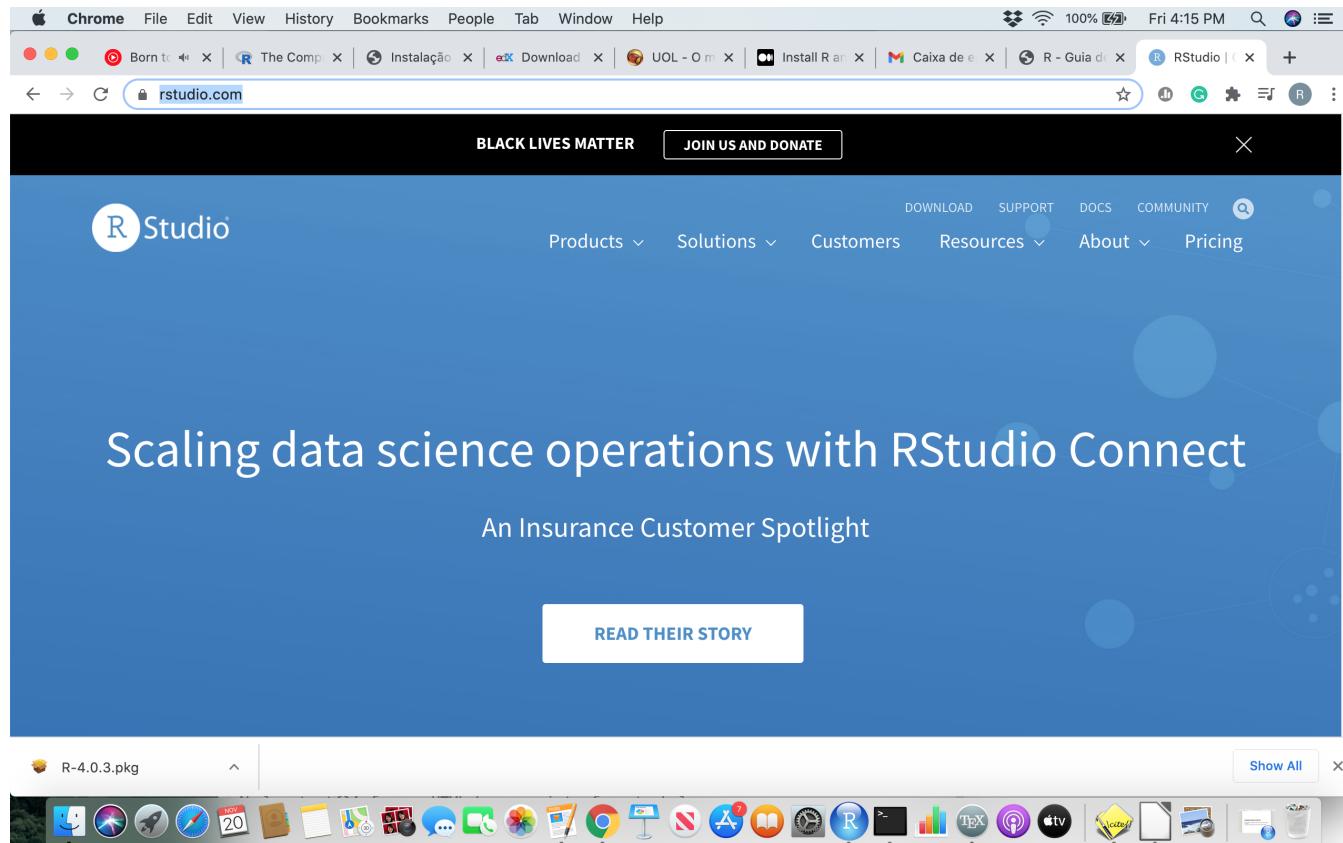
At the bottom, the Mac OS X Dock is visible with various application icons.

# Instalação do software R no Mac OS X

- b. Salve o arquivo .pkg, clique duas vezes nele para abri-lo e siga as instruções de instalação.
- c. Agora que o R está instalado, você precisa baixar e instalar o RStudio.

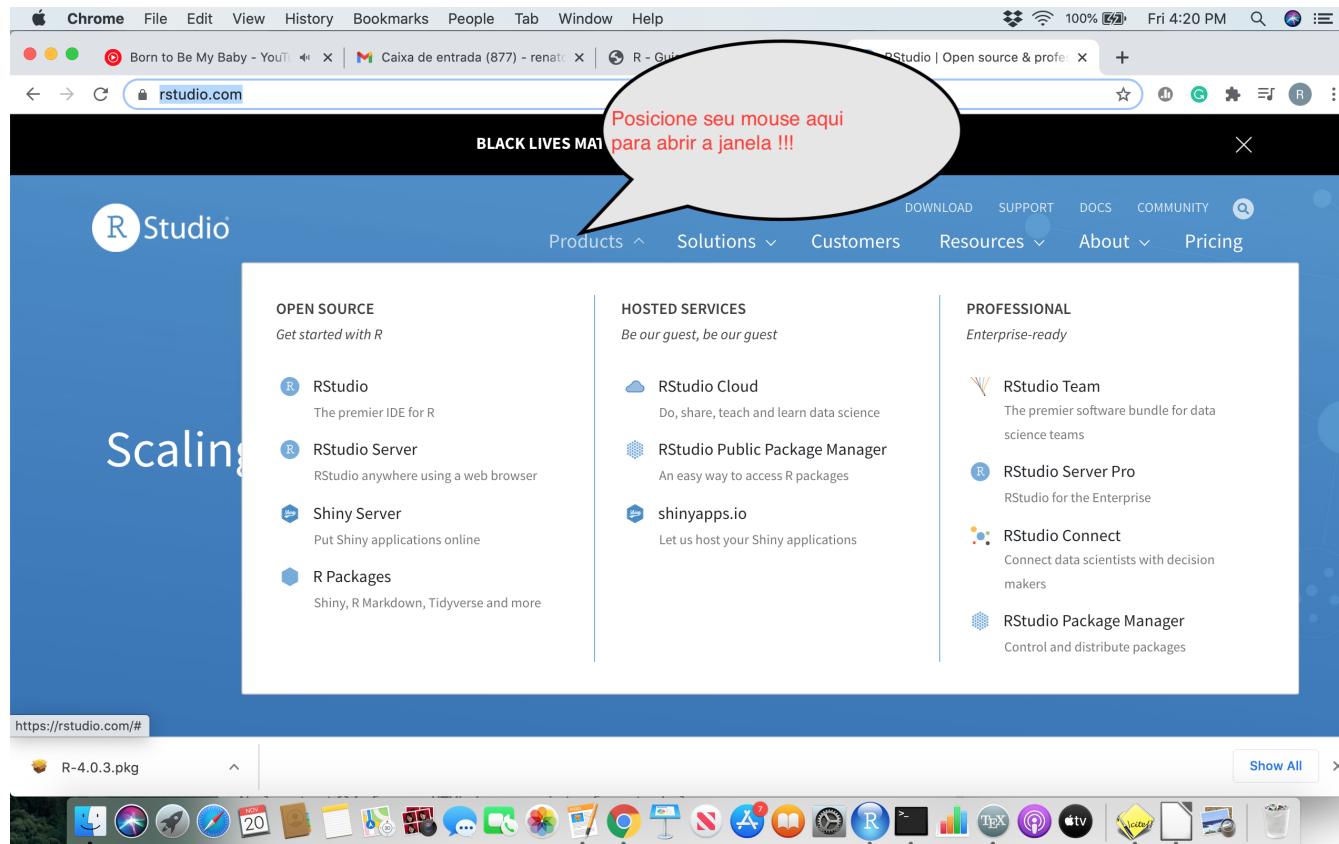
# Instalação do software R Studio - Windows

- Acesse o endereço <https://rstudio.com/>

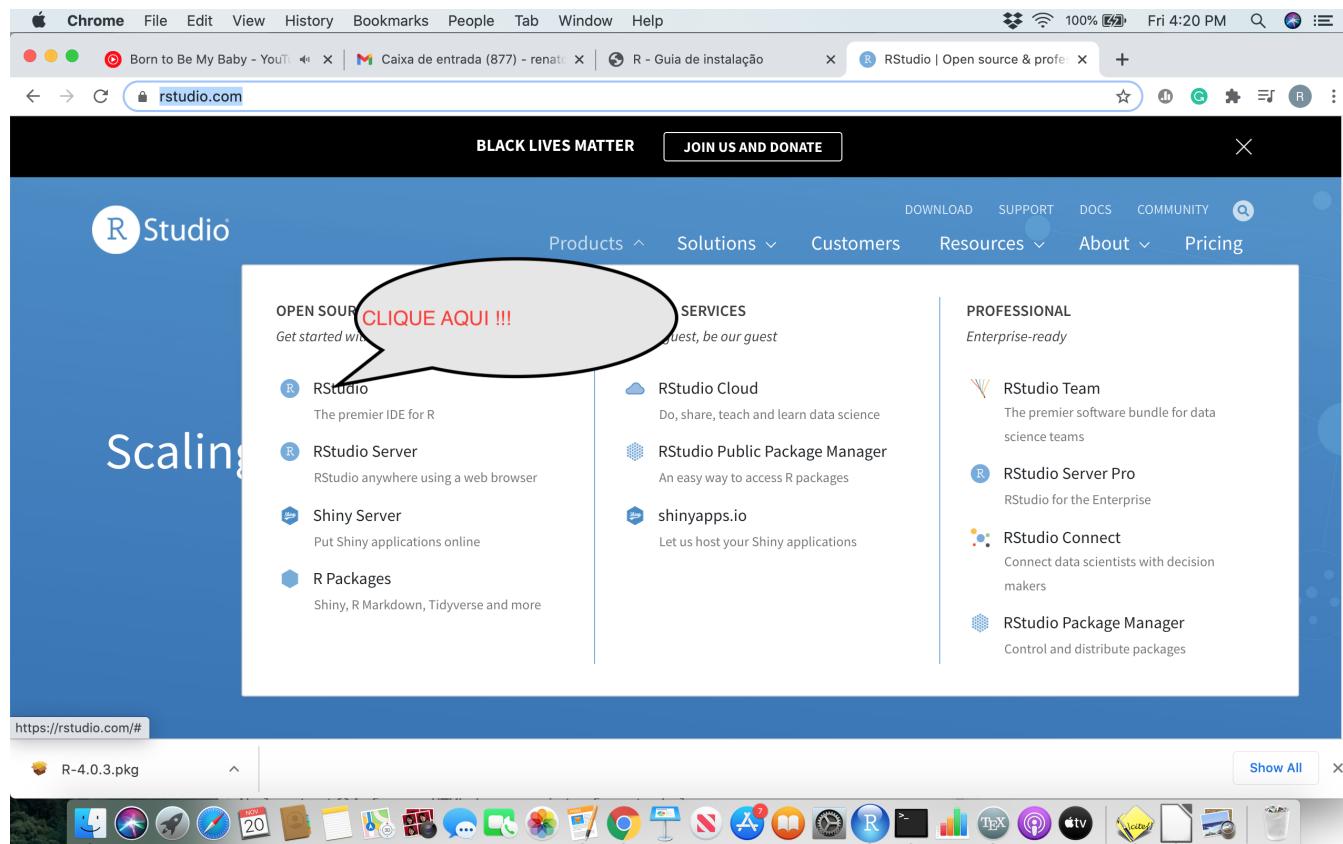


# Instalação do software R Studio - Windows

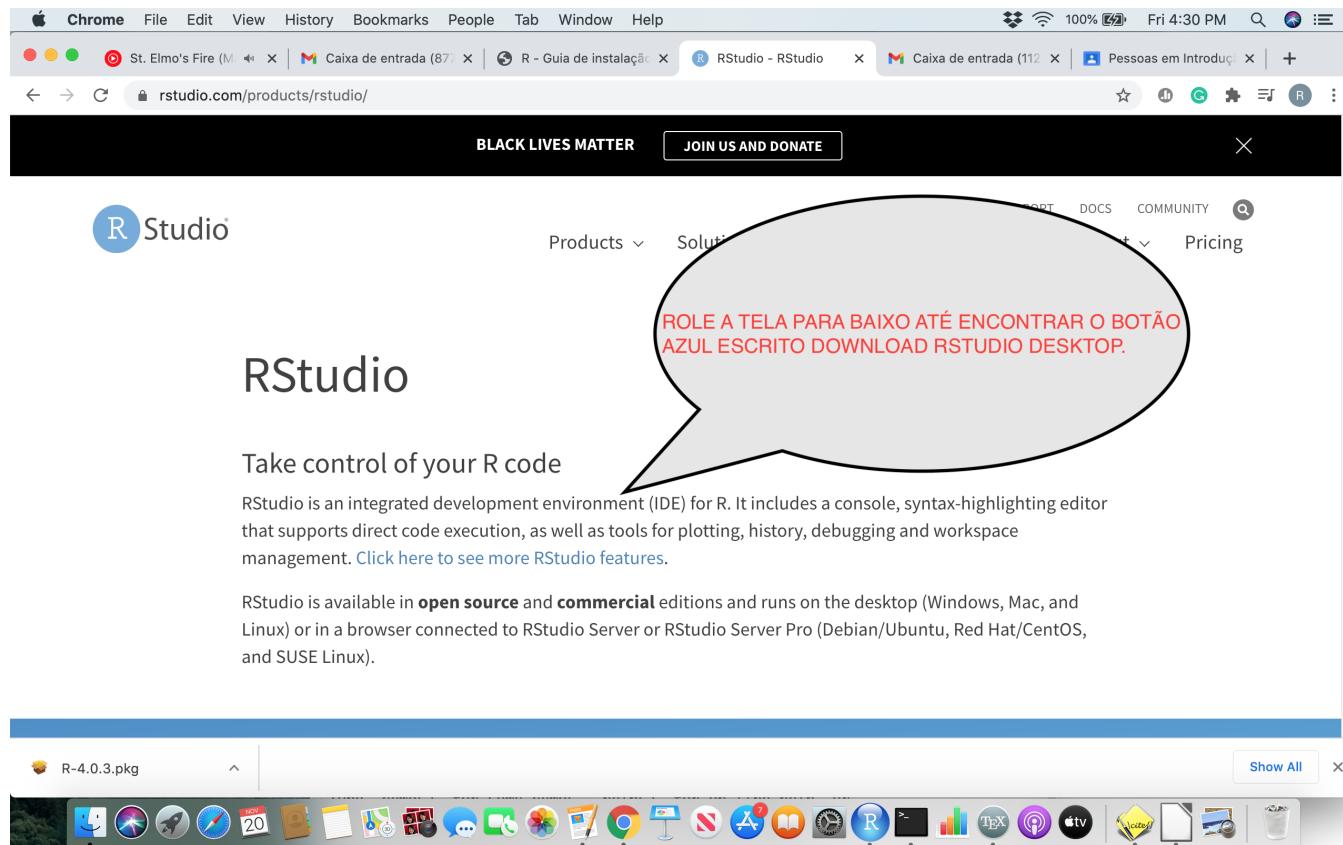
- Passe o mouse na palavra Products



# Instalação do software R Studio - Windows



# Instalação do software R Studio - Windows



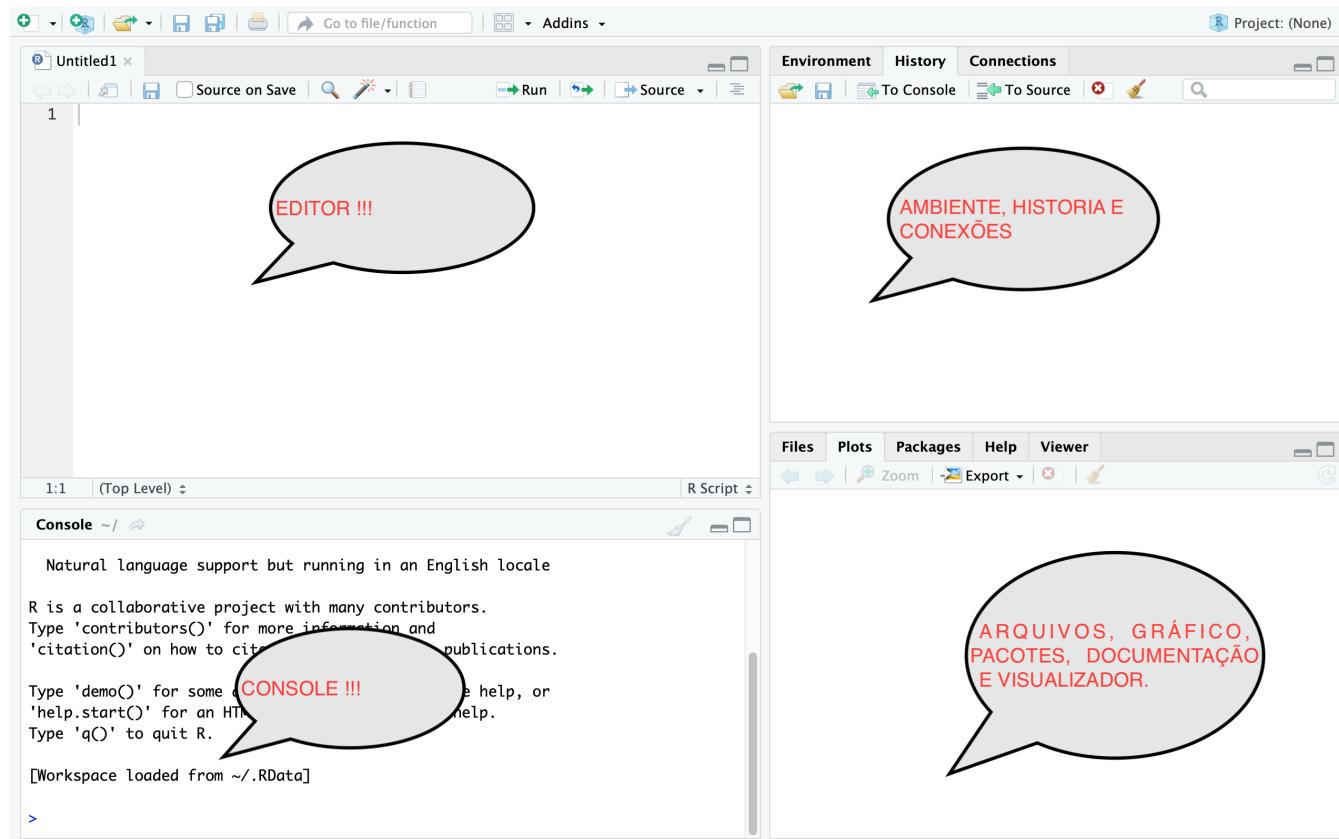
Será exibida uma página com a recomendação para você baixar o RStudio Windows 10/8/7

# Instalação do software R Studio - Mac OS X

A screenshot of a Mac OS X desktop. At the top is the Dock with various icons. Above the Dock is a menu bar with Apple, Chrome, File, Edit, View, History, Bookmarks, People, Tab, Window, Help, and system status indicators. Below the menu bar is a browser window for Chrome. The address bar shows [rstudio.com/products/rstudio/download/](https://rstudio.com/products/rstudio/download/). The page content is for RStudio Desktop 1.3.1093. It lists two steps: 1. Install R. (with a link to RStudio repository) and 2. Download RStudio (with a link to the download page). A large blue button at the bottom says "DOWNLOAD RSTUDIO FOR MAC 1.3.1093 | 148.66MB". A red speech bubble with the text "CLIQUE AQUI !!!" points to the second step. To the right of the browser is a preview of the RStudio interface on a monitor.

Dúvidas sobre instalação, [clique aqui](#).

# Layout R Studio



# Layout R Studio

## Editor

- Local para escrever os códigos. Pode-se executar script de duas formas: (1) através do ícone "Run"; ou através do atalho "ctrl + enter".
- Para salvar o script em um arquivo clique no ícone do disquete.

## Console

- Local os comandos são executados. Como R é uma linguagem interpretada, os comandos aparecem instantâneamente.
- É possível executar comandos diretamente no console.

# Layout R Studio

## Ambiente

- Mostra os objetos, dados e variáveis criadas.

## Históricos

- Local onde armazenado seu histórico do script.

## Conexões

- Aba relacionada a bases de dados. Dificilmente, vamos usar aqui.
- Interessados em maiores detalhes, [clique aqui](#)

# Layout R Studio

## Arquivos:

- Endereço do seu diretório de trabalho, ou seja, do diretório no qual você está trabalhando. Comando `getwd()` imprime o diretório de trabalho. Comando `setwd()` define o diretório de trabalho.

## Gráficos:

- É a aba onde o gráfico vai aparecer

## Pacotes:

- É a aba onde aparece uma lista de pacotes que você pode selecionar e instalar na hora. O comando `install.packages()` instala pacotes de modo mais direto.

# Layout R Studio

## Documentação:

- É a aba onde aparece a documentação com exemplos e explicações sobre alguma função que você possa ter dúvidas de como utilizar. Use o comando `? para pedir ajuda.`

## Visualizador

- A aba "Viewer" é utilizada para visualização de arquivos interativos produzidos.

# Comandos básicos

```
#Adição
```

```
2 + 2
```

```
## [1] 4
```

```
#Subtração
```

```
5 - 10
```

```
## [1] -5
```

```
#Multiplicação
```

```
3 * 2
```

```
## [1] 6
```

```
#Divisão
```

```
80 / 100
```

```
## [1] 0.8
```

# Comandos básicos

```
#Potenciação
```

```
10^2
```

```
## [1] 100
```

```
#Raiz Quadrada
```

```
sqrt(100)
```

```
## [1] 10
```

```
#Logaritmo de 100 na base 10, Logaritmo Natural
```

```
log10(100); log(100)
```

```
## [1] 2
```

```
## [1] 4.60517
```

```
#Função exponencial
```

```
exp(100)
```

```
## [1] 2.688117e+43
```

# Comandos básicos

```
#Note que  
-8^(1/3); (-8)^(1/3)
```

```
## [1] -2
```

```
## [1] NaN
```

```
#Operação módulo - Restante da divisão Euclideana  
5 %% 3
```

```
## [1] 2
```

# Comandos básicos

```
#Valores absolutos  
abs(-4)
```

```
## [1] 4
```

```
#Valores absolutos  
abs(4)
```

```
## [1] 4
```

# Comandos básicos

```
#cosseno
```

```
cos(90); cos(pi/2)
```

```
## [1] -0.4480736
```

```
## [1] 6.123234e-17
```

```
#seno
```

```
sin(90); sin(pi/2)
```

```
## [1] 0.8939967
```

```
## [1] 1
```

```
#tangente
```

```
tan(45); tan(pi/4)
```

```
## [1] 1.619775
```

```
## [1] 1
```

# Tipos de dados/ objetos básicos: numérico, caracteres and lógicos

```
#Numérico
```

```
minha_idade = 38
```

```
minha_idade
```

```
## [1] 38
```

```
#Caracteres
```

```
meu_nome = "Renato"
```

```
meu_nome
```

```
## [1] "Renato"
```

# Tipos de objetos: numéricos, caracteres and lógicos

```
### Trabalha na área da saúde (sim/não) <=> (TRUE/FALSE)
```

```
is_saude = TRUE # variável lógica
```

```
is_saude
```

```
## [1] TRUE
```

Vetores: uma combinação de valores múltiplos (numéricos, caracteres ou lógicos)

```
idades = c(27, 25, 29, 26)
```

```
idades
```

```
## [1] 27 25 29 26
```

# Vetores e atribuição

- Atribuição pode ser também feito com a função can also be made using the function `assign()`.

```
assign("x", c(10.4, 5.6, 3.1, 6.4, 21.7))  
x
```

```
## [1] 10.4 5.6 3.1 6.4 21.7
```

# Vetores e reciclagem no R

- R gosta de operar em vetores do mesmo comprimento,
- Portanto, se ele encontrar dois vetores de comprimentos diferentes em uma operação binária,
- Ele simplesmente replica o vetor menor até que tenha o mesmo comprimento do vetor mais longo e, então, faz a operação.
- Se o vetor menor reciclado tiver que ser "cortado" para torná-lo do comprimento do vetor mais longo, você receberá um aviso, mas ainda retornará um resultado:

# Vetores e reciclagem no R

```
x <- c(1,2,3)
y <- c(1,10)

x * y
```

```
## Warning in x * y: longer object length is not a multiple of shorter object
## length

## [1]  1 20  3
```

# Reciclagem em outros contextos

```
x = 1:20  
x * c(1,0)  
  
## [1] 1 0 3 0 5 0 7 0 9 0 11 0 13 0 15 0 17 0 19 0
```

# Reciclagem em outros contextos

```
x * c(0, 0, 1)
```

```
## Warning in x * c(0, 0, 1): longer object length is not a multiple of short
## object length
```

```
## [1] 0 0 3 0 0 6 0 0 9 0 0 12 0 0 15 0 0 18 0 0
```

```
x < ((1:4)^2)
```

```
## [1] FALSE TRUE TRUE TRUE FALSE FALSE TRUE TRUE FALSE FALSE FALSE FALSE TRUE
## [13] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```

```
1/x
```

```
## [1] 1.00000000 0.50000000 0.33333333 0.25000000 0.20000000 0.16666667
## [7] 0.14285714 0.12500000 0.11111111 0.10000000 0.09090909 0.08333333
## [13] 0.07692308 0.07142857 0.06666667 0.06250000 0.05882353 0.05555556
## [19] 0.05263158 0.05000000
```

# Geração de sequências regulares

- Existem muitas maneiras de gerar uma sequência. Por exemplo,

```
1:10
```

```
## [1] 1 2 3 4 5 6 7 8 9 10
```

```
10:1
```

```
## [1] 10 9 8 7 6 5 4 3 2 1
```

```
(1/2):5
```

```
## [1] 0.5 1.5 2.5 3.5 4.5
```

# Geração de sequência regulares

```
seq(-5, 5, by=.2)
```

```
## [1] -5.0 -4.8 -4.6 -4.4 -4.2 -4.0 -3.8 -3.6 -3.4 -3.2 -3.0 -2.8 -2.6 -2.4
## [16] -2.0 -1.8 -1.6 -1.4 -1.2 -1.0 -0.8 -0.6 -0.4 -0.2  0.0  0.2  0.4  0.6
## [31]  1.0  1.2  1.4  1.6  1.8  2.0  2.2  2.4  2.6  2.8  3.0  3.2  3.4  3.6
## [46]  4.0  4.2  4.4  4.6  4.8  5.0
```

```
seq(length=20, from=-5, to=5)
```

```
## [1] -5.0000000 -4.4736842 -3.9473684 -3.4210526 -2.8947368 -2.3684211
## [7] -1.8421053 -1.3157895 -0.7894737 -0.2631579  0.2631579  0.7894737
## [13]  1.3157895  1.8421053  2.3684211  2.8947368  3.4210526  3.9473684
## [19]  4.4736842  5.0000000
```

# Geração de sequências regulares

```
rep(1:5, times=5)
```

```
## [1] 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5
```

```
rep(1:5, each=5)
```

```
## [1] 1 1 1 1 1 2 2 2 2 3 3 3 3 3 4 4 4 4 4 5 5 5 5 5
```

```
rep(c(1,4,3), c(2,7,6))
```

```
## [1] 1 1 4 4 4 4 4 4 4 3 3 3 3 3 3
```

# Valores faltantes

- Em alguns casos, os componentes de um vetor podem não ser completamente conhecidos.
- Quando um elemento ou valor “não está disponível” ou um “valor ausente” no sentido estatístico, um lugar dentro de um vetor pode ser reservado para ele atribuindo-lhe o valor especial NA.
- Em geral, qualquer operação em um NA torna-se umNA.
- A motivação para esta regra é simplesmente que se a especificação de uma operação estiver incompleta, o resultado não pode ser conhecido e, portanto, não está disponível.
- A função `is.na` (`x`) fornece um vetor lógico do mesmo tamanho que `x` com valor TRUE se e somente se o elemento correspondente em `x` forNA.

# Valores faltantes

- Observe que há um segundo tipo de valores “perdidos” que são produzidos por cálculo numérico, os chamados valores Não um Número, ‘NaN’. Exemplos são

```
0 / 0
```

```
## [1] NaN
```

```
Inf - Inf
```

```
## [1] NaN
```

# Data frames

- Um `data.frame` é uma lista com a classe "data.frame". Existem restrições nas listas que podem ser feitas em frames de dados, nomeadamente
- Os componentes devem ser vetores (numéricos, caracteres ou lógicos), fatores, matrizes numéricas, listas ou outros quadros de dados.
- Matrizes, listas e quadros de dados fornecem tantas variáveis para o novo quadro de dados quantas colunas, elementos ou variáveis têm, respectivamente.
- Vetores numéricos, lógicos e fatores são incluídos no estado em que se encontram e, por padrão, os vetores de caracteres são coagidos a serem fatores, cujos níveis são os valores únicos que aparecem no vetor.
- As estruturas vetoriais que aparecem como variáveis do quadro de dados devem ter o mesmo comprimento e as estruturas matriciais devem ter o mesmo tamanho de linha.

# Fazendo data.frames

```
dat = data.frame(student = c("Renato", "Joao", "Pedro", "Maria", "Joana"),
                 grade = c(10.0, 2.0, 4.0, 9.0, 9.5),
                 approved = c(TRUE, FALSE, FALSE, TRUE, TRUE))

##   student  grade approved
## 1 Renato    10.0     TRUE
## 2 Joao      2.0    FALSE
## 3 Pedro      4.0    FALSE
## 4 Maria      9.0     TRUE
## 5 Joana     9.5     TRUE
```

# Configurando diretório de trabalho

- Antes de ler quaisquer dados, você deve definir o diretório de trabalho R para o local dos dados (SAMHDA, 2017?).
- `setwd ("...")` definirá o diretório de trabalho atual para um local específico
- `getwd ()` imprimirá o diretório atual.
- `> setwd (" C: / mydata ")` (Windows)

**Definir o diretório de trabalho corretamente pode eliminar a confusão de caminho.**

# Lendo dados de arquivos

- Ler dados em um sistema estatístico para análise e exportar os resultados para algum outro sistema para escrever relatórios podem ser tarefas frustrantes.
- Às vezes, leva muito mais tempo do que a própria análise estatística, embora a maioria dos leitores ache esta última muito mais atraente.
- A função `read.table` é a maneira mais conveniente de ler em uma grade retangular de dados.
- Por causa das muitas possibilidades, existem várias outras funções que chamam `read.table`, mas alteram um grupo de argumentos padrão.
- Esteja ciente de que `read.table` é uma maneira ineficiente de ler matrizes numéricas muito grandes.

# Lendo Arquivos de Dados Delimitados - Delimitado por Espaço

Função: `read.table ()`

Parâmetros comuns:

- File: o nome do arquivo do qual os dados devem ser lidos.
- Header: TRUE quando a primeira linha inclui nomes de variáveis. O default é falso.
- Sep: Uma string indicando o que está separando os dados. O default é

```
'> dataSPACE <-read.table("C:/mydata/survey.dat", header=TRUE, sep= " ")'
```

or

```
'> dataSPACE <-read.table("C:/mydata/survey.txt", header=TRUE, sep= " ")'
```

# Lendo Arquivos de Dados Delimitados - Delimitado por Tabulação

Função: `read.table()`

Exemplo:

```
> dataTAB <-read.table("survey.dat", header=TRUE, sep= "\t")
```

ou

```
> dataTAB <-read.table("survey.txt", header=TRUE, sep= "\t")
```

# Lendo Arquivos de Dados Delimitados - Delimitado por Tabulação

```
dat = read.table("Goiaba.txt",header = TRUE)  
head(dat)
```

```
##      A  B  C  D  
## 1  2  1 12  7  
## 2  2  0 10  9  
## 3  1  0 14 15  
## 4  1  1 17  8  
## 5  0  1 11 10
```

```
#View(dat)
```

# Lendo Arquivos de Dados Delimitados - Delimitado por Tabulação

```
dat = read.table("tabela2.1.csv", sep=";", header = TRUE)  
head(dat)
```

```
##   N estado_civil      grau_instrucao n_filhos salario idade_anos idade_meses  
## 1 1     solteiro ensino fundamental     NA    4,00       26          1  
## 2 2     casado  ensino fundamental      1    4,56       32          1  
## 3 3     casado  ensino fundamental      2    5,25       36          1  
## 4 4     solteiro  ensino médio        NA    5,73       20          1  
## 5 5     solteiro ensino fundamental     NA    6,26       40          1  
## 6 6     casado  ensino fundamental      0    6,66       28          1  
##   reg_procedencia  
## 1     interior  
## 2      capital  
## 3      capital  
## 4      outra  
## 5      outra  
## 6     interior
```

```
#View(dat)
```

# Lendo arquivos delimitados por vírgulas

- `read.csv()` lê arquivos delimitados por vírgulas,
- `read.csv2()` lê arquivos separados por ponto e vírgula (comum em países onde é usado como a casa decimal).

# Lendo arquivos delimitados por vírgulas

```
dat = read.csv("student-por.csv", header = TRUE)
```

```
head(dat)
```

```
##   school sex age address famsize Pstatus Medu Fedu      Mjob      Fjob
## 1     GP    F  18        U    GT3       A     4     4 at_home teacher
## 2     GP    F  17        U    GT3       T     1     1 at_home   other
## 3     GP    F  15        U    LE3       T     1     1 at_home   other
## 4     GP    F  15        U    GT3       T     4     2 health services
## 5     GP    F  16        U    GT3       T     3     3   other    other
## 6     GP    M  16        U    LE3       T     4     3 services   other reput
##   guardian traveltime studftime failures schoolsup famsup paid activities
## 1   mother          2         2       0       yes     no   no     no
## 2   father          1         2       0      no     yes   no     no
## 3   mother          1         2       0      yes     no   no     no
## 4   mother          1         3       0      no     yes   no   yes
## 5   father          1         2       0      no     yes   no     no
## 6   mother          1         2       0      no     yes   no   yes
##   nursery higher internet romantic famrel freetime goout Dalc Walc health
## 1   yes    yes     no      no       4     3     4     1     1     3
## 2   no     yes     yes     no       5     3     3     1     1     3
## 3   yes    yes     yes     no       4     3     2     2     3     3
## 4   yes    yes     yes     yes      3     2     2     1     1     5
```

# Lendo arquivos delimitados por vírgulas

```
dat = read.csv2("tabela2.1.csv", header = TRUE)  
head(dat)
```

```
##   N estado_civil      grau_instrucao n_filhos salario idade_anos idade_meses  
## 1 1     solteiro ensino fundamental     NA    4.00       26          1  
## 2 2     casado  ensino fundamental      1    4.56       32          1  
## 3 3     casado  ensino fundamental      2    5.25       36          1  
## 4 4     solteiro  ensino médio        NA    5.73       20          1  
## 5 5     solteiro ensino fundamental     NA    6.26       40          1  
## 6 6     casado  ensino fundamental      0    6.66       28          1  
##   reg_procedencia  
## 1           interior  
## 2            capital  
## 3            capital  
## 4             outra  
## 5             outra  
## 6           interior
```

# Referências

- O Que é GitHub e para que é usado?. HOSTINGER [2015?]. Disponível em: <https://www.hostinger.com.br/tutoriais/o-que-github/>. Acesso em: 19 de nov. de 2020.
- The Devmountain blog, DEVMOUNTAIN [2015?]. Disponível em: <https://blog.devmountain.com/git-vs-github-whats-the-difference/#:~:text=GitHub%E2%80%A6,what's%20the%20difference%3F,help%20> Acesso em: 19 de nov. de 2020.
- GOGONI, R. O que é um sistema operacional? Tecnoblog. Disponível em: <https://tecnoblog.net/303055/o-que-e-um-sistema-operacional/>. Acesso em: 20 de nov. de 2020.
- GCFAprendeLivre. Informática Básica - Sistemas operacionais para o computador. Disponível em: <https://edu.gcfglobal.org/pt/informatica-basica/sistemas-operacionais-para-o-computador/1/>. Acesso em: 20 de nov. de 2020.

# Referências

- Caminho (computação). WIKIPEDIA [2000?]. Disponível em: [https://pt.wikipedia.org/wiki/Caminho\\_\(computa%C3%A7%C3%A3o\)](https://pt.wikipedia.org/wiki/Caminho_(computa%C3%A7%C3%A3o)). Acesso em: 20 de nov. de 2020.
- Formatos de caminho de arquivo em sistemas Windows. MICROSOFT [2000?]. Disponível em: <https://docs.microsoft.com/pt-br/dotnet/standard/io/file-path-formats>. Acesso em: 20 de nov. de 2020.
- Como Encontrar o Caminho de um Arquivo no Windows. wikiHow [2020?]. Disponível em: <https://pt.wikihow.com/Encontrar-o-Caminho-de-um-Arquivo-no-Windows>. Acesso em: 20 de nov. de 2020.
- FONTANA, B.; CASTRO, M.;C.;A. Comandos linux para manipulação de diretórios. Curso Integrado em Telecomunicações. Disciplina: Introdução a Computação (ICO). 2014.
- HORNIK 2020, “The R FAQ”, and give the above, official URL: <https://CRAN.R-project.org/doc/FAQ/R-FAQ.html>. Acesso em: 20 de nov. de 2020.
- Instalação. CURSO-R. [2018?] Disponível em: <http://material.curso-r.com/installacao/>. Acesso em: 20 de nov. de 2020.

# Referências

- Notes on R: A Programming Environment for Data Analysis and Graphics (W. N. Venables, D. M. Smith and the R Core Team, 2004)
- ANDERSON, E. C. Reproducible Research Course. Disponível em: [https://eriqande.github.io/rep-res-web/lectures/vectorization\\_recycling\\_and\\_indexing.html](https://eriqande.github.io/rep-res-web/lectures/vectorization_recycling_and_indexing.html). Acesso em: 20 de nov. de 2020.
- How do I read data into R? SAMHDA [2017?]. Disponível em: <https://www.datafiles.samhsa.gov/faq/how-do-i-read-data-r-nid3445>. Acesso em: 20 de nov. de 2020.