

Do script ao pacote de R

um exemplo desde a biologia

Andrea Sánchez-Tapia
Núcleo de Computação Científica e Geoprocessamento
Jardim Botânico do Rio de Janeiro

R-Ladies Rio de Janeiro



Do script ao pacote de R

um exemplo desde a biologia

Andrea Sánchez-Tapia
Núcleo de Computação Científica e Geoprocessamento
Jardim Botânico do Rio de Janeiro

R-Ladies Rio de Janeiro

Apresentação

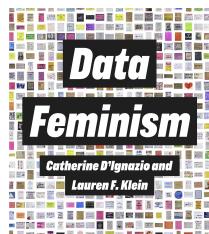


- Bióloga - Universidade Nacional da Colômbia
- Mestre em Ecologia - UFRJ
- Doutora em Botânica - JBRJ
- Pós-doc - Núcleo de Computação Científica e Geoprocessamento do JBRJ

Ecologia de comunidades vegetais, restauração ecológica, ecologia quantitativa

Informática da biodiversidade, modelagem de nicho ecológico

Ciência aberta e reproduzível, ética na ciência de dados, ciência de dados feminista



Catherine D'Ignazio & Lauren Klein <http://datafeminism.io/>

Apresentação



- Como trabalhar apenas com software *libre*?
- Usuária de **R** desde 2009
- R-Ladies Rio de Janeiro desde 2017
- **O alvo hoje:** Executar e ensinar a fazer projetos de análise reprodutíveis. Desde o *download* e processamento de dados até produzir o manuscrito, relatório ou apresentação (**rmarkdown, knitr**)

Curso Projetos de análise de dados usando R



Sara Mortara (R-Ladies Rio de Janeiro)



Boas práticas em análise de dados



Por que pensar em pacotes?



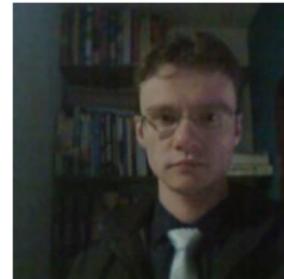
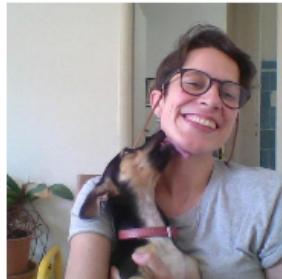
- Filosofia de R: transição de usuárie a programadore
- Filosofia da **R Foundation** e de **R Forwards**: mais raças, gêneros, países, línguas representados na comunidade de R ([useR! 2020 panel](#))
- Úteis para compartilhar grandes conjuntos de funções que têm um propósito similar
- Grande experiência de aprendizado: portabilidade, transparência, robustez, comunicação.

Habilidades que vale muito a pena aprender *bem antes de chegar ao pacote*

Pacote de R para modelagem de nicho ecológico: **modleR**



- **Unificar** diferentes partes do processo de MNE
- Fornecer **metadados** e **documentar** decisões de parametrização
- Se **integrar** ao resto de ferramentas que existem no ambiente de R



Marinez F Siqueira, Sara Mortara, Diogo Rocha, Guilherme Gall, Felipe Sodré

 <https://model-r.github.io/modleR/>

 Aula no curso ENM-2020

modleR: a workflow for ecological niche models

modleR is a workflow designed to automatize and document some of the common steps when performing ecological niche models (ENM). Given the occurrence records and a set of environmental predictors, `setup_sdmdata()` prepares the data by cleaning for duplicates, removing occurrences with no environmental information and applying some geographic and environmental filters. It also partitions data into test and training sets, using crossvalidation or bootstrap procedures. `do_any()` or `do_many()` fit the ecological niche models using several algorithms, some of which are already implemented in the **dismo** package (Hijmans et al 2017), and others come from other packages in the R environment, such as `glm`, Support Vector Machines (`kernlab` and `e1071`) and Random Forests (`randomForest`). A function to join individual partitions in several ways is provided in `final_model()`. Finally, `ensemble_model()` assembles models from distinct algorithms and provides summary rasters.



New Results

Comment on this paper

modleR: a modular workflow to perform ecological niche modeling in R

Andrea Sánchez-Tapia, Sara Ribeiro Mortara, Diogo Souza Bezerra Rocha, Felipe Sodré Mendes Barros, Guilherme Gall, Martinez Ferreira de Siqueira

doi: <https://doi.org/10.1101/2020.04.01.201105>

This article is a preprint and has not been certified by peer review [what does this mean?].

Abstract

Full Text

Info/History

Metrics

Preview PDF



Modelos de nicho ecológico ou de distribuição das espécies (ENM, SDM)

- Dados de ocorrência das espécies: **objetos espaciais**, bases de dados, limpeza de dados.
- Camadas preditoras ambientais (**rasters** de SIG) Pacotes **raster, sp, maps, rgdal**: (taskview Spatial)
- Abordagem de aprendizado estatístico
- Diferentes algoritmos
- **Inferência de áreas adequadas para a ocorrência das espécies**

A história

O trabalho: gerar uma camada de diversidade potencial de plantas da Mata Atlântica



- A "maior" quantidade possível de espécies
- Muito controle sobre a qualidade dos dados e dos modelos
- Correção taxonômica, limpeza dos registros
- Alta resolução (30s ~1km)

O fluxo de trabalho: cada espécie precisa passar por várias etapas

- Obtenção, **limpeza** de dados
- Desenho experimental, **preparação** dos dados
- Ajustar os **modelos, projetar** os modelos para vários algoritmos
- Juntar os resultados dos diferentes algoritmos
- Juntar o resultado para várias espécies



A gente já conhecia algumas boas práticas

Boas práticas ❤️ estrutura de pastas e pasta de trabalho



- Meetup anterior! 🎥 Apresentação e 💬 Tutorial
- Não use `setwd()`! 🔥
- Não mude de pasta de trabalho ao longo do projeto: use **caminhos relativos**
 - Caminhos absolutos:
`"C://Eu/Minha_pasta/arquivo_meu_v_13.csv"`
 - Caminhos relativos: `"../data", "./figs"`
- Cuide da estrutura de pastas



Os scripts

Primeira etapa: scripts soltos



- Um script para **uma** espécie (fazia todas as etapas de vez)
- Um "**for loop**" para fazer para várias espécies

```
lista_de_especies <- c("sp1", "sp2", "sp3")
```

```
for (especie in lista_de_especies) { # pode ter qualquer nome, i
  todo_o_codigo(especie)
}
```

```
especie <- sp1
todo_o_codigo(sp1)

especie <- sp2
todo_o_codigo(sp2)

especie <- sp3
todo_o_codigo(sp3)
```

Criar funções

"Se você está copiando e colando seu código mais de três vezes, é hora de escrever uma função"



Hadley Wickham

```
modelar <- function(x) {  
  y <- ... (x)  
  return(y)  
}  
  
# Uma espécie  
modelar(especie)  
  
# Várias espécies  
lapply(lista_de_especies, modelar)  
  
purrr::map(lista_de_especies,  
           ~modelar)
```

A função ainda rodava todas as etapas da análise para cada espécie!

Rodava!

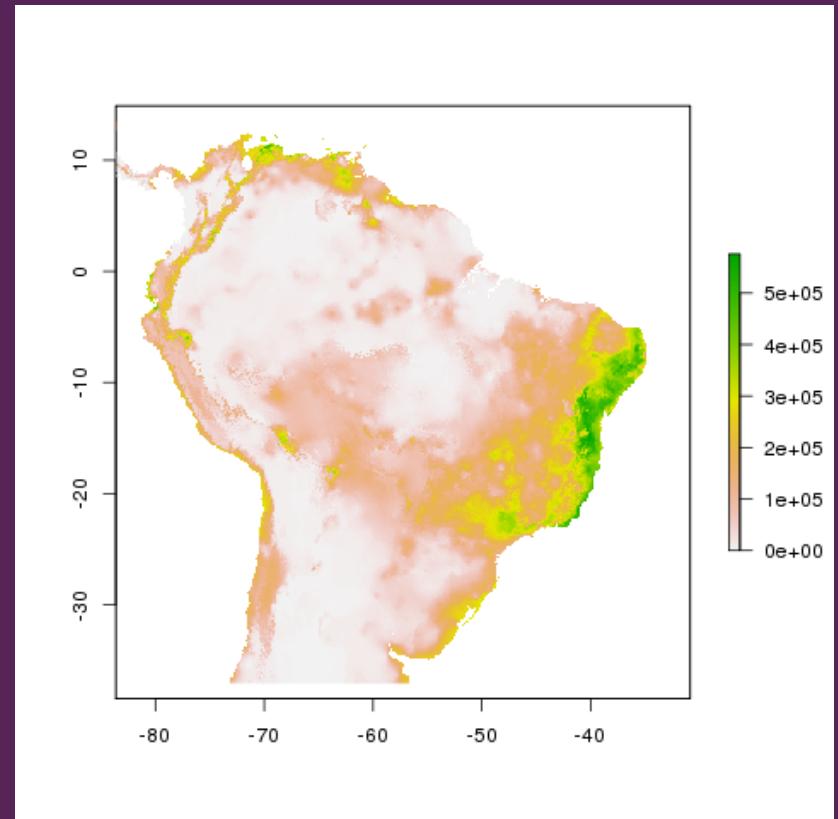
Matava a RAM :/

Demorava 10 dias ⇢

Fazia uma espécie após
a outra



Felipe Sodré



Melhorando o desempenho



Laboratório
Nacional de
Computação
Científica

- Capacidade de computação
- Computação remota (VPN, ssh)
- **Paralelização**
- **Modularização:** Separar os quatro processos em funções diferentes
- **Colaboração:**
 - Cuidar do **estilo de código**
 - **Começar a usar controle de versões** (git/GitHub)



Paralelização

- Cada processo vai para um núcleo do computador (*core*)
- Os *clusters* de computação podem ter várias centenas de núcleos
- Há várias maneiras de paralelizar processos em R

```
library(parallel)
library(snowfall)
library(foreach)
library(futures)
```

```
sfInit(parallel = T, cpus = 3)
sfExportAll() # para que cada núcleo receba as variáveis do workspace
sfLapply(lista_de_especies, modelar)
sfStop()
```

Passou de 10 dias para algumas horas --



Modularizar o código

- Cada repetição ainda executava os quatro passos
- E se a gente rodasse um processo por vez para todas as espécies?

Antes

```
{especie1 A, B, C, D} {especie 2 A, B, C, D} ...n
```

Depois

```
{passo A sp1, sp2, sp3,...spn} {passo B sp1, sp2, sp3,...spn} {passo C sp1, sp2, sp3,...spn} {passo D sp1, sp2, sp3,...spn}
```

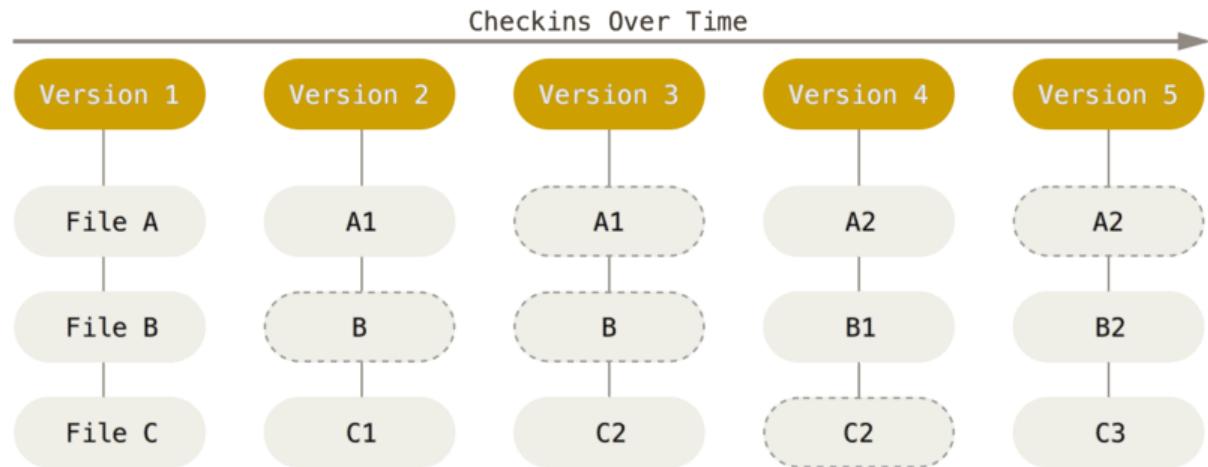
- Na hora de repetir não precisava voltar até o início
- Quatro funções! Não UMA!

Colaboração e controle de versões



- Os pacotes de R podem ser instalados diretamente desde GitHub!

Git: baseado em commits



<https://git-scm.com/book/en/v2/Getting-Started-About-Version-Control>

Git: modo diff



R/create_buffer.R

Hunk 1 : Lines 30-45

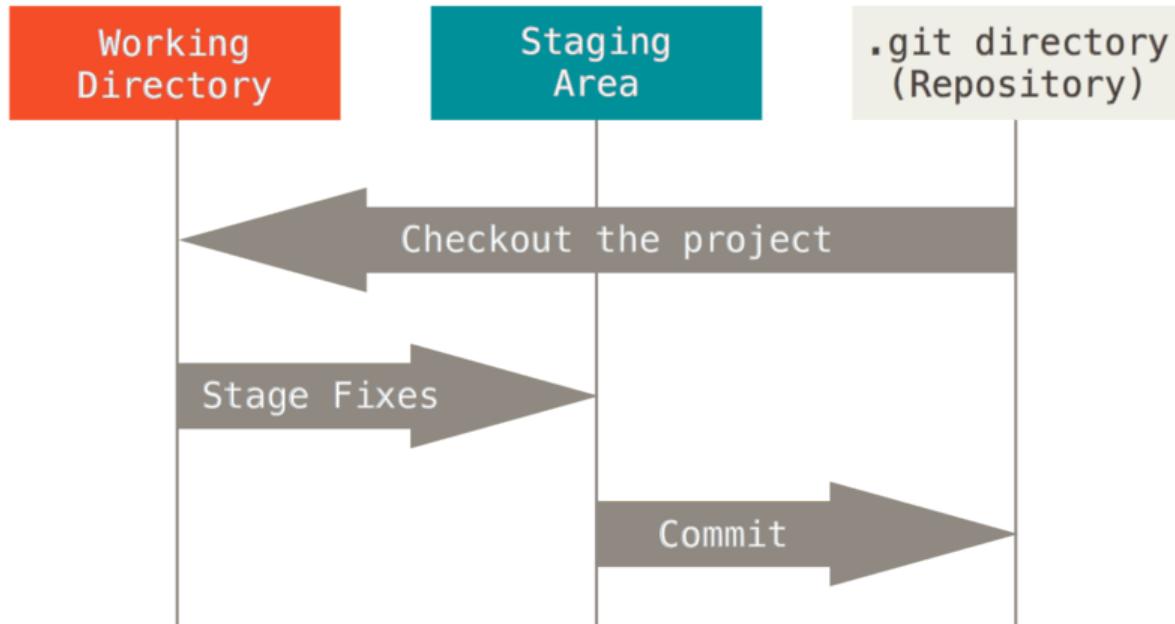
```
30 30     create_buffer <- function(occurrences,
31 -         buffer_type,
31 +         buffer_type = NULL,
32 32             predictors,
33 33             dist_buf = NULL) {
34 +     sp::coordinates(occurrences) <- ~lon + lat
35 +     raster:::crs(occurrences) <- raster:::crs(predictors)
36 +     if (is.null(buffer_type) | length(setdiff(buffer_type, c("distance", "mean", "median", "max")) > 0)) {
37 +         warning("buffer_type NULL or not recognized, returning predictors")
38 +         r_buffer <- predictors
39 +         return(r_buffer)
40 +     }
41     if (buffer_type %in% c("distance")) {
42 -         dist.buf <- distance
42 +         if (is.null(dist_buf)) stop("dist_buf must be set when using a distance buffer")
43 +         else dist.buf <- dist_buf
44 } else if (buffer_type %in% c("mean", "median", "max")) {
45 -     sp::coordinates(occurrences) <- ~lon + lat
46 -     #raster:::crs(occurrences) <- raster:::crs(predictors)
47     dists <- rgeos:::gDistance(spggeom1 = occurrences, byid = T)
```

Hunk 2 : Lines 60-61

```
54 60     r_buffer <- raster:::mask(r_buffer, buffer.shape)
55 -     if (is.null(buffer_type) | length(setdiff(buffer_type, c("distance", "mean", "median", "max")) > 0)) {
56 -         warning("buffer_type not recognized, returning predictors")
57 -         r_buffer <- predictors
58 -     }
59 61     return(r_buffer)
```

<https://git-scm.com/book/en/v2/Getting-Started-Git-Basics>

Git: local e remoto



<https://git-scm.com/book/en/v2/Getting-Started-Git-Basics>



- Backup
- Controle de versões
- Reprodutibilidade
- Workflow colaborativo: **issues** para perguntar e dar feedback, **Pull Requests** para sugerir mudanças
- Transparência

É preciso padronizar o estilo de código!*

Estilo de código

"Um bom estilo de código é como usar uma boa pontuação.
Você pode viver sem ela mas sem dúvida facilita ler as coisas."



Hadley Wickham

- Google's R style guide
- tidyverse
- Advanced R

Independente do estilo, o que importa é ser consistente!

Ctrl+Shift+A Reformata o código automaticamente

pacote **formatR**

A transformação em pacote



Guilherme Gall (LNCC) + Hadley Wickham

- Estrutura especial das pastas e arquivos especiais
- Documentação específica
- Licença de uso
- "Vignettes" e documentação adicional
- Passar testes!
- Publicar (Submissão a CRAN? #medo)



Estrutura de um pacote de R

```
.  
├── R/          # Funções  
├── man/        # Documentação  
├── data/       # Dados  
├── vignettes/ # Vignettes  
├── inst/       # Os manuais que ficam quando você instala  
└── .buildignore # Arquivos e pastas que não serão controlados  
├── DESCRIPTION # Descrição do pacote  
└── NAMESPACE   # NAMESPACE  
└── README.md   # README
```

- O próprio pacote pode ser criado usando as opções de RStudio
- O pacote **usethis**: `usethis::create_package()`
- Alguns elementos são obrigatórios (DESCRIPTION, NAMESPACE, R/, man/)

DESCRIPTION

```
.  
├── R/          # Funções  
├── man/        # Documentação  
├── data/       # Dados  
├── vignettes/ # Vignettes  
├── inst/       # Relatórios reprodutíveis a partir dos outputs  
└── .buildignore # Arquivos e pastas que não serão controlados  
   └── DESCRIPTION # Descrição do pacote  
   └── NAMESPACE  # NAMESPACE  
   └── README.md   # README
```

A DESCRIPTION do pacote é editada a mão.

Package: rladiesrio

Type: Package

Title: Isto é um esqueleto de pacote de teste

Version: 0.1.0

Author: RLadies+ Rio de Janeiro

Maintainer: Andrea Sánchez-Tapia <andreasancheztapia@gmail.com>

Description: Este esqueleto permite entender a estrutura de um pacote.

License: MIT

Encoding: UTF-8

LazyData: true

NAMESPACE

```
.  
└── R/          # Funções  
└── man/        # Documentação  
└── data/       # Dados  
└── vignettes/ # Vignettes  
└── inst/       # Relatórios reprodutíveis a partir dos outputs  
└── .buildignore # Arquivos e pastas que não serão controlados  
└── DESCRIPTION  # Descrição do pacote  
└── NAMESPACE   # NAMESPACE  
└── README.md    # README
```

O NAMESPACE indica quais funções serão **importadas** ou **exportadas** pelo pacote

```
# Generated by roxygen2: do not edit by hand
```

```
export(create_buffer)  
export(do_any)  
import(graphics)  
import(raster)  
importFrom(Rdpack,reprompt)  
importFrom(dismo,randomPoints)
```

R e data

```
•
  └── R/          # Funções
  └── man/        # Documentação
  └── data/       # Dados
  └── vignettes/ # Vignettes
  └── inst/       # Relatórios reprodutíveis a partir dos outputs
  └── .buildignore # Arquivos e pastas que não serão controlados
  └── DESCRIPTION  # Descrição do pacote
  └── NAMESPACE   # NAMESPACE
  └── README.md    # README
```

- As **funções** estão na pasta **R/**
- A documentação das funções estará em **man/**
- Os **dados** estarão na pasta **data/** (`data(cars)`)

roxygen2 e a documentação das funções

```
.  
├── R/          # Funções  
└── man/        # Documentação  
├── data/       # Dados  
├── vignettes/ # Vignettes  
├── inst/       # Relatórios reprodutíveis a partir dos outputs  
├── .buildignore # Arquivos e pastas que não serão controlados  
├── .gitignore  # Arquivos e pastas que serão ignorados por git  
├── *.Rproj     # Projeto de RStudio  
├── DESCRIPTION # Descrição do pacote  
├── NAMESPACE   # NAMESPACE  
└── README.md   # README
```

- Antes era preciso escrever à mão a documentação e o NAMESPACE
- Hoje: as funções podem ser documentadas diretamente nos arquivos .R em R/
- O pacote **roxygen2** vai transformar esses comentários em arquivos .Rd da documentação

roxygen2 e a documentação das funções

```
#' Titulo da funcao
#'
#' descricao
#'
#' @param parametro1 descrição do parâmetro1
#' @param parametro2 descrição do parâmetro2
#' @Import pacote
#' @ImportFrom pacote funcao funcao funcao
#' @details
#' @returns um data.frame
#' @export #para que entre no NAMESPACE
#' @examples
#' funcao <- function(x) {
#'   funfunfun
#' }
```

funcao <- function(x) { ...

- Opção **code > Insert Roxygen Skeleton** no menu de RStudio
- Quando a documentação está escrita: **devtools::document()**
- **roxygen2** vai criar os arquivos em man/ e editar o NAMESPACE

Outras formas de documentação: vignettes e README

```
•
├── R/          # Funções
└── man/        # Documentação
└── data/       # Dados
└── vignettes/ # Vignettes
└── inst/       # Relatórios reproduutíveis a partir dos outputs
└── .buildignore # Aqui tem que estar README.Rmd que é um arquivo opcional
└── DESCRIPTION # Descrição do pacote
└── NAMESPACE   # NAMESPACE
└── README.Rmd  # README em rmarkdown que é knittado para criar o md
└── README.md    # README
```

- README e vignettes são arquivos de **rmarkdown**



.buildignore

```
.
├── R/          # Funções
└── man/        # Documentação
  └── data/     # Dados
  └── vignettes/ # Vignettes
  └── inst/      # Relatórios reproduutíveis a partir dos outputs
  └── .buildignore # Aqui tem que estar README.Rmd que é um arquivo opcional
  └── DESCRIPTION # Descrição do pacote
  └── NAMESPACE  # NAMESPACE
  └── README.Rmd  # README em rmarkdown que é knittado para criar o md
  └── README.md   # README
```

- .buildignore é um **arquivo de texto** permite ignorar arquivos fora do padrão ou que ainda não estão prontos
- O ponto no início do nome faz com que o navegador de arquivos não veja: arquivo oculto
- O navegador do RStudio consegue ver!
- Editar: **README.Rmd**

Checando o pacote



- Durante o trabalho: `devtools::load_all()`
- Quando o pacote foi terminado: hora dos checks de CRAN!
`devtools::check()`
- Vai devolver NOTAS, ADVERTÊNCIAS e ERROS
- Resolver um a um. Internet, paciência, café.

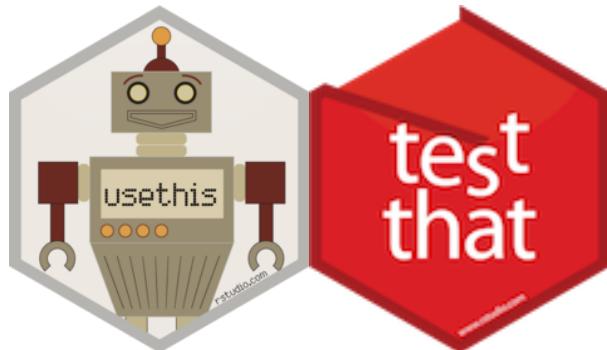
```
Status: OK

R CMD check results
0 errors | 0 warnings | 0 notes
```

- Pode mandar para GitHub e **o pacote pode ser instalado desde GitHub**

```
remotes::install_github("Model-R/modleR",
build_vignettes = TRUE)
```

Outras ferramentas para seus pacotes



- **usethis**: permite criar pacotes, usar git, organizar desde o início (**Não misture!**)
- **testthat**: criar testes com expectativas formais

Uma página para o seu pacote: pkgdown



modleR: a workflow for ecological niche models

Andrea Sánchez-Tapia, Sara Mortara & Diogo S. B. Rocha

2020-04-19

Source: vignettes/modleR.Rmd

modleR is a workflow based on package **dismo** (Hijmans et al. 2017), designed to automate some of the common steps when performing ecological niche models. Given the occurrence records and a set of environmental predictors, it prepares the data by cleaning for duplicates, removing occurrences with no environmental information and applying some geographic and environmental filters. It executes crossvalidation or bootstrap procedures, then it performs ecological niche models using several algorithms, some of which are already implemented in the **dismo** package, and others come from other packages in the R environment, such as **glm**, Support Vector Machines and Random Forests.



Installing

Currently **modleR** can be installed from GitHub:

```
# Without vignette
remotes::install_github("Model-R/modleR", build = TRUE)
# With vignette
remotes::install_github("Model-R/modleR",
                       build = TRUE,
                       dependencies = TRUE,
                       build_opts = c("--no-resave-data", "--no-manual"),
surero
```



- `pkgdown::build_site()` knittar em formato de página web e colocar em uma pasta **docs/** do pacote
- O repositório de GitHub tem que estar configurado para fazer uma página a partir de **docs/**

Do script ao pacote



- Procure **portabilidade** e **reprodutibilidade** em seus scripts
- Separe os problemas em unidades menores: resolva UMA unidade e crie **loops** e **funções** para o resto
- Invista em aprender **git** e **rmarkdown**

Do script ao pacote



- Vale a pena?
- Colaborar com pacotes existentes?
- Quantas funções, o que entra em cada fase? o que sai? Modularize
- Não faça do zero: use **roxygen2**, **devtools**, **usethis**, **testthat**
- Cuide da **documentação** e crie **vignettes** e exemplos
- Divulgue seu trabalho **#rststats**, **#rladies**
- Teste, cuide des usuáries e da comunidade ao redor do pacote ❤️

Pacotes de exemplo: coronabr e R-ladies antifa



Download de dados de COVID-19 no Brasil

coronabr é um pacote de **R** para fazer *download* e visualizar os dados dos casos diários de coronavírus (COVID-19) disponibilizados por diferentes fontes:

- [Ministério da Saúde](#);
- [Brasil I/O](#);
- [Johns Hopkins University](#)



Nosso objetivo

O nosso objetivo é facilitar o acesso aos dados de diferentes fontes, usando ferramentas de acesso aberto e que permitam reprodutibilidade.

O código é aberto. Entre em [como usar](#) para um exemplo de como utilizar o pacote. Compartilhe.

Fazemos ciência aberta, democrática e reprodutível. Este é um trabalho em desenvolvimento. Para entender como contribuir, clique [aqui](#).

Aviso!

Disponibilizamos aqui atualizações dos gráficos. Sabemos que deve haver maneiras responsáveis de apresentar estes dados e que as comparações entre regiões e países dependem de muitas variáveis.

Os dados são referentes aos casos notificados. É sabido que temos subnotificação e atraso nas notificações. Assim, mais do que nunca, não refletem a epidemia em si.

Links

Browse source code at

<https://github.com/liibre/coronabr/>

Report a bug at

<https://github.com/liibre/coronabr/issues>

License

GPL (>= 3)

Community

[Contributing guide](#)

Developers

Sara Mortara

Author, maintainer

Andrea Sánchez-Tapia

Author

Karlo Guidoni Martins

Author

[All authors...](#)

Dev status

https://github.com/liibre/r ladies_antifa



¡Obrigada!

✉ andreasancheztapia@gmail.com

🐦 @SanchezTapiaA

⌚ 💻 🔍 andreasancheztapia

⌚ R-Ladies+ Rio de Janeiro

