



Introdução ao R

Cristiane Hayumi Taniguti
Fernando Henrique Corrêa

Esse minicurso tem história. Agradecemos!

- 8 vezes ministrado (desde 2015)
- Autores e Monitores de várias áreas do Departamento de Genética da ESALQ-USP
- Grupos de extensão [GENt](#) e [GVENCK](#)





Por que o R?





O que é o R?

- Uma linguagem e ambiente para análises estatísticas e gráficos
- Um projeto GNU
- Provinda da linguagem S
- Desenvolvida no Bell Laboratories pelo John Chambers e colegas (telefones, fax, transistores, LEDs, C)
- Oficialmente: “Preferimos pensar que é um ambiente dentro do qual estatísticas são implementadas”

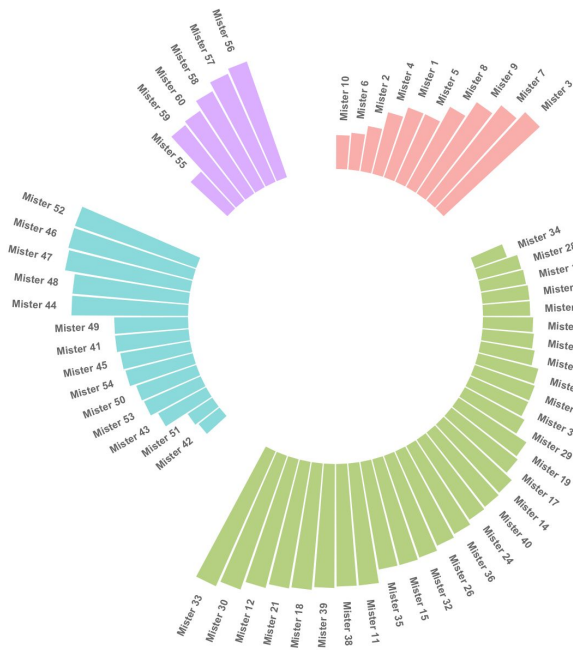
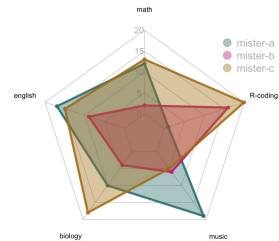
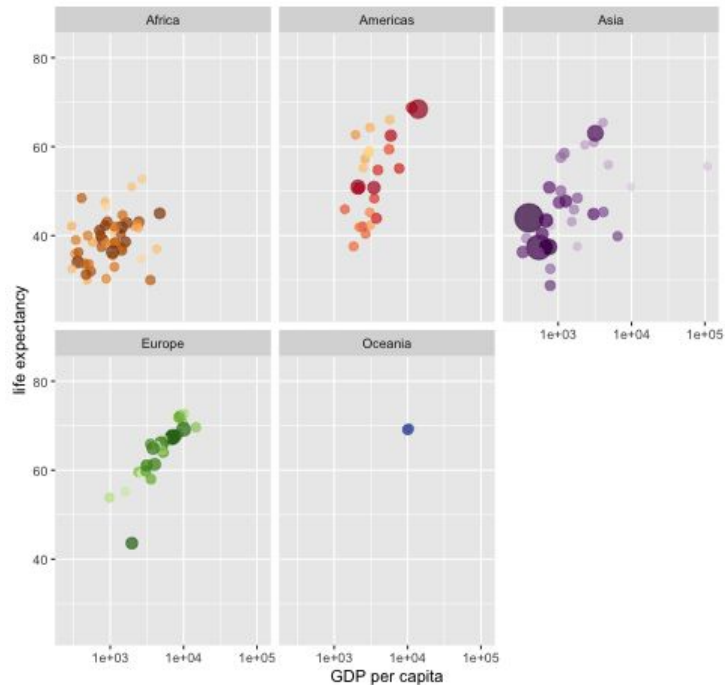


Algumas capacidades

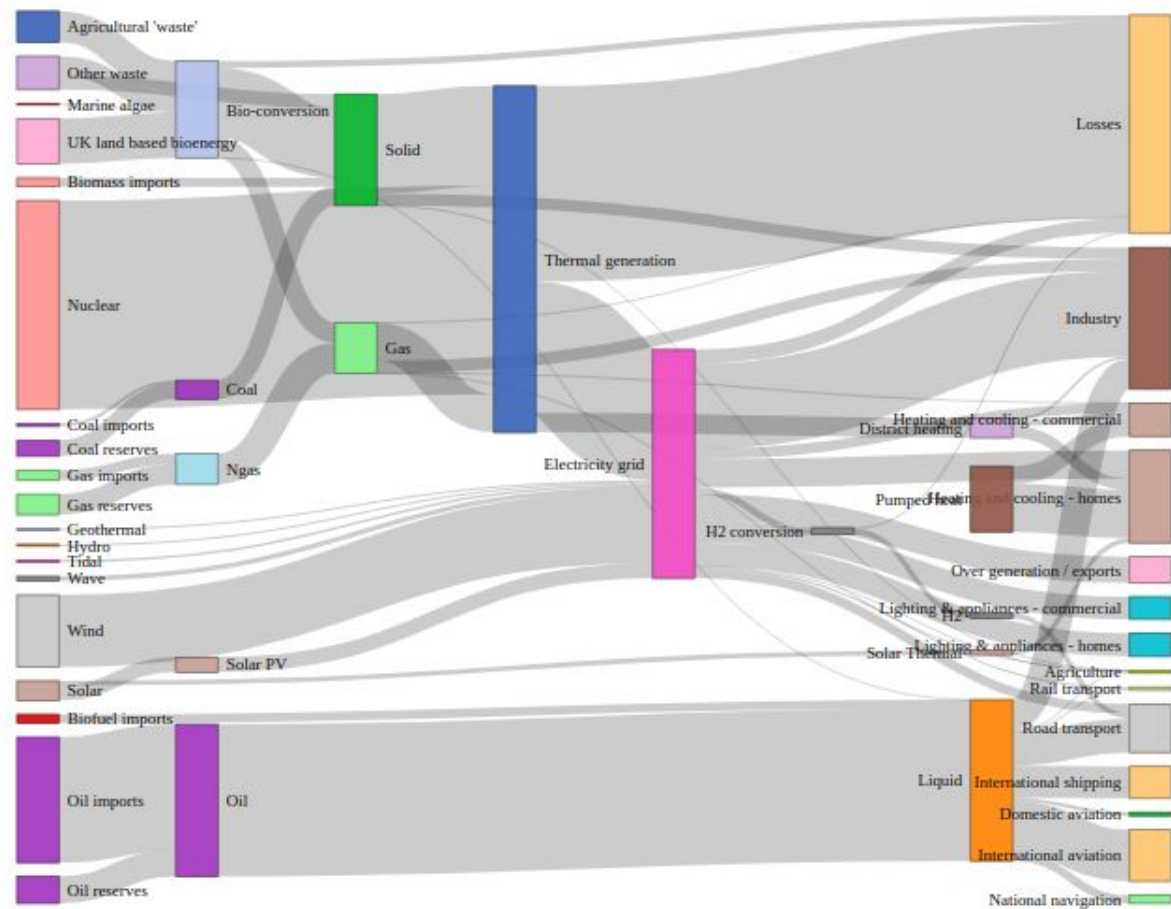
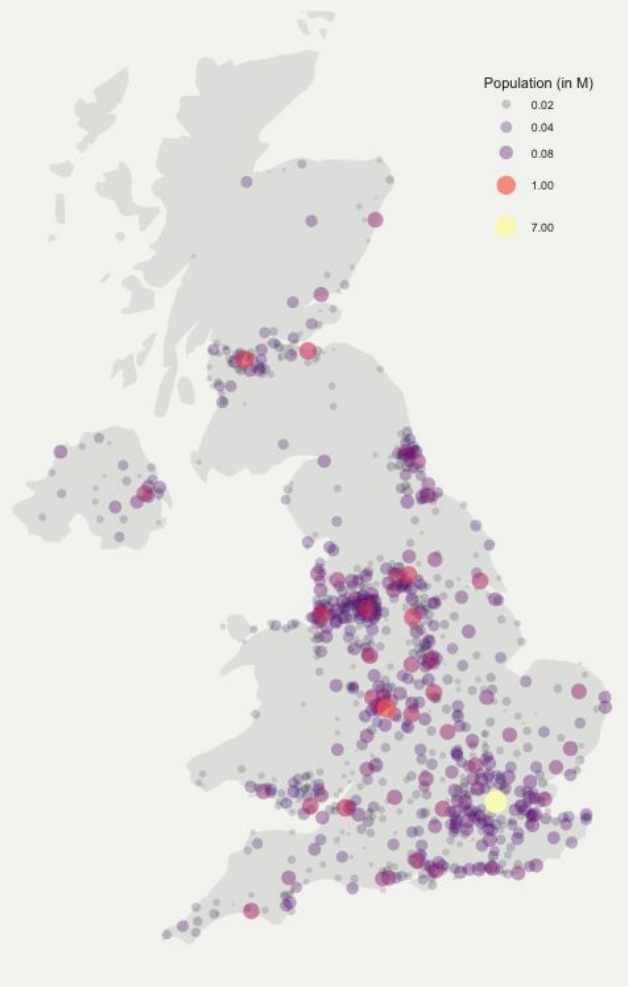
- Compatibilidade com qualquer plataforma
- Várias possibilidades de manipulação de dados
- Processamento rápido (pode ser vinculado com C, C++ e Fortran)
- Lida bem com Big Data
- Automação e reprodutibilidade
- Relatórios e Apresentações
- Estado da arte em gráficos ([belos exemplos](#))

R graph gallery

Year: 1952



The 1000 biggest cities in the UK





Um projeto GNU

Free Software Foundation's GNU general license



[ref](#)

- Rodar o programa para qualquer propósito
- Estudar como o programa funciona e adaptar isso aos seus interesses
- Distribuir cópias
- Aprimorar o programa e liberar para a comunidade acesso ao código fonte

<https://www.r-project.org/about.html>



Não é só pelo dinheiro!

“Software livre é referente à liberdade, não ao preço. Para entender o conceito, você deve pensar em “livre” como “liberdade de expressão”, não como “open bar”.

“Tira o poder dos desenvolvedores sobre os usuários”





Não é só pelo dinheiro!

No R são possíveis infinitas combinações

- Extensível (pacotes):

[CRAN](#) (15381 pacotes em 15.03.2020 - 16446 pacotes em 28.10.2020)

[Bioconductor](#) (1823 em 15.03.2020 - 1903 pacotes em 28.10.2020)

[R-forge](#) (2123 pacotes em 28.10.2020 - 2137 pacotes em 28.10.2020)

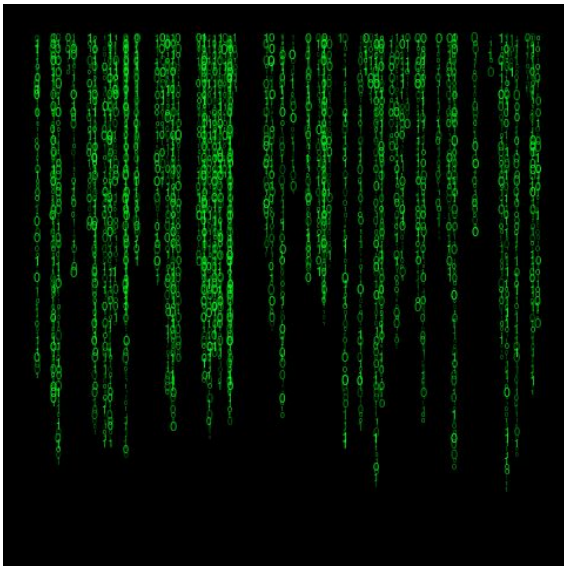
Github



Uma linguagem de programação

Nível de abstração das linguagens

Mais baixo o nível, mais liberdades e mais difícil a aprendizagem



Feito em R

Por que aprender a programar?

- Divergência entre capacidade tecnológica e aplicação

[TED talk: You should learn to program: Christian Genco at TEDxSMU](#)



- Excesso de trabalho repetitivo
- 65% das crianças terão empregos no futuro que ainda não existem (World Economic Forum)



Reprodutibilidade

Crise na reprodutibilidade de ciência

- [Manifesto pela ciência reprodutível \(nature human behavior\)](#)
- [Desafio para pesquisadores rodarem seu código novamente depois de 10 anos](#) (nature)
- Outra sugestão de ferramenta para reprodutibilidade: [Docker](#)



O poder da colaboração



[Tutorial sobre o git](#)

[Tutorial de como integrar git e RStudio](#)

Alguns exemplos:

- [Compartilhamento de dados sobre impacto da COVID-19](#) - 7 contribuidores
15.5k forks
- [Compartilhamento de dados sobre SARS-CoV-2](#) - 16 contribuidores 97 forks
- [Trabalho sobre partículas elementares Higgs Boson \(2012\)](#) - 875 contribuidores
3.4k forks
- [OneMap](#) - 9 contribuidores



Incentivos

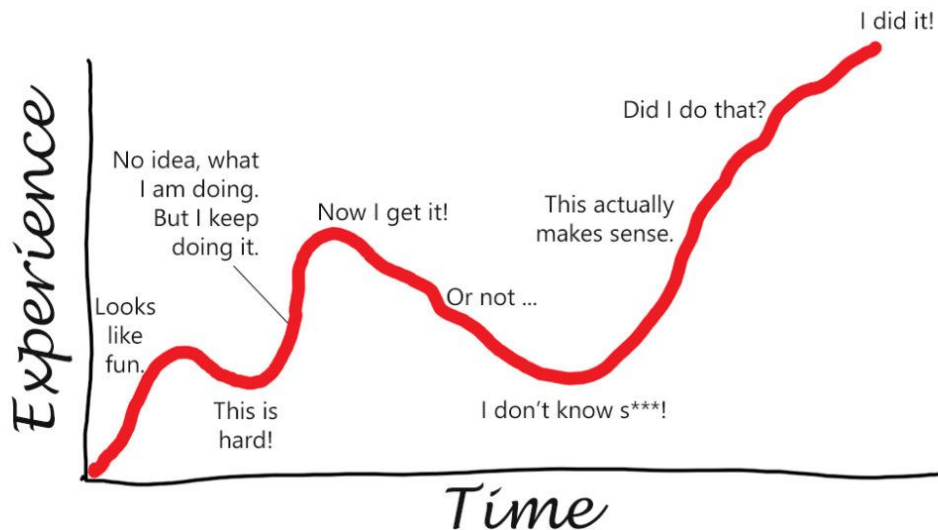
Lógica de programação:

- [Hour of Code](#)
- [Scratch](#)
- [Coursera](#)
- [Khan academy](#)
- [Code academy](#)
- [edX](#)
- [Github](#)

Para o R:

- Manuais do [CRAN](#)
- Pacote iterativo [Swirl](#)
- Canal do Youtube [The New Boston](#)
- Cursos do [Datacamp](#)
- Série de tarefas no [TryR](#)

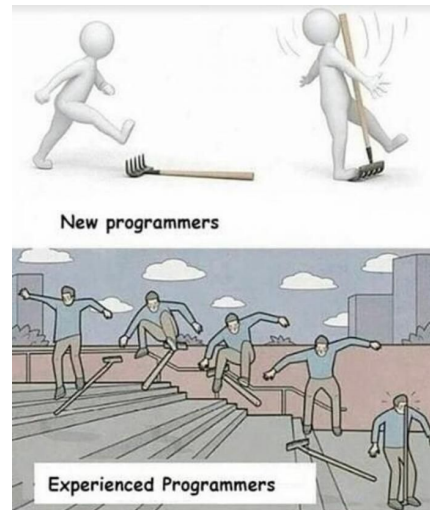
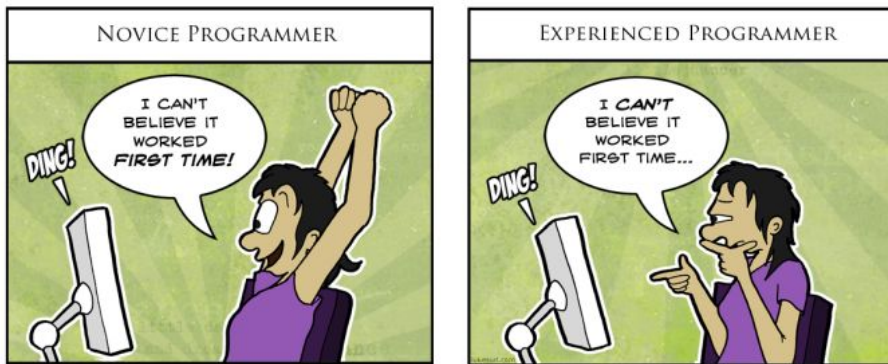
Como aprender a programar?



Frustrações

Bravura x perfeição

TED talk: [Teach girls bravery, not perfection | Reshma Saujani](#)



Referências: imagem
[direita](#) e [esquerda](#)



Dicas para combate dos “bugs”

Termo criado por [Grace Hopper](#) 1947

- Vá aos poucos
- Identifique a linha do código onde está o erro
- Google!
- Use o método científico: levante hipóteses e teste
- Deixe o erro facilmente reproduzível



Pratique!

TED talk: [Coding, creativity, and the future of work | Jason Madar](#)

Programação é uma ferramenta para expressar criatividade

Exercite o R (ou outra linguagem) no seu dia-a-dia:

- Use em disciplinas mesmo que não exigido (relatórios em [Rmarkdown](#) são muito bonitos e práticos)
- Invista tempo e fuja de programas “de cliques”



Motivações

- TED talk: [Mitch Resnich: Let's teach kids to code](#)
- TED talk: [My experience of Using the Statistical Language R | Nathan Hadjiyski](#)
- [Hour of code 1](#)
- [Hour of code 2](#)
- [Incentivo Obama](#)



Então vamos começar!

Acesse:

bit.ly/R_courses