Introdução ao R

R, história, pacotes, RStudio e primeiro contato

Prof. Me. Lineu Alberto Cavazani de Freitas

Departamento de Estatística
Laboratório de Estatística e Geoinformação

Estatística e o desenvolvimento computacional

- A popularização da Estatística se dá graças ao **desenvolvimento computacional**.
- Os computadores pessoais tornaram os métodos estatísticos mais acessíveis ao público geral por meio de **softwares** que implementam as metodologias.
- Devido ao avanço computacional, houve um aumento considerável na capacidade de **produzir e armazenar dados** provenientes das mais diversas fontes.

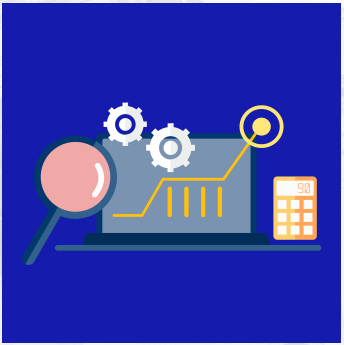


Figura 1. Extraído de pixabay.com.

Estatística e o desenvolvimento computacional

- Graças ao avanço computacional podemos lidar com a manipulação de **grandes conjuntos de dados**.
- Este grande volume de dados também força o **desenvolvimento dos métodos estatísticos** e softwares para análise de dados.
- A capacidade computacional atual também desperta o interesse por **métodos estatísticos computacionalmente intensivos**.

Figura 2. Extraído de pixabay.com.

Ferramentas para análises estatísticas

Existem diversas ferramentas disponíveis:

- R;
- Python;
- SAS;
- Spss;
- Biostat;
- Minitab;
- Tableau;
- Stata;
- E diversas outras.

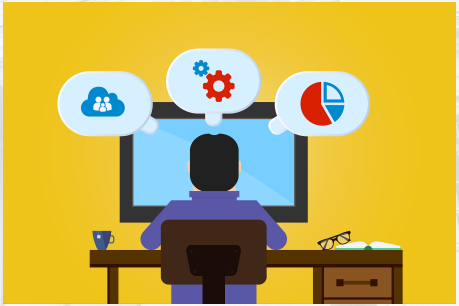


Figura 3. Extraído de pixabay.com.

Ferramentas para análises estatísticas

Existem diversas ferramentas disponíveis:

- **R**;
- Python;
- SAS;
- Spss;
- Biostat;
- Minitab;
- Tableau;
- Stata;
- E diversas outras.

Figura 4. Logo do R.

R

- R é uma linguagem e ambiente para **computação estatística** e **gráficos**.
- É **livre** e de **código aberto**.
  - Livre (free): usuários tem liberdade de:
    1. executar como desejar e para qualquer propósito.
    2. estudar o funcionamento e adapta-lo à necessidades específicas.
    3. distribuir cópias de versões originais e modificadas.
  - Código aberto (open source): o acesso ao código fonte é gratuito.

R

- Muito popular no meio **acadêmico** e tem uso cada vez maior no meio **corporativo**.
  - É acessível e gratuito.
  - Tem diversas técnicas e aplicações possíveis.

- Tem potencial uso em todas as etapas do processo de análise de dados.
  - Obtenção, importação, manipulação e tratamento.
  - Análise exploratória.
  - Ajuste de modelos estatísticos, modelos de aprendizado de máquina, dentre outros.
  - Elaboração de relatórios dinâmicos e reproduzíveis.

Antes do R

- O R sucedeu a lingugem S, de **John Chambers**, Rick Becker, Trevor Hastie, Allan Wilks e outros.
- A linguagem S foi iniciada em 1976 e disponibilizada publicamente no início dos anos 80 já como um ambiente de análise estatística.
- Uma das principais limitações da linguagem S era que ela só estava disponível em um **pacote comercial**: o S-PLUS. O que motivou a criação do R.
- Quando o R foi desenvolvido a sintaxe era muito semelhante ao S, pensando na migração dos usuários de uma linguagem para outra.

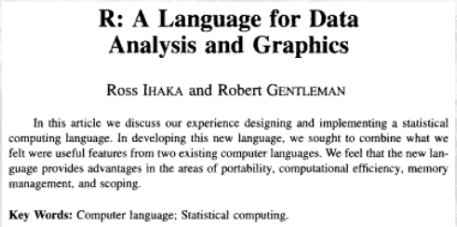
História do R

- Em **1991** o R foi criado por Ross Ihaka e Robert Gentleman (“**R & R**”) no Departamento de Estatística da Universidade de Auckland.
- Em **1993** o R foi **anunciado publicamente** pela primeira vez.
- Em **1995** o R passou a usar a **GNU General Public License**, o que tornou o R um software livre e de código aberto.

Figura 5. Robert Gentleman e Ross Ihaka.

História do R

- Em **1996** foi publicado o **artigo** em que Ross e Robert descrevem sua proposta: "*R: A language for data analysis and graphics.*" no Journal of Computational and Graphical Statistics.
- Ainda em **1996** foram criadas as listas de discussão **R-help** e **R-devel**.



R: A Language for Data Analysis and Graphics

Ross IHAKA and Robert GENTLEMAN

In this article we discuss our experience designing and implementing a statistical computing language. In developing this new language, we sought to combine what we felt were useful features from two existing computer languages. We feel that the new language provides advantages in the areas of portability, computational efficiency, memory management, and scoping.

Key Words: Computer language; Statistical computing.

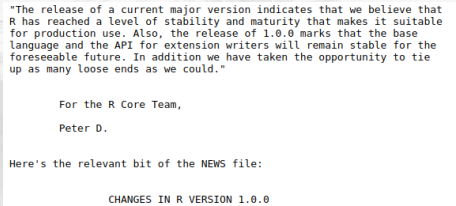
Figura 6. Artigo original do R.

História do R

- Em **1997** foi formado o **R Core Team**: um grupo de aproximadamente 20 desenvolvedores que mantêm, gerenciam, controlam o código fonte e orientam a evolução da linguagem.
- Os membros do R Core Team fundaram a **R Foundation**: uma organização sem fins lucrativos que trabalha no interesse público para dar suporte ao R.
- Os direitos autorais do código-fonte primário do R pertencem à R Foundation e são publicados sob a GNU General Public License versão 2.0.

História do R

- Em **2000** o R 1.0.0 foi lançado.
- O R base está disponível para instalação no **Comprehensive R Archive Network**, também conhecido como **CRAN**.



```
"The release of a current major version indicates that we believe that
R has reached a level of stability and maturity that makes it suitable
for production use. Also, the release of 1.0.0 marks that the base
language and the API for extension writers will remain stable for the
foreseeable future. In addition we have taken the opportunity to tie
up as many loose ends as we could."

        For the R Core Team,

        Peter D.


Here's the relevant bit of the NEWS file:

                CHANGES IN R VERSION 1.0.0
```

Figura 7. R 1.0.0.

O que é o R

- O R é uma **linguagem de programação**.
- Uma linguagem de programação é a forma que nós nos comunicamos com o computador.
- Trabalharemos com a ideia de interface de linha de comando (command line interface): escreveremos códigos para que o computador entenda e execute a tarefa de interesse.
- Faremos a análise de dados escrevendo **funções e scripts**, não apontando, clicando e arrastando caixas.
- Para quem nunca programou, parece assustador. Mas o R é fácil de aprender e guiado a análise de dados.

O que é o R

- É possível instalar e usar o R nos principais sistemas operacionais.
- Assim como vários outros softwares livres e de código aberto, o R tem lançamentos frequentes de **versões**.
- A comunidade R é altamente ativa com usuários no mundo todo que contribuem, desenvolvem pacotes e ajudam uns aos outros por meio de materiais online como listas de discussão e tutoriais.

R e os pacotes

- **Pacotes R** são coleções de funções R, dados e código compilado.
- O R já vem com um conjunto de pacotes por padrão e outros podem ser adicionados para estender os recursos.
- Os pacotes hoje disponíveis são o resultado de anos de colaboração de pessoas de todo o mundo.
- Uma das funções do CRAN é hospedar diversos pacotes complementares.

Figura 8. Extraído de pixabay.com.

R-base

- O R “base” contém 15 pacotes básicos necessários para executar o R e as funções mais fundamentais.
- As funções já vem disponíveis, prontas para chamada e uso.

1. `base`
2. `compiler`
3. `datasets`
4. `grDevices`
5. `graphics`
6. `grid`
7. `methods`
8. `parallel`
9. `splines`
10. `stats`
11. `stats4`
12. `tcltk`
13. `tools`
14. `translations`
15. `utils`.

R-recommended

- O R vem equipado com outros 15 pacotes “recomendados”.
- Apesar de já instaladas, estas bibliotecas precisam ser chamadas para que seja possível usar as funções.

1. `KernSmooth`
2. `MASS`
3. `Matrix`
4. `boot`
5. `class`
6. `cluster`
7. `codetools`
8. `foreign`
9. `lattice`
10. `mgcv`
11. `nlme`
12. `nnet`
13. `rpart`
14. `spatial`
15. `survival`.

Outros pacotes

- Você pode facilmente obter e instalar pacotes além dos 30 que já vem com a instalação tradicional do R.
- A principal fonte de pacotes é o próprio CRAN, que hoje conta com **mais de 19000 pacotes**.
- Fontes secundárias envolvem páginas web e repositórios como github, onde desenvolvedores mantém pacotes em desenvolvimento.

Available CRAN Packages By Name

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

A3	Accurate, Adaptable, and Accessible Error Metrics for Predictive Models
Aalen.Johansen	Conditional Aalen-Johansen Estimation
AATtools	Reliability and Scoring Routines for the Approach-Avoidance Task
ABACUS	Apps Based Activities for Communicating and Understanding Statistics
abbreviate	Readable String Abbreviation
abbyyR	Access to Abbyy Optical Character Recognition (OCR) API
abc	Tools for Approximate Bayesian Computation (ABC)
abc.data	Data Only: Tools for Approximate Bayesian Computation (ABC)
ABC.RAP	Array Based CpG Region Analysis Pipeline
ABCanalysis	Computed ABC Analysis
abclass	Angle-Based Large-Margin Classifiers
ABCoptim	Implementation of Artificial Bee Colony (ABC) Optimization
ABCp2	Approximate Bayesian Computational Model for Estimating P2
abcrf	Approximate Bayesian Computation via Random Forests
abcrlda	Asymptotically Bias-Corrected Regularized Linear Discriminant Analysis
abctools	Tools for ABC Analyses
abd	The Analysis of Biological Data
abdiv	Alpha and Beta Diversity Measures
abe	Augmented Backward Elimination
abess	Fast Best Subset Selection
abglasso	Adaptive Bayesian Graphical Lasso
ABHgenotypeR	Easy Visualization of ABH Genotypes
abind	Combine Multidimensional Arrays
abjData	Databases Used Routinely by the Brazilian Jurimetrics Association

Figura 9. Lista de pacotes disponíveis por nome no CRAN.

Em resumo

O R fornece

- Diversos recursos de Estatística.
- Diversos recursos gráficos.
- Uma vasta coleção de pacotes oficiais e não oficiais.
- Uma linguagem de programação bem desenvolvida, simples e eficaz.
- Possibilidade de instalação e uso nos mais comuns sistemas operacionais.
- Possibilidade de uso de códigos C, C++ e Fortran para tarefas computacionalmente intesivas.
- Documentação padronizada.

IDEs e Editores

- Existem softwares adicionais úteis para ajudar a programar de forma mais rápida e eficiente.
- As IDE's (*Integrated Development Environment*) são softwares que oferecem algumas facilidades para se programar em determinada linguagem.
- Já os editores tendem a ser úteis para múltiplas linguagens e fornecem mais alternativas de customização.
- Para trabalhar em R, dentre IDEs e editores, o RStudio IDE é a opção mais famosa.

Figura 10. Logo do RStudio.

Rstudio

- Atenção:
  - O R é a **linguagem**.
  - O RStudio é a **interface**.
  - Quem faz o trabalho (manda a solicitação para que o computador execute) é o R!

Figura 11. Extraído de pixabay.com.

Abrindo o RStudio

Abrindo o RStudio

- O RStudio como IDE tem o papel de facilitar o trabalho em R.
- Portanto, ao abrir o RStudio, tudo que você precisa deve estar à mostra e com fácil acesso.
- Ao abrir o RStudio você verá uma estrutura organizada em paineis.

Figura 12. Logo do RStudio.

Abrindo o RStudio

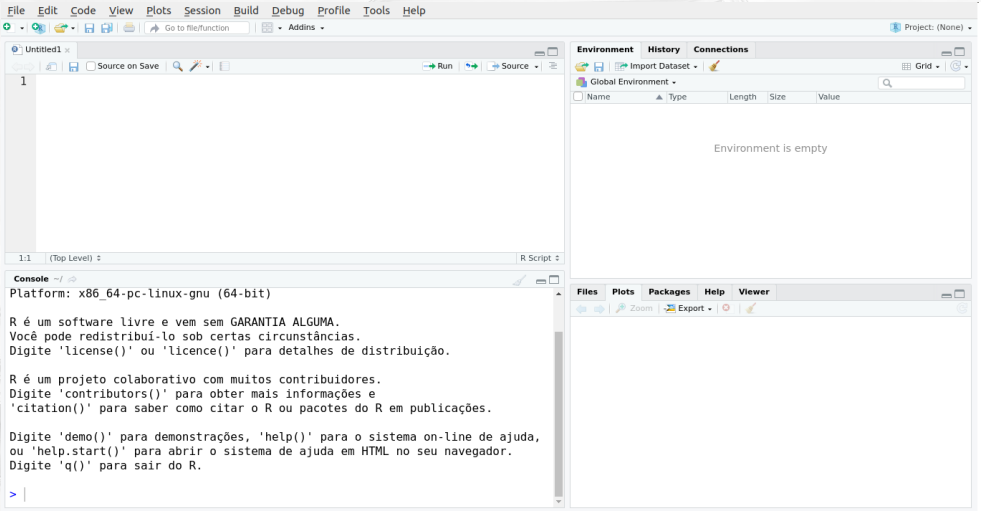


Figura 13. Tela inicial do RStudio.

Paineis

- No **canto superior esquerdo** é onde digitamos o código R que será executado: o **editor**.
  - Se o R está recém instalado ou a sessão é nova, crie um arquivo .R clicando em `File > New File > R` ou use as teclas de atalho `Ctrl + Shift + N`.
- No **canto inferior esquerdo** é onde o terminal R está, é ali onde o código é interpretado e executado: o **console**.
- No canto superior direito são mostradas informações do ambiente de trabalho, histórico, conexões, etc.
- No canto inferior direito são apresentados os arquivos da pasta de trabalho, gráficos, pacotes, documentação etc.

Editor

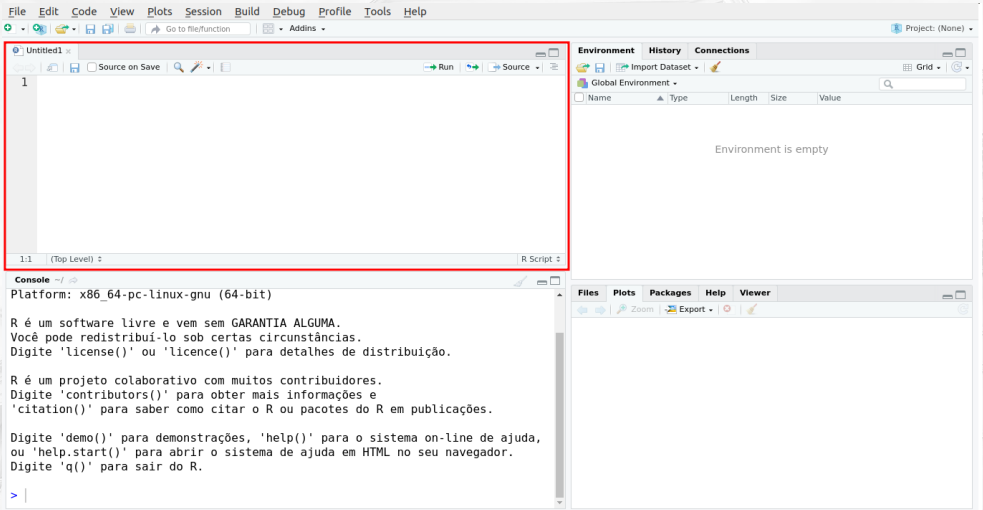


Figura 14. Tela inicial do RStudio. Foco no editor.

Console

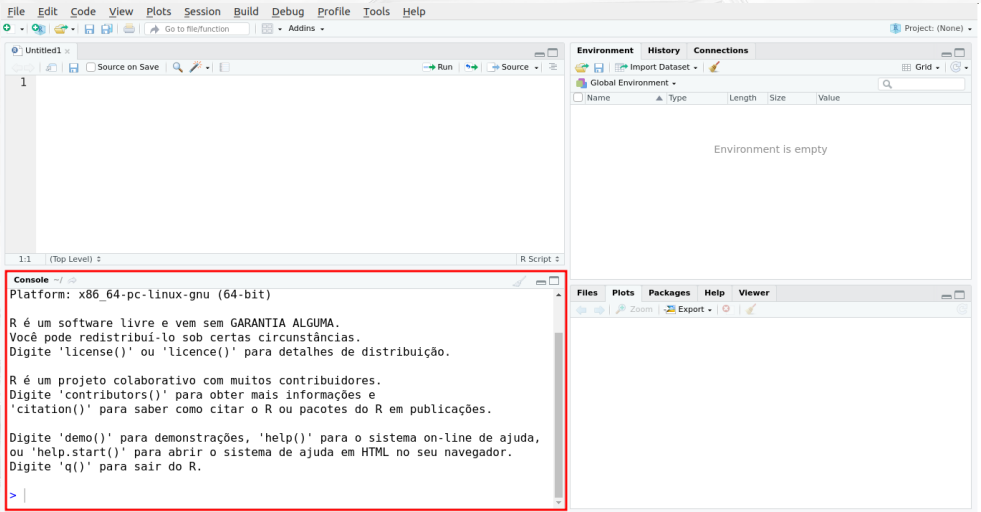


Figura 15. Tela inicial do RStudio. Foco no console.

Ambiente, histórico, conexões

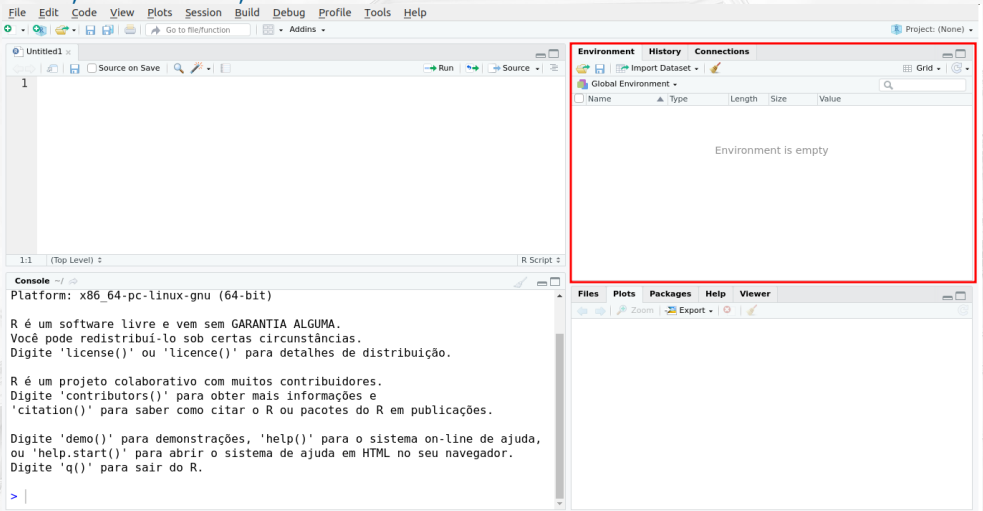


Figura 16. Tela inicial do RStudio. Foco no ambiente, histórico e conexões.

Arquivos, gráficos, pacotes e documentação

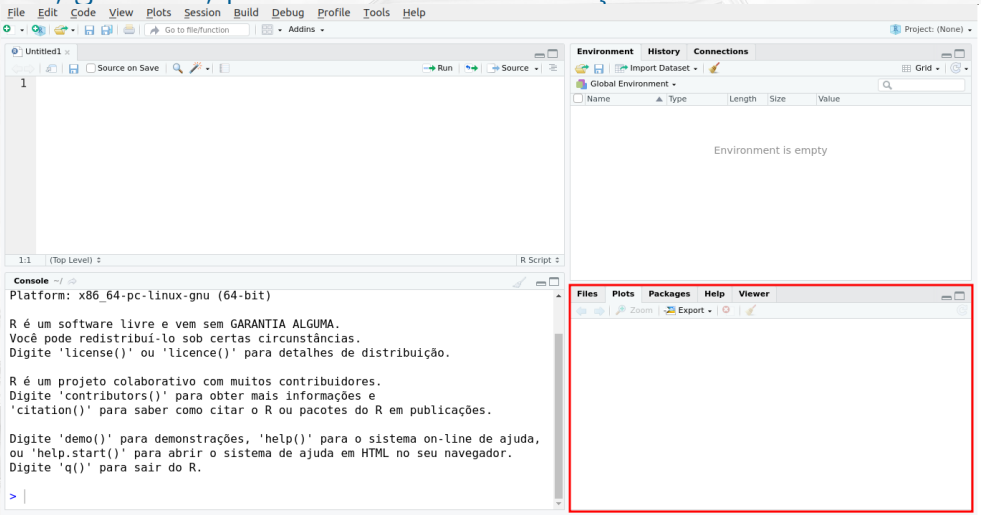


Figura 17. Tela inicial do RStudio. Foco nos arquivos, pacotes e documentação.

Principais elementos

Em resumo:

- **Editor**: onde escrevemos os códigos.
- **Console**: onde os resultados são printados.
- **Environment**: mostra todos os objetos criados.
- **History**: mostra todos os códigos executados.
- **Files**: mostra os arquivos no diretório atual.
- **Plots**: mostra os outputs de códigos que geram gráficos.
- **Packages**: mostra os pacotes instalados.
- **Help**: mostra a documentação de funções e pacotes.

Dica: Para obter todos os atalhos da interface digite pressione `ALT+SHIFT+K`.

Customizando o RStudio

- O RStudio permite customizar os elementos.
- Experimente acessar `Tools > Global Options > Appearance` para trocar temas, ordem dos paineis, tamanho da fonte, etc.
- Deixe o RStudio confortável para você.

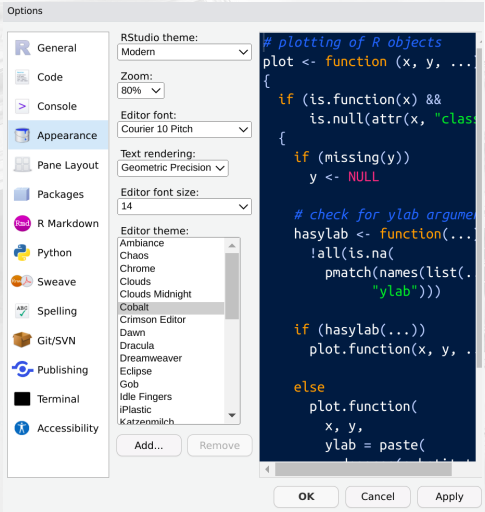


Figura 18. Tela inicial do RStudio.

Trabalhando com R no RStudio

Primeiros passos

1. Crie uma pasta em algum lugar do seu computador para trabalhar.
2. Abra o RStudio e defina o diretório de trabalho.
   - O diretório de trabalho é a pasta do R onde estamos trabalhando.
   - `Session > Set Working Directory > Choose Directory...` ou `CTRL + SHIFT + H`.
     - Ambas as opções darão a possibilidade de escolher a pasta de interesse no computador onde manteremos nossos arquivos.

Primeiros passos

3. Crie um arquivo com extensão .R.
   - `File > New File > R script`.
   - `CTRL + SHIFT + N`.
4. Salve seu script.
   - Esta etapa funciona como qualquer arquivo do computador.
   - `File > Save` ou `CTRL + S`.
   - De um nome para seu arquivo.
     - Opte por nomes curtos, intuitivos e evite espaços, acentos e caracteres.

Agora estamos prontos para trabalhar!

Instruções e comentários

- Uma **instrução** é um código a ser executado.
- Um **comentário** é algo que escrevemos no script mas que não temos interesse que seja executado.
- Comentários podem e devem ser usados para **documentar** o código.
- Tudo que vier após # é um comentário e não será executado pela linguagem.

Executando

- Para executar uma instrução no RStudio basta ir até a linha de interesse e teclar CTRL + ENTER.
- O código é interpretado, executado e o resultado é mostrado na tela.
- Uma recomendação para scripts mais organizados é não ultrapassar 72 ou 80 caracteres por linha.
- Outra recomendação diz respeito à indentação ou alinhamento do código.
  - Use `CTRL+i` no RStudio para indentar automaticamente.

R como calculadora

R como calculadora

- O R pode ser usado como uma poderosa **calculadora científica**.
- Os operadores seguem uma hierarquia, ou seja, uma ordem de precedência.
  - Inicialmente são efetuadas as operações entre parênteses seguindo a ordem: exponenciação, multiplicação/divisão e por fim adição/subtração.
- Para utilizar os operadores no R basta digitar os valores e a operação diretamente no console (caso queira ver somente o resultado) ou no editor (caso deseje salvar o código no arquivo .R).

Operações aritméticas básicas

```r
# Soma
1 + 1
```

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

```r
# Subtração
1 - 1
```

```
## [1] 0
```

```r
# Multiplicação
2 * 2
```

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

```r
# Divisão
4/2
```

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

```r
# Potenciação
5^2
```

```
## [1] 25
```

```r
# Radiciação
2^(1/3)
```

```
## [1] 1.259921
```

```r
# Resto
10 %% 3
```

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

```r
# Parte inteira
10 %/% 3
```

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

Funções trigonométricas

```r
# Seno
sin(0)
```

```
## [1] 0
```

```r
# Arco seno
asin(0)
```

```
## [1] 0
```

```r
# Cosseno
cos(0)
```

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

```r
# Arco cosseno
acos(1)
```

```
## [1] 0
```

```r
# Tangente
tan(0)
```

```
## [1] 0
```

```r
# Arco tangente
atan(0)
```

```
## [1] 0
```

Funções matemáticas especiais

```r
# Exponencial base e
exp(1)
```

```
## [1] 2.718282
```

```r
# Log qualquer base
log(10, base = 5)
```

```
## [1] 1.430677
```

```r
# Arredondamento para cima
ceiling(1.2)
```

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

```r
# Raiz quadrada
sqrt(4)
```

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

```r
# Fatorial
factorial(4)
```

```
## [1] 24
```

```r
# Arredondamento para baixo
floor(1.2)
```

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

```r
# Log neperiano
log(10)
```

```
## [1] 2.302585
```

```r
# Valor absoluto
abs(-1)
```

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

```r
# Arredondamento
round(1.2, digits = 0)
```

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

Operadores lógicos

```r
# São iguais?
1 == 1
```

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

```r
# São iguais?
1 == 2
```

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

```r
# São diferentes?
2 != 2
```

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

```r
# São diferentes?
1 != 2
```

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

```r
# 2 é menor ou igual a 1?
2 <= 1
```

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

```r
# 2 é maior ou igual a 1?
2 >= 1
```

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

```r
# 2 é maior do que 1?
2 > 1
```

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

```r
# 2 é menor do que 1?
2 < 1
```

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

E, OU e NÃO

- Combinação de resultados lógicos.
- 1 é menor que 5 **E** 2 é maior ou igual a 3?

```
(1 < 5) & (2 >= 3)
```

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

- 1 é menor que 5 **OU** 2 é maior ou igual a 3?

```
(1 < 5) | (2 >= 3)
```

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

- 2 é menor que 5? Inverta a resposta lógica.

```
!(2 < 5)
```

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

Valores especiais

- NA: valores ausentes.
- NULL: objetos vazios.
- Inf e -Inf: infinitos.
- NaN: indeterminações.

```
1 + NA
```

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

```
1 + NULL
```

```
## numeric(0)
```

```
1/0
```

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

```
0/0
```

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

Variáveis

Variáveis

- No R podemos usar **objetos** para atribuir valores que serão usados recorrentemente.
- Por exemplo, suponha que estamos trabalhando com o valor 10 e que este valor será usado várias e várias vezes no código.
- Para poupar algum trabalho, podemos atribuir este valor a um objeto. Por exemplo, `x`:

```
x <- 10
```

Variáveis

- Usamos o operador de atribuição <- (lemos RECEBE) para atribuir o valor 10 à variável x.
- O sinal de igual (=) também pode ser usado, mas o <- é mais comum e recomendado.
- Note que ao fazer uma atribuição o resultado não é printado no terminal.
- Para que o resultado seja printado, digite o nome da variável em uma nova linha e execute.

Variáveis

- Ao fazer uma atribuição, criamos um objeto na nossa área de trabalho.
- Para listar os objetos criados na área de trabalho usamos a função `ls()`.
- Se repetirmos o código de atribuição com outro valor, vamos o sobrescrever. Portanto, cuidado!
- Quando há a necessidade de apagar um objeto da área de trabalho usamos a função `rm()`.

Variáveis

```r
# Atribui o valor 10 à variável x
x <- 10

# Printa o valor de x
x
```

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

```r
# Lista as variáveis
ls()
```

```
## [1] "format_field" "thm"          "x"
```

```r
# Remove a variável x
rm(x)

# Lista as variáveis
ls()
```

```
## [1] "format_field" "thm"
```

Funções básicas

Funções básicas

- Além dos objetos, **funções** são um elemento importante em R.
- Na prática funções também são objetos.
- O R já vem com diversas funções básicas.
- Algumas já vimos nos operadores matemáticos.

Funções básicas

Função	Tarefa
c()	Cria um Vetor
setwd()	Muda o Diretório de Trabalho Atual
getwd()	Mostra o Diretório de Trabalho
dir()	Lista os Arquivos do Diretório de Trabalho Atual
sessionInfo()	Mostra algumas informações da sessão
install.packages()	Instala um pacote
library()	Carrega um pacote
require()	Carrega um pacote
example()	Mostra exemplos de alguma função
print()	Imprime o resultado de uma variável
q()	Fecha a Sessão
objects()	Exibe objetos criados
str()	Mostra a estrutura de um objeto

Funções de ajuda

Funções de ajuda

- Algo ótimo de se trabalhar em R é a **documentação interna**.
- Em R cada função e objeto tem a sua própria documentação.
- Em alguns casos, para pacotes que fazem algumas análises especificas, temos tutoriais que são chamados de **vinhetas** (vignettes).
- Existem funções que auxiliam no acesso à documentação.

Funções de ajuda

- Para acessar a documentação específica de uma função ou objeto podemos usar as funções `?` ou `help()`.
- A documentação aparecerá no painel no canto inferior direito do RStudio.
- Suponha que você não saiba exatamente o nome da função ou objeto para o qual você quer consultar a documentação.
- Neste caso, você pode procurar por funções e objetos por meio de algum termo de busca com a função `help.search()`.

Funções de ajuda

- Uma outra forma de obter ajuda é usar a função `apropos()`.
- Ela vai procurar por objetos no caminho de procura que batem com o termo que você está procurando.
- Já as vignettes estão associadas a pacotes específicos.
- Podemos consultar todos os vignettes disponíveis dentro de um pacote com a função `browseVignettes()`.
- O R abrirá uma nova janela em seu browser onde mostrará os títulos dos vignettes disponíveis para o pacote solicitado.
- Outra possibilidade é a função `RSiteSearch()` que fará uma busca mais geral procurando pelo termo no site oficial do R (r-project.org).

Funções de ajuda

```r
# Solicita a documentação interna do pacote base
?base
help(base)

# Busca pelo termo 'linear models'
help.search("linear models")

# Busca funções e objetos pelo nome parcial
apropos("plot")

# Busca vinhetas
browseVignettes("grid")

# Busca um termo no site do R
RSiteSearch("plot")
```

Campos da documentação

Os campos e seus respectivos conteúdos são os seguintes:

- **Cabeçalho**: indica o pacote.
- **Título**: título da função.
- **Description**: descrição do que o objeto é/faz.
- **Usage**: como usar ou fazer a chamada.
- **Arguments**: quais os argumentos formais da função.
- **Value**: o que a função retorna.
- **Details**: detalhes adicionais de implementação.
- **Note**: notas adicionais sobre uso e afins.
- **See Also**: referências para documentação relacionada.
- **References**: referências bibliográficas.
- **Authors**: autores da função.
- **Examples**: exemplos de uso.

Funções de ajuda

- O R possui uma documentação completa e com diversas opções para acesso.
- Não se preocupe em decorar comandos.
- Quando necessário, saiba onde consultar!
- Com o tempo e experiência acabamos nos habituando com diversos comandos.
- Mas crie o hábito de acessar documentação interna e também pesquisar na web "como fazer… no R".
- Pesquisas em inglês aumentam a chance de êxito.

Arquivos da linguagem

Arquivos da linguagem

- Ao usar uma sessão R existem alguns arquivos que são gerados.
- Mencionaremos 2 que costumam ser bastante úteis: **.Rhistory** e **.RData**.
- **.Rhistory**: salva todos os comandos executados em uma sessão R. Ele é cirado por default ao abrir uma sessão e caso exista um .Rhistory no atual diretório de trabalho ele é carregado automaticamente.
- **.RData**: durante uma sessão podemos criar muitos objetos. Estes objetos podem ser importantes, de difícil obtenção ou até mesmo custosos em termos de tempo e tamanho. O .RData serve para salvar os objetos de uma sessão. Ao abrir uma nova sessão e carregar o .RData, todos os objetos estarão lá e não precisarão ser gerados novamente.

Arquivos da linguagem

```r
# Salvando o histórico de comandos
savehistory(file = "historico.Rhistory")

# Salvando todas as variáveis criadas na sessão
save.image(file = "variaveis.RData")
```

Pacotes

Pacotes

- No R a principal forma de contribuição da comunidade é por meio de pacotes.
- Os pacotes são coleções de funções e/ou conjuntos de dados organizados e documentados.
- Um pacote R pode conter código R e também de outras linguagens como C, Fortran e C++.
- Alguns pacotes podem depender de bibliotecas do seu sistema operacional, as chamadas libs.

Pacotes

- Existem repositórios oficiais como o CRAN, Biocondutor e MRAN.
- O CRAN o mais famoso e usual.
- A instalação do pacote é feita usando a função `install.packages()`.
- Outra possibilidade é obter arquivos compactados (em geral .tar.gz) e fazer uma instalação manual.
- Outra fonte de pacotes são os repositórios de próprios desenvolvedores em plataformas de versionamento de código como o Git, GitHub, GitLab, entre outros.

Pacotes

- Por exemplo, podemos instalar o pacote `ggplot2` pelo próprio R.

```
# instalando o pacote ggplot2 do CRAN
install.packages("ggplot2")
```

- Para as funções de um pacote poderem ser usadas precisamos carregar o pacote com a função `library()`.

```
library(ggplot2)
```

- Podemos analisar o conteúdo e a documentação de um pacote.

```
ls("package:ggplot2") # conteúdo do pacote
help(package = "ggplot2") # documentação do pacote
```

Links

- The Comprehensive R Archive Network.
- RStudio Desktop.
- Posit Cloud.

Materiais para aprender R

- Mayer, FP; Bonat, WH; Zeviani, WM; Krainski, EK; Ribeiro Jr., PJ. Estatística Computacional com R. DEST/UFPR, 2018.
- Ribeiro Jr., PJ. Introdução ao Ambiente Estatístico R. 2011.
- Horton, NJ; Pruim, R; Kaplan, DT. A Student's Guide to R. 2015.
- Maindonald, JH. Using R for Data Analysis and Graphics. 2008.
- Paradis, E. R for Beginners. 2005.

Referências

PENG, Roger D. R programming for data science. Victoria, BC, Canada: Leanpub, 2016.

IHAKA, Ross. R: Past and future history. Computing Science and Statistics, v. 392396, 1998.

Microsoft R Application Network. What is R?

The R Project for Statistical Computing. What is R?