

# Análise dos Dados com R

---

## Introdução

James R. Hunter, PhD

27 de outubro de 2024

# Introdução

Onde tudo começa

# O Que É Nossa Objetivo

- Aprender análise de dados **em prática**
  - Fazer análises de cabo ao rabo
  - Com programação (não só clicar botões)
- Utilizar a Linguagem **R**

# Professor James Hunter

- Professor Afiliado, DIPA
- DSc., Laboratório de Retrovirologia, DIPA, UNIFESP
- Projeto de Cura de HIV
- Carreira anterior em consultoria de negócios e urbanismo
- Foco em Estatística e Métodos Quantitativos desde 1973
- Trabalho com R desde 2010

# Contato com o Professor

- email: jameshunterbr@gmail.com
- Bluesky: [jimhunterbr](#)
- cel: 11-95629-6628
- Office Hours:
  - Quinta-feira 14h - 16h
  - EP2, Rua Pedro de Toledo 669, 6º Andar Fundos

# Filosofia da Matéria

- Única maneira para aprender uma linguagem de computação é escrever ela
- Mais código que você escreve, mais fácil será a próxima vez
- Solucionar problemas práticos com R

**Don't Panic...**







# Perguntas

- Fazem muitas!
- Se você tiver uma dúvida, outros na turma terão também
- **Não existe perguntas burras**

# Carl Sagan sobre Perguntas Burras

- Astrofísico que escreveu e era apresentador do programa de TV original **Cosmos**
- Livro : **The Demon-Haunted World: Science as a Candle in the Dark (O mundo assombrado pelos demônios)**

There are naive questions, tedious questions, ill-phrased questions, questions put after inadequate self-criticism. But every question is a cry to understand the world. There is no such thing as a dumb question.

# Sempre Existe uma Segundo Ponto de Vista



# Quanta Matemática Você Precisa Dominar?

- O que aprendeu no colegial suficiente
- Não precisa cálculo
- Somas (  $\Sigma$  ), logaritmos e expoentes
- Equação de uma linha reta

$$y = b_0 + b_1x$$

# Informação e Conhecimento

"We are drowning in information, but we are starved for knowledge". -- John Naisbitt

Apesar esta frase seja atribuído a futurólogo John Naisbitt, esta citação tem muitos pais e mães. Usei aqui do livro de Danielle Navarro, **Learning statistics with R: A tutorial for psychology students and other beginners**, 2020, <http://compcogscisydney.org/learning-statistics-with-r>

# Porque Nós Precisamos Análise dos Dados?

- Podemos ver as coisas que estudamos? NÃO
  - Vírus, bactérias, células, nucleotídeos, proteínas
- Maquinas que produzem os dados genômicas que estudamos são probalísticos
  - Palavra "*calling* bases" - sugestão de erro
- Processo natural de replicação celular ou viral - propenso a erros
- Resposta humano às doenças, remédios, tratamentos
  - Nível alto de incerteza e variancia
  - Diferenças naturais entre pessoas

# Estatística Ajuda a Encontrar Verdades Subjacentes

- Desenvolver conjunto das regras para processar informação que recebemos
  - Script/Programa
- Tirar conclusões que outros podem entender, concordar ou discordar
- Como alunos, precisam poder conduzir análises básicas
  - Modelos e métodos mais avançados ficam com especialistas



# Habilidade Necessária para Todo Cientista

- Entender as estatísticas que você vê em papers e livros
- Separar o que é importante do que não é importante
- Separar a verdade de falsidade
- "Call Bullshit"\* quando você está sendo enganado
- Resultado: precisamos maneiras probabilísticas para achar essas verdades subjacentes

# **R - Uma Ferramenta para Manipulação e Análise dos Dados**

# CRAN: The Comprehensive R Archive Network

- Uma ONG educacional quem é o dono do código mãe de R
- Fonte oficial para cópias do software base e pacotes averiguados por eles

R is a system for statistical computation and graphics. It consists of a language plus a run-time environment with graphics, a debugger, access to certain system functions, and the ability to run programs stored in script files.

# Historia de R

- Baseada na linguagem de programação estatística ("S")
  - S desenvolvida por Bell Labs em 1976
  - Ainda existe como um produto comercial
- R desenvolvida por Ross Ihaka e Robert Gentleman em 1995 em Nova Zelândia
- Comunidade ativa de desenvolvedores e usuários
- Mais que 19.800 pacotes adicionais disponíveis no repositório de CRAN
  - Muitos úteis para as análises biológicas
  - Bioconductor -- outro 2.000 pacotes
  - Muitos outros espalhados em vários repos (e.g. GitHub)

# Virtudes de R para Análise de Dados

- Analisar via programas e scripts invés de clicar botões
  - Controlar a sequência e opções de operações em sua análise
- Programas sempre fazem a mesma coisa - produzem mesmo resultado
  - Sem surpresas porque você clicou em um botão que mudou sua análise
  - Só usam opções e parâmetros que você entende
- Criar um registro de como você chegou no resultado
- **De Graça** Sem custo, para sempre!
  - Não tem uma versão "estudantil" estupidamente cara
  - Nem precisa cópias piratas do software

# A Crise de Reprodutibilidade

- Sendo capaz de reproduzir análises em tempos diferentes e em labs diferentes
- Maioria dos estudos científicos não podem ser reproduzidos
- *Nature's* Checklist de Reprodutibilidade

Workflows based on point-and-click interfaces, such as Excel, are not reproducible. Enshrine your computations and data manipulation in code.\*

- R e Python trunfa Excel, Graphpad e seus amigos

\*Perkel. Challenge to Scientists Nature 584, no. 7822 (2020).

# R - Difícil de Aprender?

- Se você nunca programou antes, todas as linguagens de computação parecem difíceis ao início
- R muito mais fácil que a maioria
- Passos Iniciais
  - Criar vetores e conjuntos de dados ("*data frames*")
    - Executar funções estatísticas e matemáticas
  - Vamos começar hoje escrever código
- R torna mais difícil quando você começa de escrever suas próprias funções
  - Quando não pode achar eles nos pacotes que tem

# O Que Vocês Devem Fazer

- Investir tempo entre as aulas
- Instalar os softwares (R e RStudio) nos seus laptops
- Ler o material sugerido aqui
- Experimente um dos cursos de R Básico no internet
  - Ter um segundo olhar sobre o mesmo material



# RStudio -- Comunicação Sofisticada com R

- Integrated Development Environment ("IDE") para R
- Disponível desde 2010
- Sede de *Tidyverse*
- Onde vocês vão fazer seu trabalho em R
- Também **De Graça**

# R & Python

- Python - outra linguagem bastante popular
  - Baseada em conceitos similares aos do R
  - Outra linguagem de alto-nível interpretada
- Lançado em 1991
  - Guido van Rossum de Holanda
  - Nome vem do grupo comédico inglês, "Monty Python's Flying Circus"
  - Não a espécie de cobra
- Para estatística, mais fraco de R
  - Precisa funções de vários módulos para conseguir completar operações básicas de estatísticas

# Recursos para a Matéria

# Arquivos, Slides, etc.

- Arquivado na página do curso no Google Classroom e repo de Github

# Leituras Chaves

- Textos de Estatística
  - Diez, Barr & Cetinkaya-Rundel, **OpenIntro Statistics 4**
  - Navarro, D. **Learning statistics with R: A tutorial for psychology students and other beginners**
- Livros sobre R - Nível Básico
  - Wickham & Grolemund, **R for Data Science**
  - Ismay & Kim, **Statistical Inference via Data Science: A modern dive into R and the Tidyverse**
  - Irizarry, **Introduction to Data Science**
  - Frank E. Harrell, **R Workflow** (<http://hbiostat.org/rflow/>)

# RStudio "Cheat Sheets"

Série de resumos de 1 e 2 páginas de um número de pacotes de funções em R

## Base R Cheat Sheet

### Getting Help

Accessing the help files

**?mean**  
Get help of a particular function.  
**help.search('weighted mean')**  
Search the help files for a word or phrase.  
**help(package = 'dplyr')**  
Find help for a package.

More about an object

**str(iris)**  
Get a summary of an object's structure.  
**class(iris)**  
Find the class an object belongs to.

### Using Packages

**install.packages('dplyr')**  
Download and install a package from CRAN.

**library(dplyr)**  
Load the package into the session, making all its functions available to use.

### Vectors

#### Creating Vectors

<code>c(2, 4, 6)</code>	<code>2 4 6</code>	Join elements into a vector
<code>2:6</code>	<code>2 3 4 5 6</code>	An integer sequence
<code>seq(2, 3, by=0.5)</code>	<code>2.0 2.5 3.0</code>	A complex sequence
<code>rep(1:2, times=3)</code>	<code>1 2 1 2 1 2</code>	Repeat a vector
<code>rep(1:2, each=3)</code>	<code>1 1 1 2 2 2</code>	Repeat elements of a vector

#### Vector Functions

<b>sort(x)</b> Return x sorted.	<b>rev(x)</b> Return x reversed.
<b>table(x)</b> See counts of values.	<b>unique(x)</b> See unique values.

#### Selecting Vector Elements

By Position

<code>x[4]</code>	The fourth element.
<code>x[-4]</code>	All but the fourth.
<code>x[2:4]</code>	Elements two to four.

### Programming

#### For Loop

```
for (variable in sequence){  
  Do something  
}
```

Example

```
for (i in 1:4){  
  j <- i + 10  
  print(j)  
}
```

#### While Loop

```
while (condition){  
  Do something  
}
```

Example

```
while (i < 5){  
  print(i)  
  i <- i + 1  
}
```

#### If Statements

```
if (condition){  
  Do something  
} else {  
  Do something different  
}
```

Example

```
if (i > 3){  
  print('Yes')  
} else {  
  print('No')  
}
```

#### Functions

```
function_name <- function(var){  
  Do something  
  return(new_variable)  
}
```

Example

```
square <- function(x){  
  squared <- x*x  
  return(squared)  
}
```

### Reading and Writing Data

Also see the [readr](#) package.

# Cursos Online

- edX - Cursos de Harvard sobre R com Prof. R. Irizzary
  - <https://www.edx.org/learn/r-programming/harvard-university-data-science-r-basics>
- Coursera - Cursos de Johns Hopkins sobre R e outros sobre R em aplicações biomedicos
  - <https://www.coursera.org/specializations/jhu-data-science>
- Coursera - Duke University - sequence of R courses by Cetinkaya-Rundel
  - <https://www.coursera.org/specializations/statistics?>

All excellent

# Sites sobre R

- R Bloggers (<https://www.r-bloggers.com/>)
- Tidyverse (<https://www.tidyverse.org/learn/>)
- Stack Overflow (<https://stackoverflow.com/questions/tagged/r>)
- Twitter (#rstats)



# Sistemas de Ajuda de R e RStudio

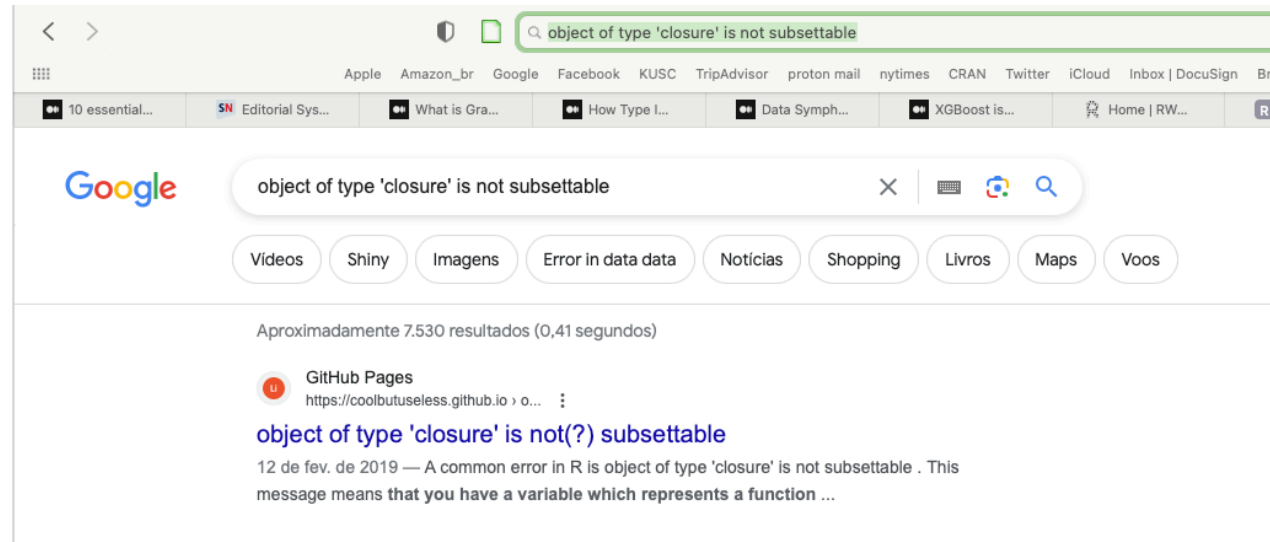
- Completo
- Toda função tem uma tela de ajuda
- Escrito por nerds para outros nerds
  - Explicações às vezes opacas
  - Especialmente mensagens de erro
- Último recurso: copiar a mensagem de erro e colar ele no Google
  - Alguém, em algum lugar, também não entendeu a mesma coisa que é problemática para você

# Aplicando Google para um Erro

- O Erro

```
> mean[1:10]  
Error in mean[1:10] : object of type 'closure' is not subsettable
```

- Último recurso: copiar a mensagem de erro e colar ele no Google
  - Alguém, em algum lugar, também não entendeu a mesma coisa que é problemática para você



# Seu Programa Primeiro

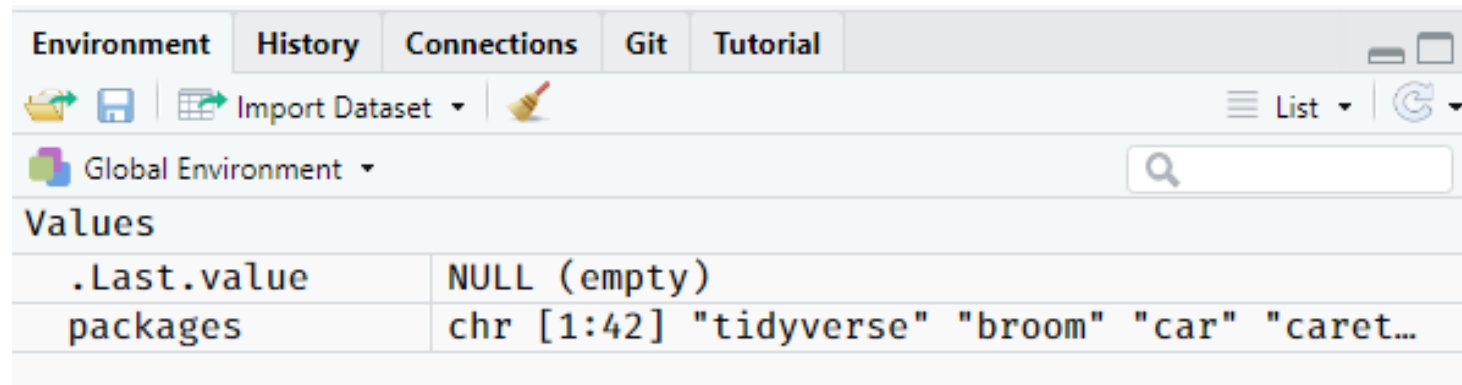
# Carregar os Pacotes Importantes

- Os pacotes mais importantes que potencializam R
- Usaremos a maioria durante estas 4 semanas
- Script simples (`pacotes_iniciais.r`)

```
packages <- c("tidyverse", "broom", "car", "caret", "corrr", "corrplot",  
             "data.table", "DescTools", "devtools", "gapminder", "ggpubr",  
             "ggsci", "glue", "gmodels", "gt", "gtsummary", "here", "Hmisc",  
             "hms", "janitor", "jsonlite", "kableExtra", "knitr", "lattice",  
             "lubridate", "nortest", "nycflights13", "paletteer",  
             "patchwork", "plotly", "palmerpenguins", "pROC", "psych",  
             "quarto", "Rcpp", "readxl", "reticulate", "ROCR", "shiny",  
             "styler", "summarytools", "tidymodels", "titanic", "usethis",  
             "DataExplorer")  
  
install.packages(packages)
```

# O Que Faz Este Script - Linha 1

- Linha 1: **atribuição** de conjunto de pacotes ao nome `packages`
  - Utiliza `<-` para fazer a atribuição
- Conjunto de pacotes é combinado num **vetor** de nomes de pacotes
  - Função `c()` cria um vetor de vários elementos
  - `c()` - *combinar* or *concatenar*
  - *vector* - matriz unidimensional
- Elementos de `packages` - "**strings**" de classe *character*
  - Entre aspas ("" )
- Resultado da Linha 1



The screenshot shows the RStudio Environment pane. At the top, there are tabs for 'Environment', 'History', 'Connections', 'Git', and 'Tutorial'. Below the tabs is a toolbar with icons for file operations and a search bar. The 'Global Environment' is selected. Under the 'Values' section, a table displays the contents of the environment. The table has two columns: the variable name and its value. The first row shows '.Last.value' as 'NULL (empty)'. The second row shows 'packages' as 'chr [1:42] "tidyverse" "broom" "car" "caret...'. The ellipsis indicates that there are more elements in the vector.

Values	
.Last.value	NULL (empty)
packages	chr [1:42] "tidyverse" "broom" "car" "caret...

# VSS: Operadores de Atribuição

- Principal: `<-`
- Pode usar (mas não é considerada uma boa prática) `=`
  - **Vai confundir com o sinal para igualdade lógica** `==`
    - Vai acontecer! Todos nos fazemos

# O Que Faz Este Script - Linha 2

- Instala os pacotes que estão no vetor "packages"
- Procura no site de CRAN (espelho) no internet
- Faz os downloads e instala os pacotes
- Vários dos pacotes têm dependências
  - Instalará esses pacotes também
- Dependências: outros pacotes que um pacote precisa para executar as funções do pacote primeiro

# Scripts vs. Console

- Escrever seus comandos num script de R Markdown invés do Console
  - Pode salvar seu trabalho
- Console é o lugar onde os comandos são executados
  - Mais fácil de salvar comandos em scripts que salvando a historia dos comandos do Console



# Executar "pacotes\_iniciais.r"

- Download o arquivo do repo da aula para sua pasta de R
- A aba `Files` no painel inferior direito do RStudio
  - Clicar em `pacotes_iniciais.r`
- Script abrirá no painel superior esquerdo
- Clicar no botão `Source` na barra de comandos
- Pode seguir o progresso no Console

# R - Operações Básicas

# R como um Calculadora

```
5 + 5
```

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

```
36 * 2500000
```

```
## [1] 9e+07
```

```
5876/35.44320
```

```
## [1] 165.7864
```

```
2^25 # exponent
```

```
## [1] 33554432
```

```
25 * (12 + 27)
```

```
## [1] 975
```

# Funções Matemáticas em R

Função	O Que Ela Faz
<code>abs(x)</code>	valor absoluto de x
<code>sqrt(x)</code>	raiz quadrado de x
<code>log(x)</code>	logaritmo natural (naperiano) de x
<code>exp(x)</code>	exponente natural de x
<code>log10(x)</code>	logaritmo base 10 de x
<code>round(x, n)</code>	arredondar x para n casas decimais

# Funções em Operação

```
sqrt(9849)
```

```
## [1] 99.24213
```

```
log(377898)
```

```
## [1] 12.84238
```

```
exp(12.84238)
```

```
## [1] 377898.2
```

```
log10(377898)
```

```
## [1] 5.577375
```

```
round(exp(12.84238), 0)
```

```
## [1] 377898
```

# Sobre `log()` e `exp()`

- No exemplo acima, expoente do 12.84238 é 377898.2, não 377898
- R relata 5 casas decimais na tela
  - Internamente, é 12.8423795969182 (13 casas decimais)
- Sabemos que  $\log(x) = e^x$
- Não quebramos as leis da matemática.

```
x <- 377898
y <- log(x) # calcular o log de x e atribuir a y
y
```

```
## [1] 12.84238
```

```
exp(y)
```

```
## [1] 377898
```

# Comentários

- Linha 2 do script tem um comentário à direta

```
y <- log(x) # calcular o log de x e atribuir a y
```

- Comentários começam com um hashtag #
  - Tudo à direita do # não será interpretado (executado)
  - Podem ser em linhas separadas (até melhor)
- Comentários nos lembra o que fizemos e porque
- **Hiper importantes**
- Usem eles MUITO

# Ordem de Cálculo (*PEMDAS*)

Operação	Simbolo	Exemplo	PEMDAS
parênteses	()	$5 * (7 + 2) = 45$	P
exponentes	^	$5^2 = 25$	E
multiplicação	*	$5 * 7 = 35$	M
divisão	/	$25/5 = 5$	D
adição	+	$5 + 7 = 12$	A
subtração	-	$5 - 7 = -2$	S

- Se você retirar os parênteses de  $5 * (7 + 2)$ ?
- $5 * 7 + 2 = 37$
- VSS: regras de matemática não mudam porque usamos um computador



# Atribuição

- (nome de objeto) <- (definição do objeto)
- definição = valores que compõem o conteúdo do objeto

# Atribuição - Estilos

- Esses servem

```
x <- 6  
x <- "Hi!"
```

- Esses funcionam mas não recomendo e não uso

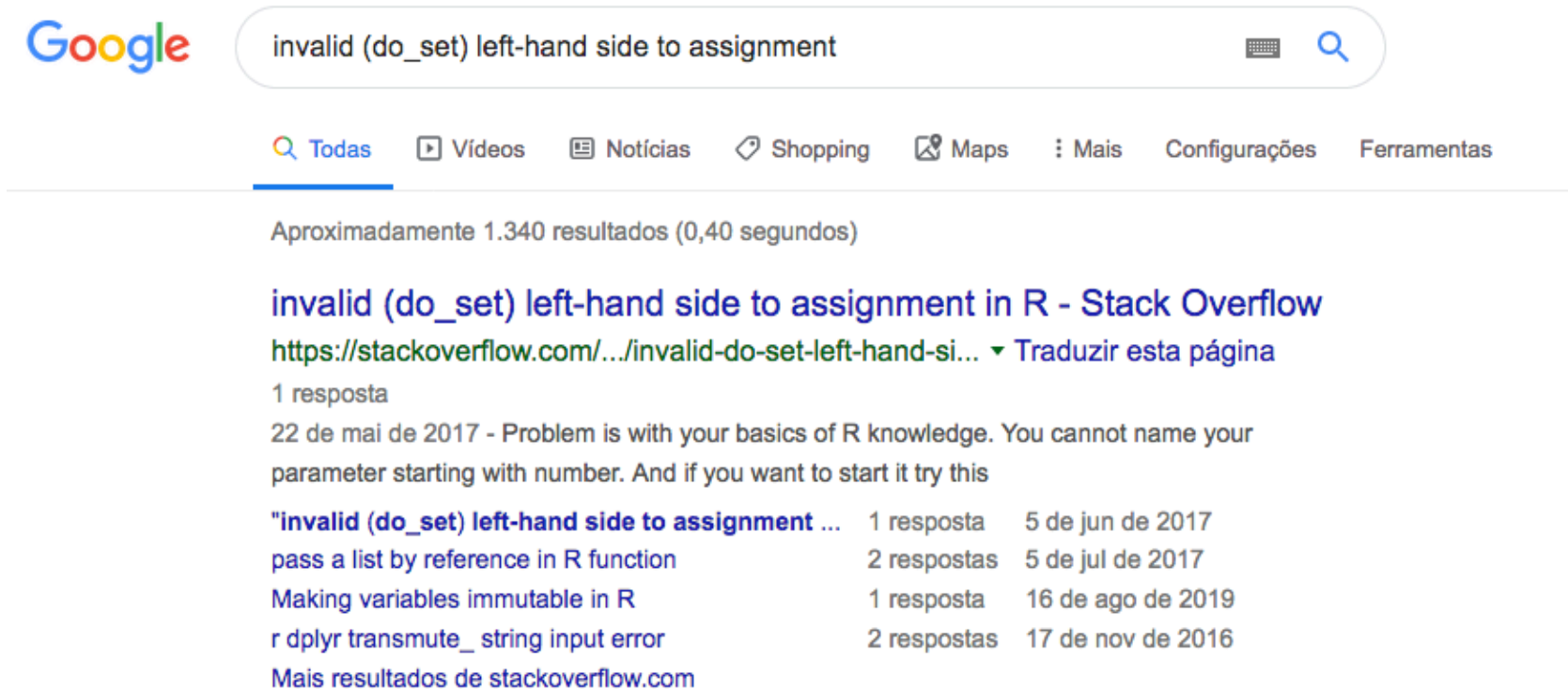
```
x = 6  
6 -> x
```

- Esse produz um erro (não pode iniciar um comando com um número)

```
> 6 = x  
Error in 6 = x : invalid (do_set) left-hand side to assignment  
> |
```

# Mensagens de Erro Estranhas?

- Consulte Dr. Google



The image shows a Google search interface. The search bar contains the text "invalid (do\_set) left-hand side to assignment". Below the search bar, there are navigation links: "Todas", "Vídeos", "Notícias", "Shopping", "Maps", "Mais", "Configurações", and "Ferramentas". The search results show approximately 1,340 results in 0.40 seconds. The top result is titled "invalid (do\_set) left-hand side to assignment in R - Stack Overflow" with a URL starting with "https://stackoverflow.com/.../invalid-do-set-left-hand-si...". It has 1 response and is dated May 22, 2017. Below this, there is a list of related search suggestions with their respective response counts and dates.

Google

invalid (do\_set) left-hand side to assignment

Todas Vídeos Notícias Shopping Maps Mais Configurações Ferramentas

Aproximadamente 1.340 resultados (0,40 segundos)

**invalid (do\_set) left-hand side to assignment in R - Stack Overflow**  
<https://stackoverflow.com/.../invalid-do-set-left-hand-si...> ▼ Traduzir esta página

1 resposta

22 de mai de 2017 - Problem is with your basics of R knowledge. You cannot name your parameter starting with number. And if you want to start it try this

"invalid (do_set) left-hand side to assignment ...	1 resposta	5 de jun de 2017
pass a list by reference in R function	2 respostas	5 de jul de 2017
Making variables immutable in R	1 resposta	16 de ago de 2019
r dplyr transmute_ string input error	2 respostas	17 de nov de 2016

Mais resultados de stackoverflow.com

# Atribuição -- Nomes das Variáveis

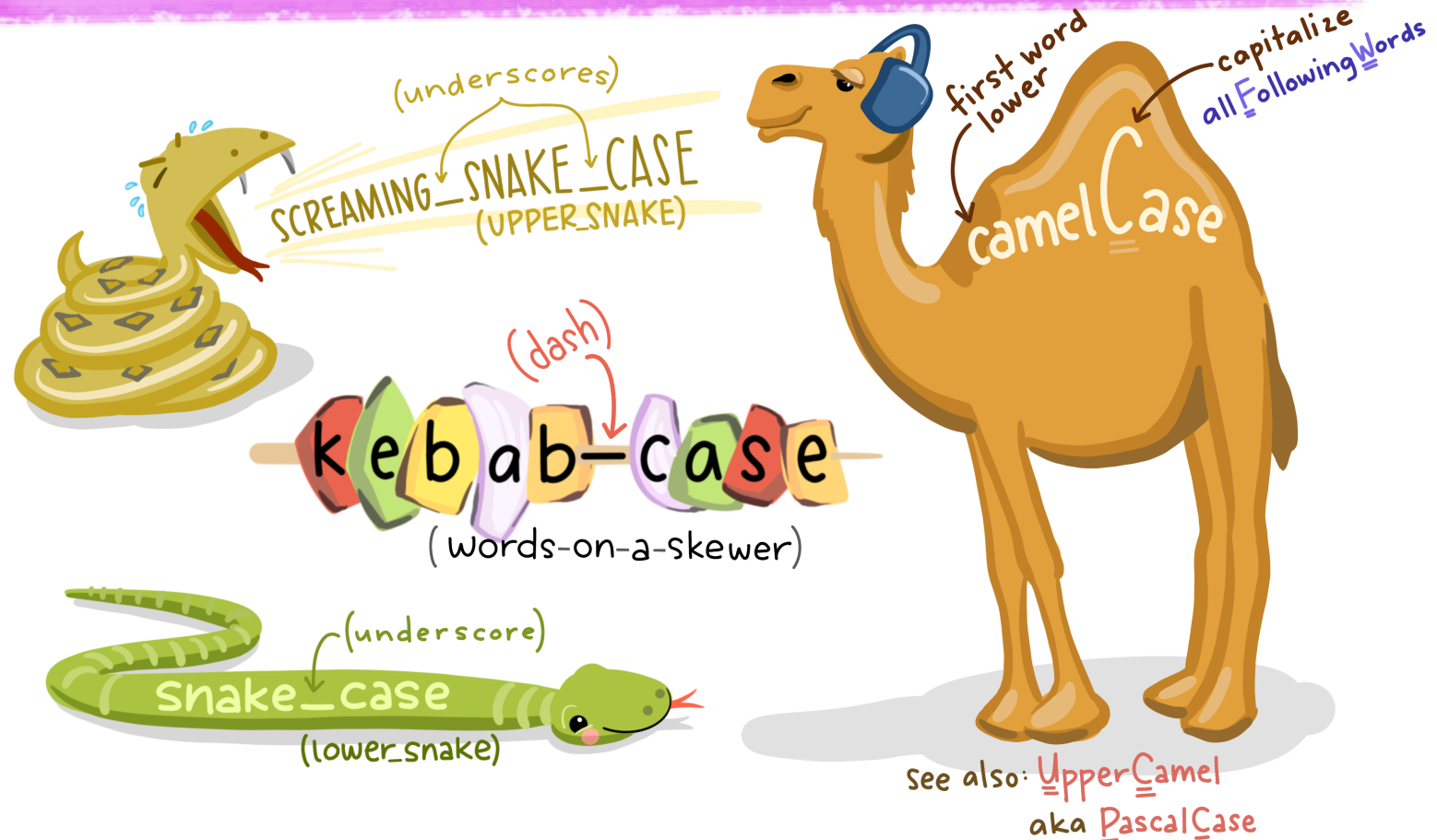
- Regras importantes de R
  1. Deve usar só letras (maiúsculas ou minúsculas), números ou símbolos `.` or `_`.
  2. Deve iniciar o nome com uma **letra**

# Nomes das Variáveis -- Corolários

- Não devem incluir espaços
  - "Snake case" supera essa restrição
    - Conectar palavras com sublinhar "\_"
- Palavras reservadas de R não podem ser usados como nomes de variáveis
  - Exemplos: `TRUE`, `FALSE`, `if`, `else`, `for`, `function`
- Nomes de variáveis diferenciam maiúsculas de minúsculas
  - `Variable` e `variable` são 2 nomes diferentes
  - Mesmo para `x` e `X`

# Casos em R

in that case...



# Nomes das Variáveis - Ainda Mais

- Usar nomes claros e informativos
  - `x`, apesar de ser popular, é inútil como um nome

```
## 1a versao
peso <- 55 ## Pessoa pesa 55 kg.

## 2a versao
peso_kg <- 55 ## Mais claro

## 3a versao, pode converter às libras
peso_lb <- peso_kg * 2.2
peso_lb
```

```
## [1] 121
```

# Nomes das Variáveis - Último

- Faça um dicionário dos dados
  - Tabela dos nomes das variáveis, qual tipo de dados, e o intervalo dos valores
- Tente de fazer os nomes mais curtos possíveis



# Estilo

- Estilo é importante
- Guia de Estilo de R
  - Wickham, H. **R Style Guide** (<https://style.tidyverse.org/>)
- Olhe no documento **style\_guide.pdf** no repo da aula/Google Classroom