

Aula prática dia 1: Introdução ao ambiente R e boas práticas. Organização de projetos com R e RStudio usando Git e Github.

Tainá Rocha

2021/09/08 (updated: 2021-09-13)

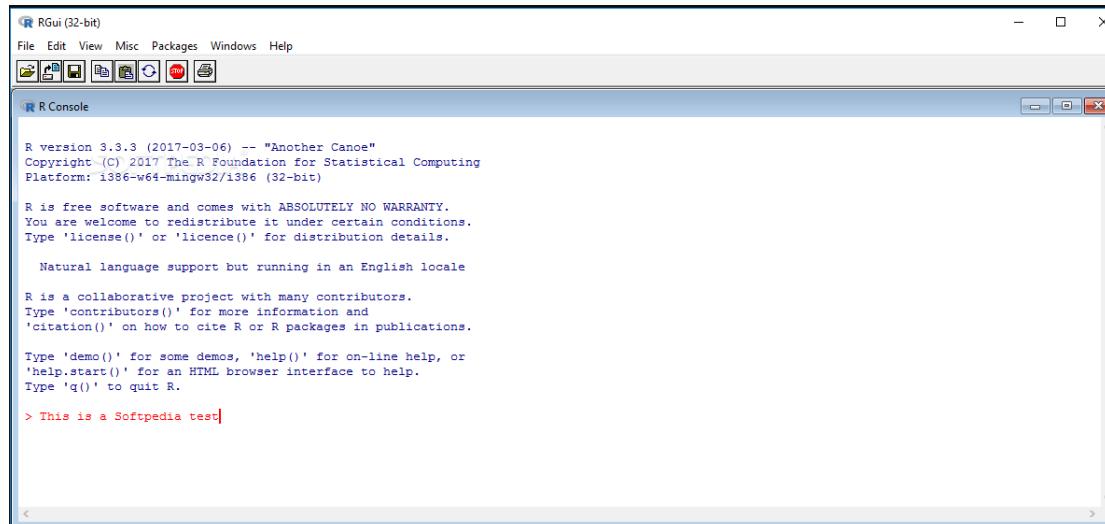
A prática é dividida em 4 momentos

- Parte 1 : Definições básicas sobre as ferramentas que usaremos
- Parte 2: Instalações e configurações
- Parte 3: Criando o repositório dos projetos
- Parte 4: Organização dos projetos

Parte 1

Linguagem R

Linguagem de programação de código aberto voltada principalmente para realização de análises estatísticas e construção de gráficos. R pode ser rodado em qualquer terminal (linux,mac, windows). A interface original de R é um console onde o código é escrito.



Parte 1

Linguagem R

Vantagens

- Open Source. Pode ser usado e modificado para qualquer fim
- Pode ser usado em qualquer sistema operacional
- Variedade de pacotes. Mais de 20 mil pacotes à nossa disposição e assim não precisamos "reinventar a roda"
- Comunidade forte e colaborativa. Existe muito material disponível

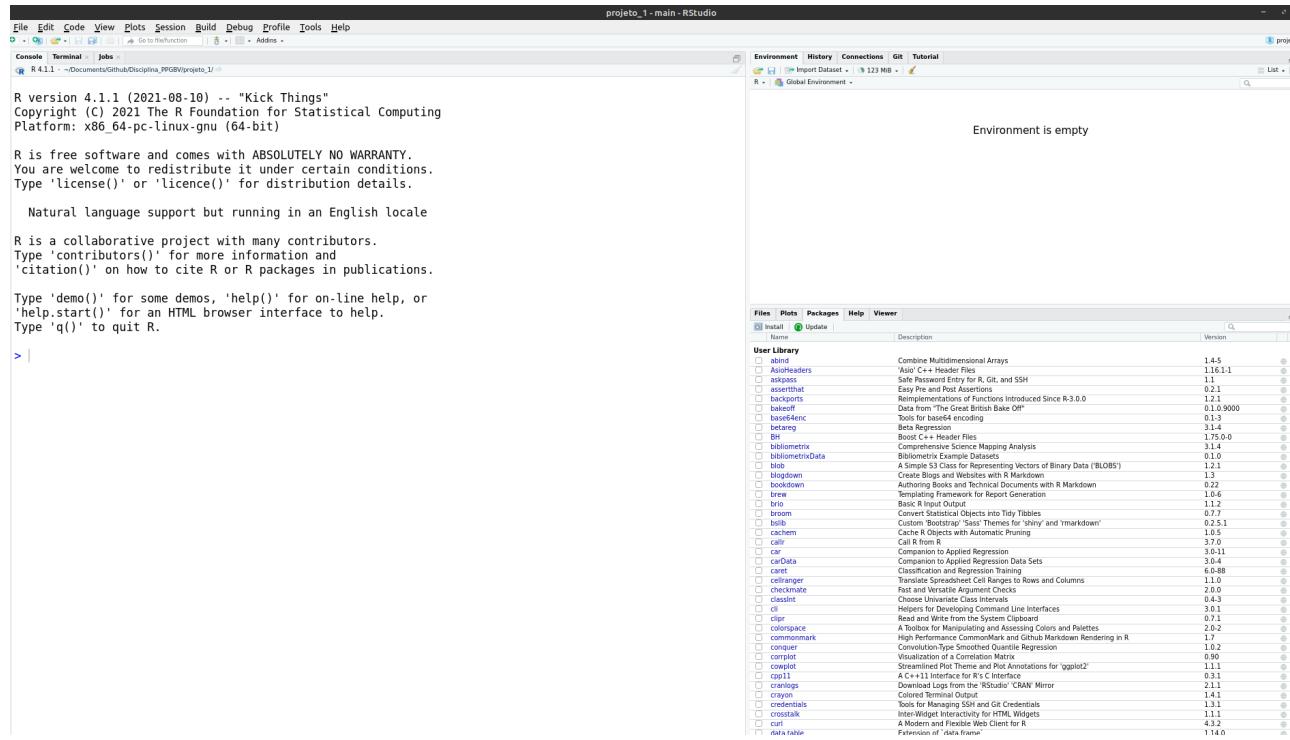
Desvantagens

- Curva de aprendizado mais demorada
- Gerenciamento de memória e velocidade ainda são alguns desafios comparados a outras linguagens de programação

Parte 1

RStudio

Uma interface gráfica para R.



Parte 1



É o sistema de controle de versões mais usado. Ele registra todas as alterações, e cira assim um histórico dessas alterações. Isso pode ser feito com quase todos os tipos de arquivos (.txt, .csv etc...), inclusive arquivos .R

Todo comando de git no terminal começa por **git**. Os principais comandos que vamos usar

git commit : para adicionar arquivos

git push : colocar nûvem as alterações

git pull : para baixar para seu computador as atualizações que estão na nûvem (no nosso caso no Github).

Isso será melhor entendido mais adiante.

Parte 1

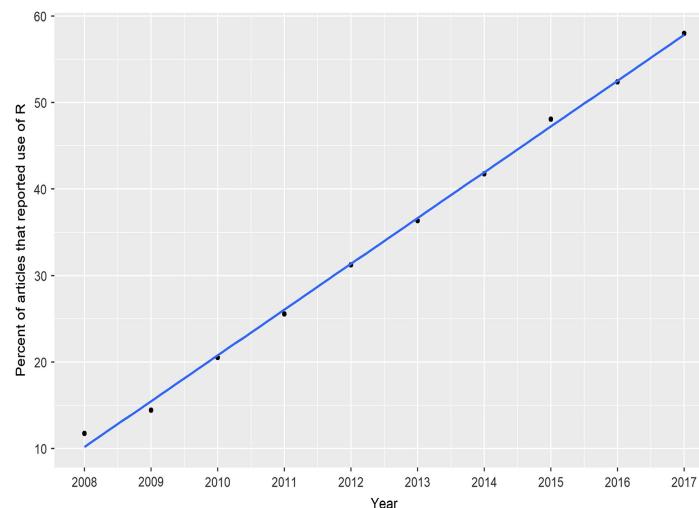


GitHub é um serviço baseado em nuvem que hospeda os arquivos que usam Git como controle de versão. Além de várias funcionalidades (como criar webpages), ele permite que os usuários colaborem e façam mudanças em projetos compartilhados enquanto mantêm um registro detalhado do progresso do seu código.

Parte 1

E qual relação dessas ferramentas com Modelagem de Nicho Ecológicos (ENMs) e Distribuição Potencial de Espécies (SDMs) ?

A popularização do uso da linguagem R, principalmente na ecologia é relativamente recente. Veja este gráfico de Lai et al. 2019 do artigo "Evaluating the popularity of R in ecology"



Parte 1

E qual relação dessas ferramentas com Modelagem de Nicho Ecológicos (ENMs) e Distribuição Potencial de Espécies (SDMs) ?

nature ecology & evolution

Explore content ▾ About the journal ▾ Publish with us ▾

nature > nature ecology & evolution > perspectives > article

Perspective | Open Access | Published: 23 September 2019

A checklist for maximizing reproducibility of ecological niche models

Xiao Feng  Daniel S. Park, Cassandra Walker, A. Townsend Peterson, Cory Merow & Monica Papse

Nature Ecology & Evolution 3, 1382–1395 (2019) | Cite this article

13k Accesses | 32 Citations | 57 Altmetric | Metrics

Abstract

Reporting specific modelling methods and metadata is essential to the reproducibility of ecological studies, yet guidelines rarely exist regarding what information should be noted. Here, we address this issue for ecological niche modelling or species distribution modelling, a rapidly developing toolset in ecology used across many aspects of biodiversity science. Our quantitative review of the recent literature reveals a general lack of sufficient information to fully reproduce the work. Over two-thirds of the examined studies neglected to report the version or access date of the underlying data, and only half reported model

Parte 1

E qual relação dessas ferramentas com Modelagem de Nicho Ecológicos (ENMs) e Distribuição Potencial de Espécies (SDMs) ?

FRONTIERS IN ECOLOGY *and the* ENVIRONMENT

Write Back

Doubling demands in programming skills call for ecoinformatics education

Xiao Feng✉, Huijie Qiao, Brian J Enquist

First published: 01 April 2020 | <https://doi.org/10.1002/fee.2179> | Citations: 1

[Read the full text >](#)

 PDF  TOOLS  SHARE

Parte 1

E qual relação dessas ferramentas com Modelagem de Nicho Ecológicos (ENMs) e Distribuição Potencial de Espécies (SDMs) ?

Você pode fazer suas análises através de códigos R, registrando todas alterações nesses códigos. Seus códigos e histórico de alterações podem ser hospedado pelo Github. Usar essas ferramentas integradas e da maneira correta é uma boa prática que garante maior reproduzibilidade de suas análises e facilita trabalhos feitos em equipe quando mais de uma pessoa trabalha no mesmo código.

Parte 1

Vamos ao que interessa!

Essa é uma disciplina de teoria e prática de ENMs e SDMs, portanto o objetivo maior é entender a teoria e conseguir gerar seus modelos. Usaremos pacotes de R no Rstudio para gerar esses modelos.

Em casos de imprevistos, também temos um plano B. Então, não se preocupem!

Parte 2

Um checklist de instalações.

Pedimos a instalação do R, Rstudio e Qgis como pré-requisitos, uma vez que o público de pós-graduação no geral já usa/usou essas ferramentas. Se não for o caso, nos links a seguir existem instruções para essas instalações:

R

É possível encontrar as instruções de instalação de acordo com seu sistema operacional (Linux, Mac ou Windows) no link abaixo:

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

Outros links úteis para instalações:

<https://livro.curso-r.com/1-1-instalacao-do-r.html> (Linux, Mac , Windows)

<http://www.des.ufscar.br/software-r> (Windows)

Obs1.: No geral, instalação de R no windows requer a instalação do Rtools

<https://cran.r-project.org/bin/windows/Rtools/>

Parte 2

RStudio

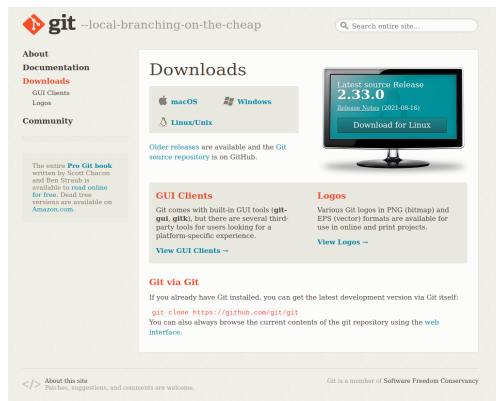
Escolha a opção **RStudio para desktop** e de acordo com seu sistema operacional. Link para instalação:

<https://www.rstudio.com/products/rstudio/download/>

Parte 2

Git

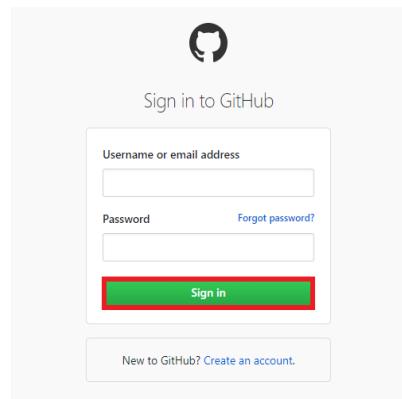
Escolha a opção de download para seu sistema operacional e siga as instruções do link a seguir:
<https://git-scm.com/downloads>



Parte 2

Github

Crie sua conta no Github pelo link <https://github.com/> e caso não tenha uma conta clique e **Create an account**, destacado em azul



Parte 2

Github

Preencha os campos necessários e clique em **Create account**

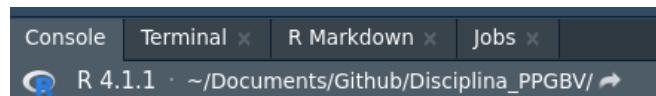
The screenshot shows the GitHub sign-up process. At the top, there's a navigation bar with 'Explore', 'Marketplace', 'Pricing', a search bar, and 'Sign in' and 'Sign up' buttons. The 'Sign up' button is highlighted with a red box. Below the navigation is a 'Join GitHub' link and the main title 'Create your account'. The form consists of several input fields: 'Username *' containing 'ana' with a green checkmark; 'Email address *' containing 'ana.andrade@...' with a green checkmark; and 'Password *' containing '*****'. A note below the password field says: 'Make sure it's at least 15 characters OR at least 8 characters including a number and a lowercase letter.' with a 'Learn more.' link. Under 'Email preferences', there's a checkbox for 'Send me occasional product updates, announcements, and offers.' followed by a 'Verify your account' section. This section contains a challenge message: 'Por favor, resolva esse desafio para que possamos saber que você é uma pessoa de verdade.' with a 'Responda' button. At the bottom is a large blue 'Create account' button.

Parte 2

Configuração Git. Para configurar seu Git existem duas possibilidades.

Configurando pelo Terminal:

No RStudio procure pela aba **Terminal** ou abra seu próprio terminal (conhecido por Prompt no Windows).



Informe para o Git seu nome de usuário (o mesmo do Github) e seu e-mail (o mesmo do Github).

```
git config --global user.name "Tai-Rocha"
```

```
git config --global user.email taina013@gmail.com
```

Parte 2

Configuração Git. Para configurar seu Git existem duas possibilidades.

Configurando pelo pacote R usethis :

Caso não tenha o pacote, instale pelo seguinte comando:

```
install.packages("usethis")
```

Em seguida use a função `use_git_config`

```
usethis::use_git_config(user.name = "Tai-Rocha", user.email = "taina013@gmail.com")
```

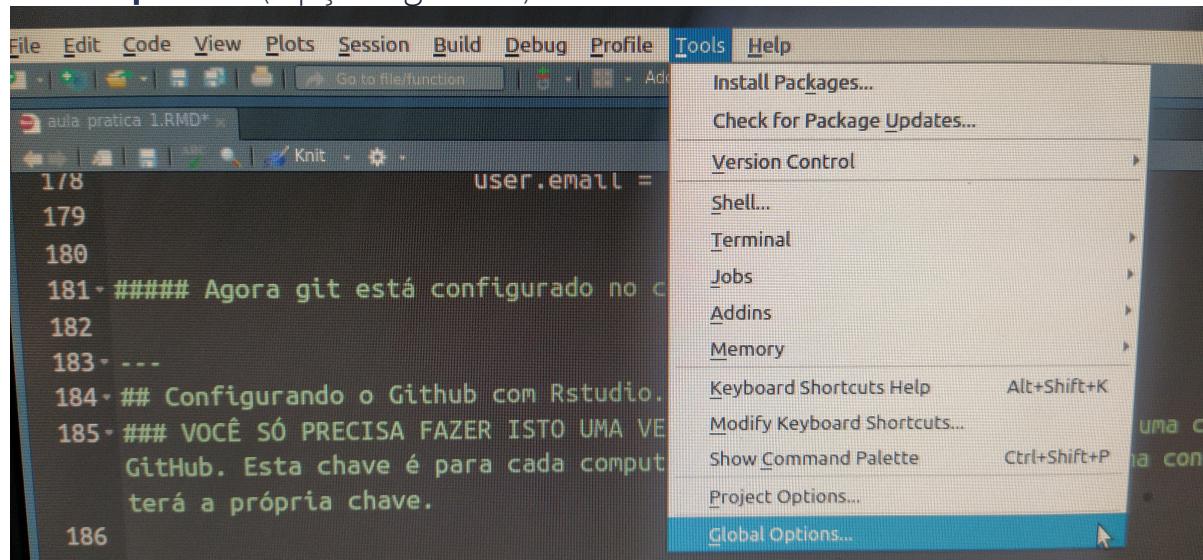
Agora git está configurado no computador e sabe quem é você.

Parte 2

Configurando o Github com RStudio.

VOCÊ SÓ PRECISA FAZER ISTO UMA VEZ EM CADA COMPUTADOR. Vamos gerar uma chave de segurança (que identifica o computador) e copiar para o GitHub. Esta chave é para cada computador individual. Você pode ter uma conta de GitHub e trabalhar em diferentes computadores, e cada um terá a própria chave.

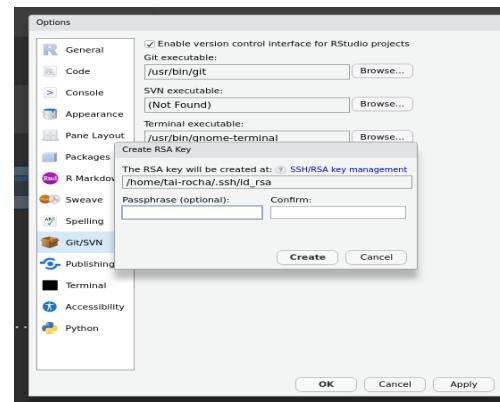
No RStudio, vá até **Global option...** (Opções globais)



Parte 2

Configurando o Github com RStudio.

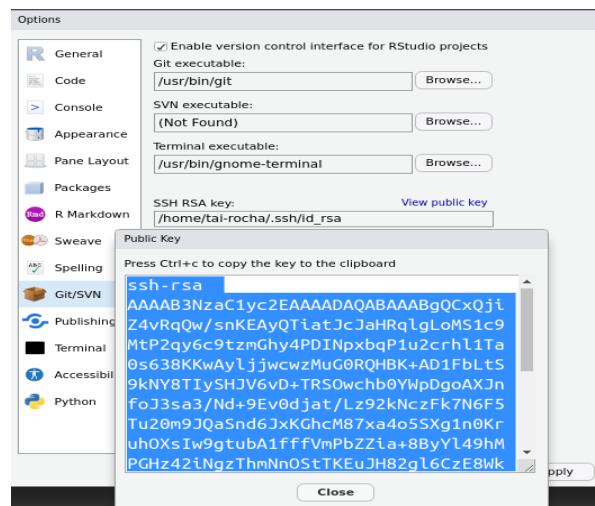
Na opção **Git/SVN** cheque se git está apontando para um arquivo **git.exe** em windows. Mac ou linux devem ter algo como **/usr/bin/git** ou **/usr/bin/gnome-terminal**. Se você nunca fez isto, não deve ter nada no campo da chave RSA, clique em "Criar RSA Key (Create)". O preenchimento dos campos que aparecem é opcional



Parte 2

Configurando o Github com RStudio.

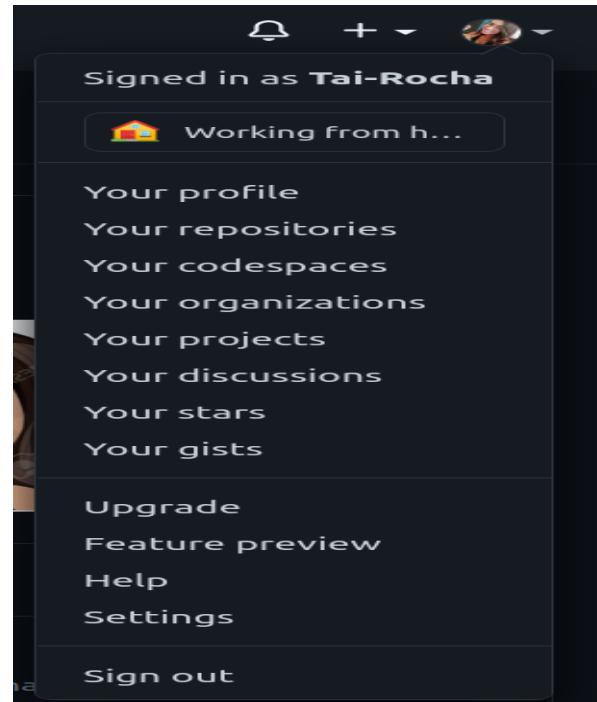
Em seguida clique em **View public key** destacado em azul. Copie a RSA Key. Ela é a chave que identifica seu computador



Parte 2

Configurando o Github com RStudio.

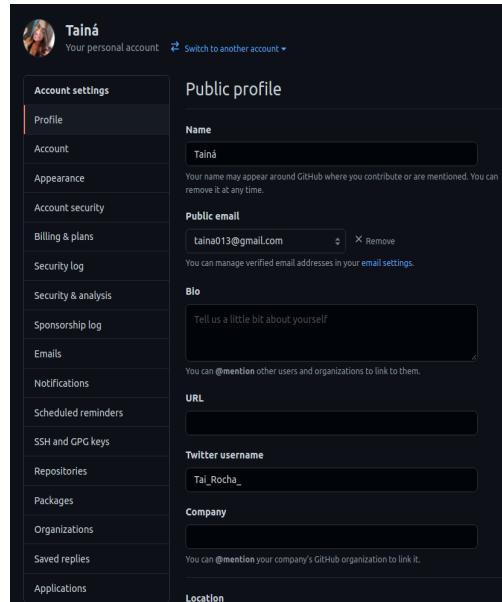
Para copiar no GitHub, vá até o canto superior direito da sua página inicial do Perfil e selecione a opção **Settings**



Parte 2

Configurando o Github com RStudio.

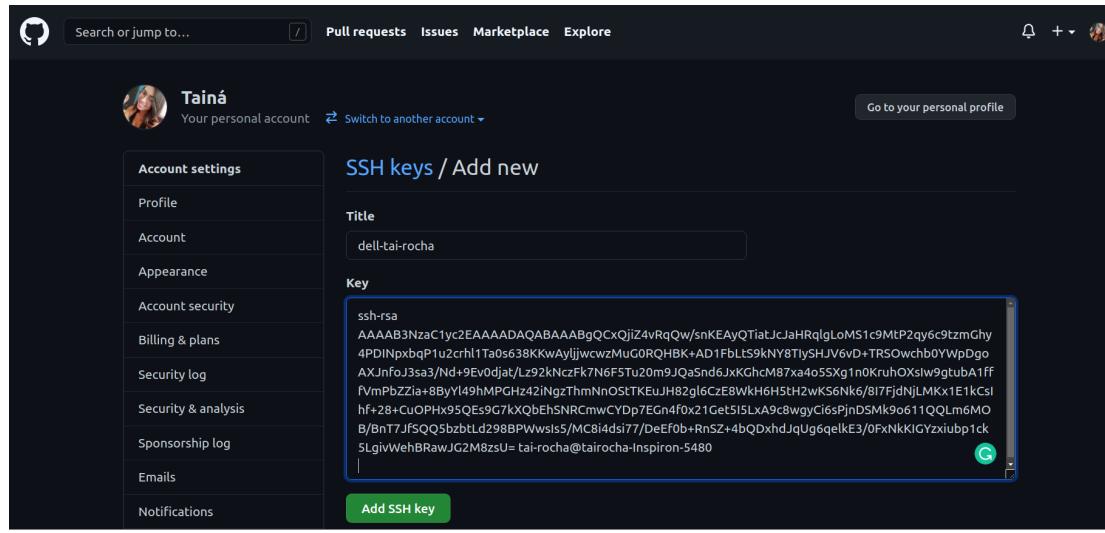
Em seguida vá na opção **SSH and GPG Keys**



Parte 2

Configurando o Github com RStudio.

Cole e adicione a chave

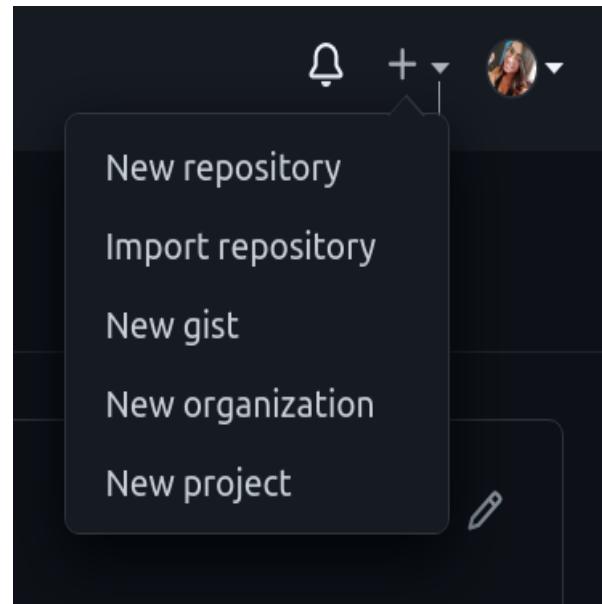


Tudo configurado! Agora R, RStudio, Git e Github podem se comunicar. O próximo passo é criar o repositório do projeto da disciplina. Vamos lá?

Parte 3

Criando o repositório do projeto da disciplina

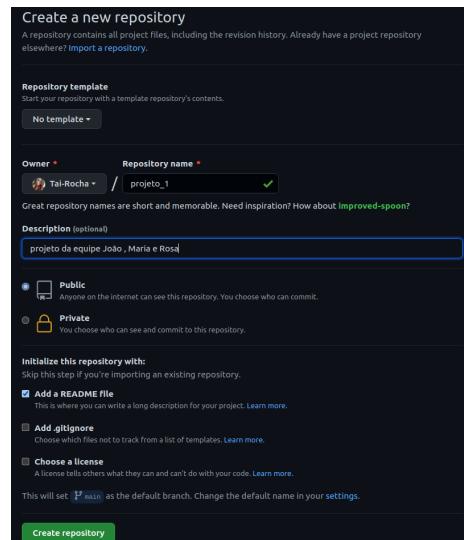
No canto superior direito na sua conta do Github, clique no **+**, em seguida em "New repository"



Parte 3

Criando o repositório do projeto da disciplina

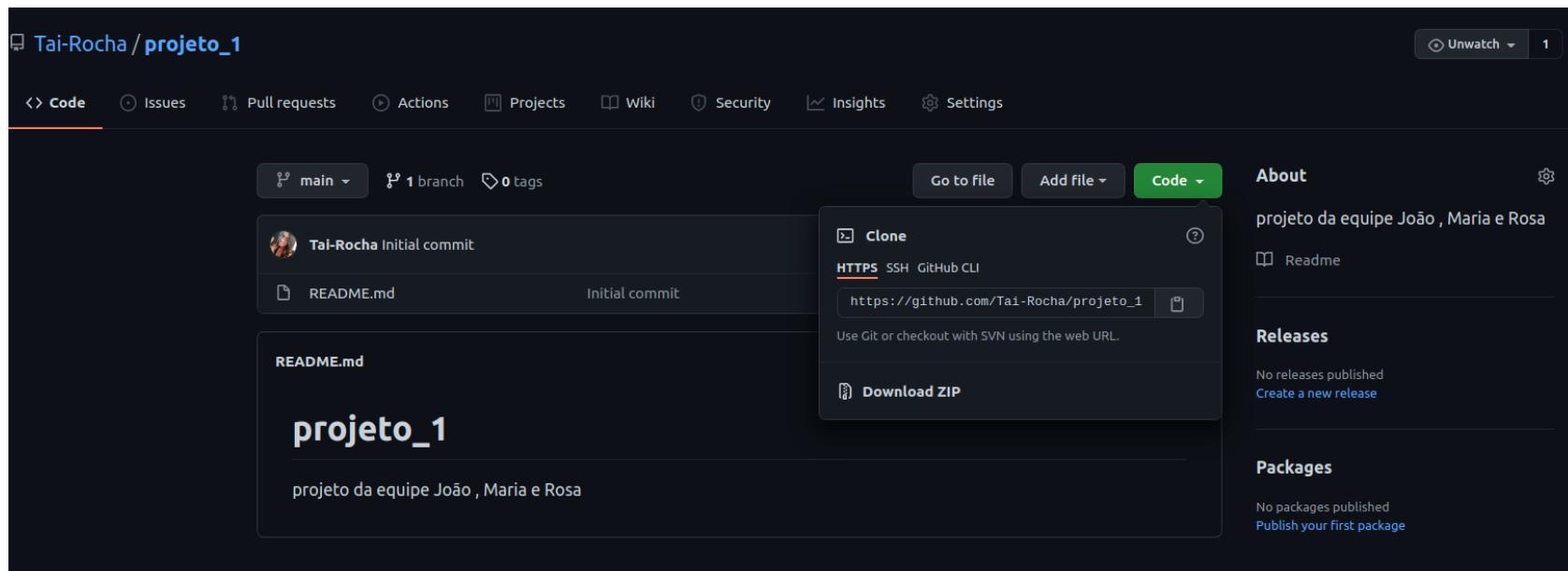
Adicione um nome informativo para seu repositório, crie uma descrição igualmente informativa. Salecione a opção **Public** e opção **Add Readme**. Finalize clicando em **Create repository**



Parte 3

Criando o repositório do projeto da disciplina

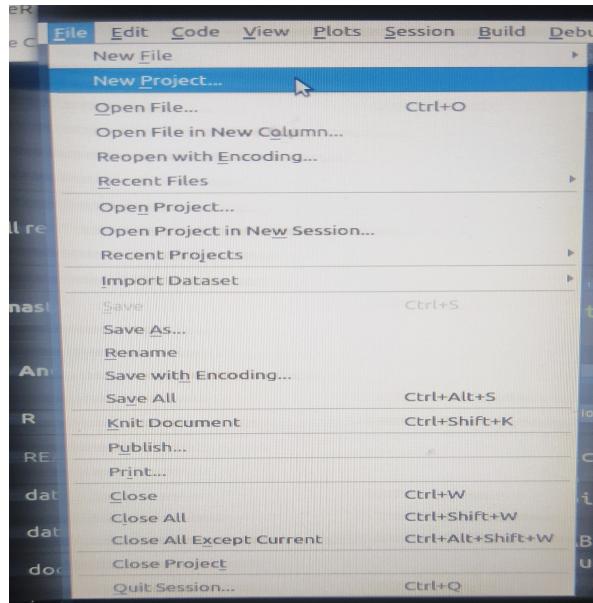
Já no seu repositório , clique na opção **Code** e copie caminho HTTPS



Parte 3

Criando o repositório do projeto da disciplina

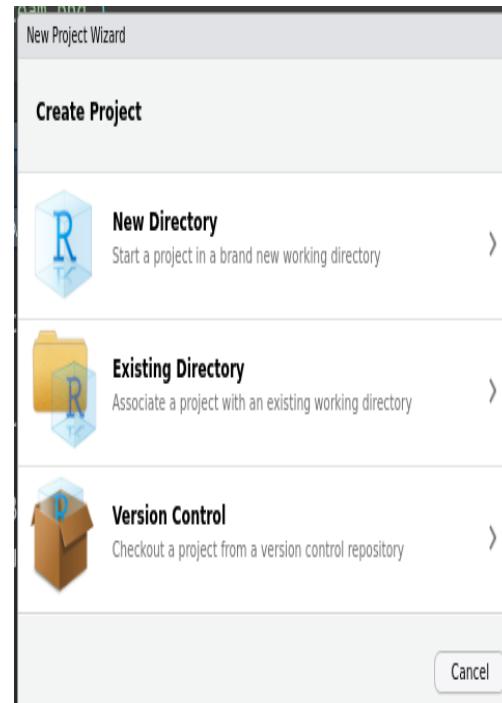
No RStudio vá em **File**, em seguida **New project**



Parte 3

Criando o repositório do projeto da disciplina

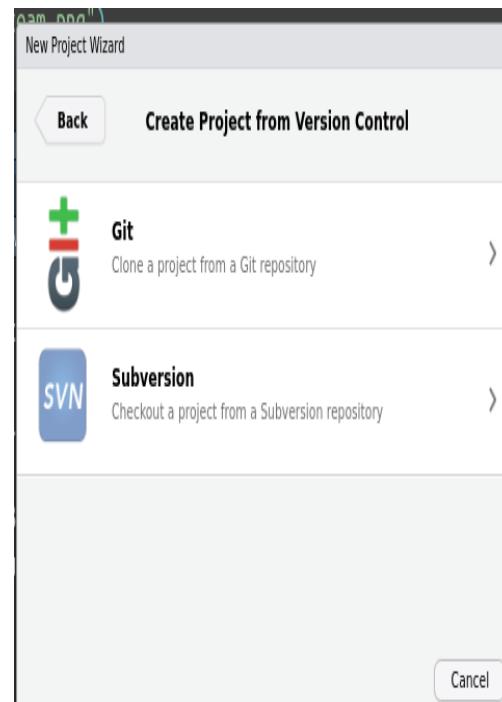
Selecione a opção **Version control**



Parte 3

Criando o repositório do projeto da disciplina

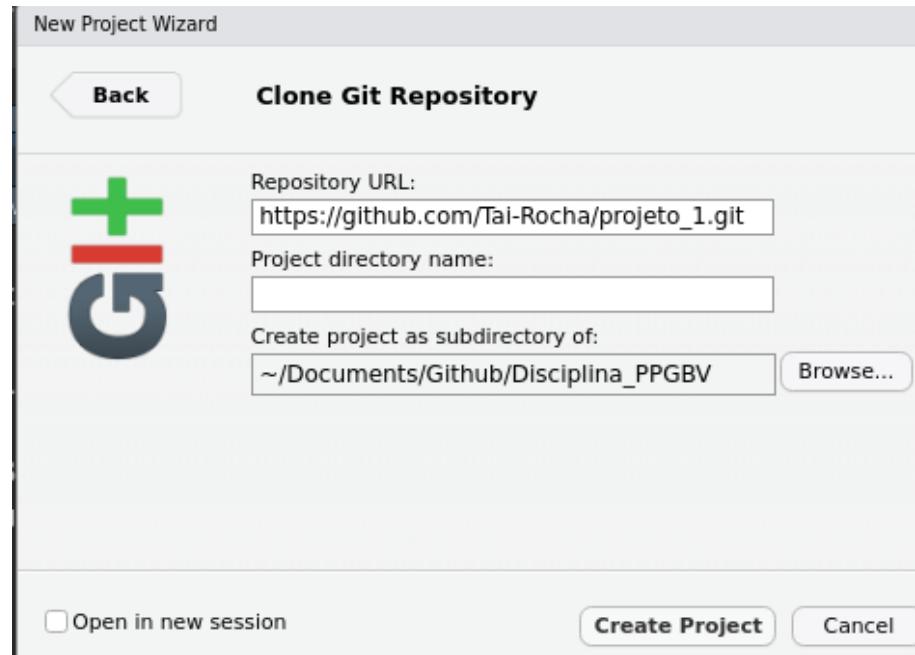
Selecione a opção **Git**



Parte 3

Criando o repositório do projeto da disciplina

Indique o caminho onde seu projeto/repostório ficará salvo

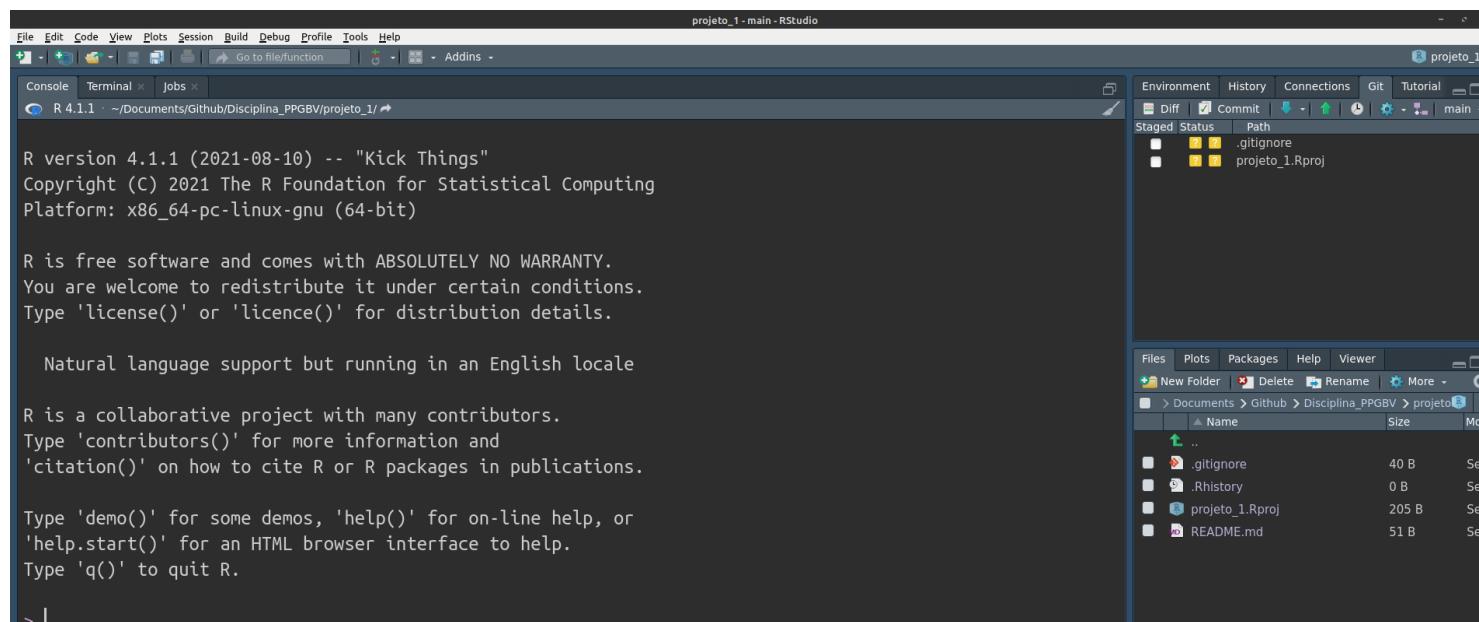


Para terminar clique em **Create Project**

Parte 3

Criando o repositório do projeto da disciplina

Ao criar seu projeto desta forma, no seu RStudio deve surgir uma aba chamada Git. É por onde vamos realizar o controle de versão (histórico de alterações dos nossos scripts)



Parte 4

Organização dos projetos

Vamos criar uma estrutura de pastas para seu projeto. Isso facilitará o gerenciamento do projeto. Análises de ENMs/SDMs envolvem diferentes tipos de arquivos (é fácil se perder se não houver essa organização).

```
project/
└── data/
└── docs/
└── figs/
└── R/
└── output/
└── README.md
```

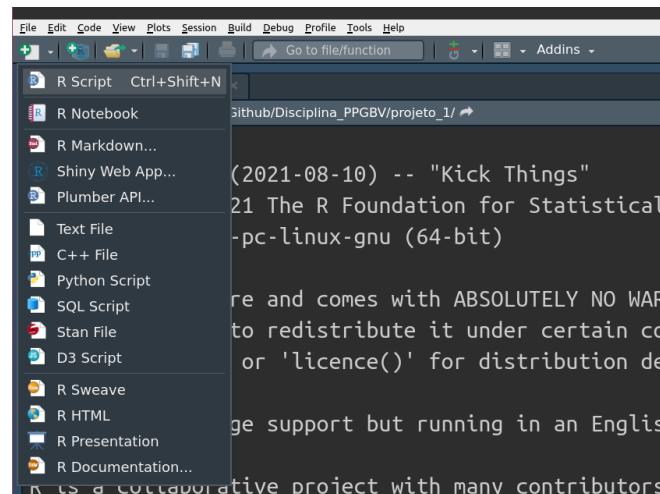
Uma estrutura simples envolve basicamente pastas para os dados, para os scripts, para os resultados, para documentos gerais (.doc, .pdf, .ppt) e para figuras. Dentro destas podem existir outras subpastas. A melhor organização fica a seu critério.

Parte 4

Organização dos projetos

Testando o controle de versão para um script R

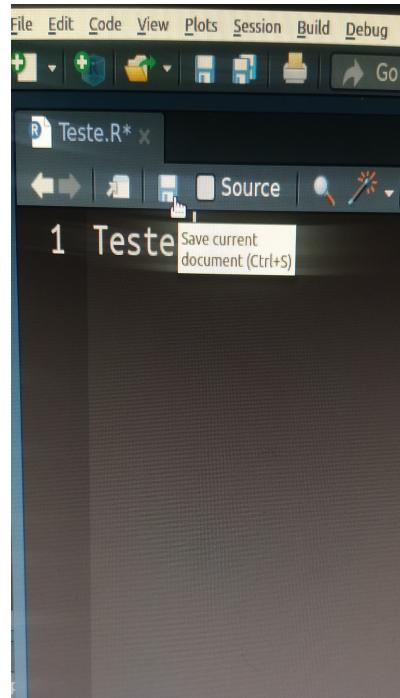
Vamos criar um script de teste. Vá no ícone "New file" e clique em "R script"



Parte 4

Organização dos projetos

Digite **Teste** no seu arquivo R que até agora está em branco. Em seguida salve este script (clicando no ícone do disquete) com o nome "Teste" na pasta **R**. Salvaremos todos os scripts R nesta pasta.

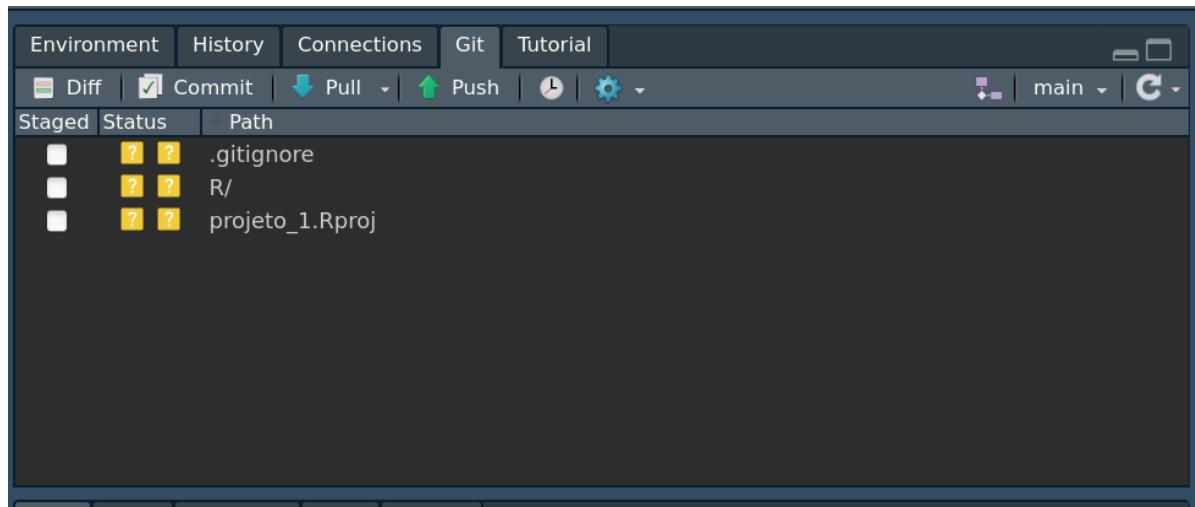


Parte 4

Organização dos projetos

Testando o controle de versão para um script R

Vá para aba Git e clique em **R/** (Essa pasta subirá junto com seu script para o Github)

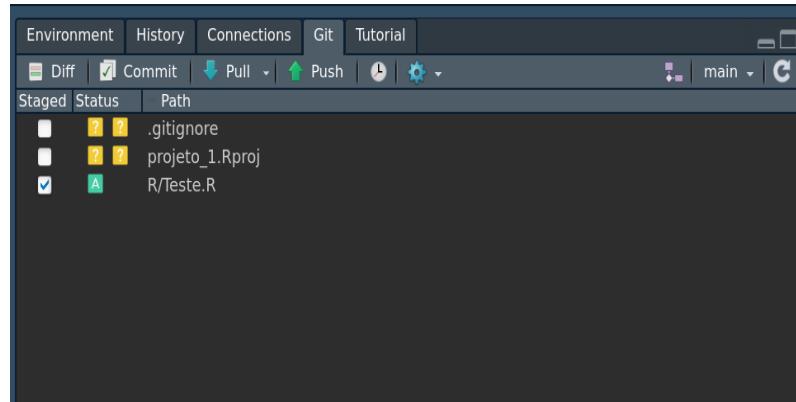


Parte 4

Organização dos projetos

Testando o controle de versão para um script R

Após clicar em **R/**, o Git sinaliza que este arquivo foi adicionado (**A**).

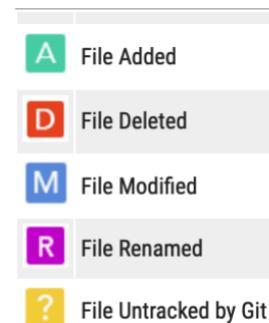


Parte 4

Organização dos projetos

Testando o controle de versão para um script R

O Git reconhece as seguintes ações:

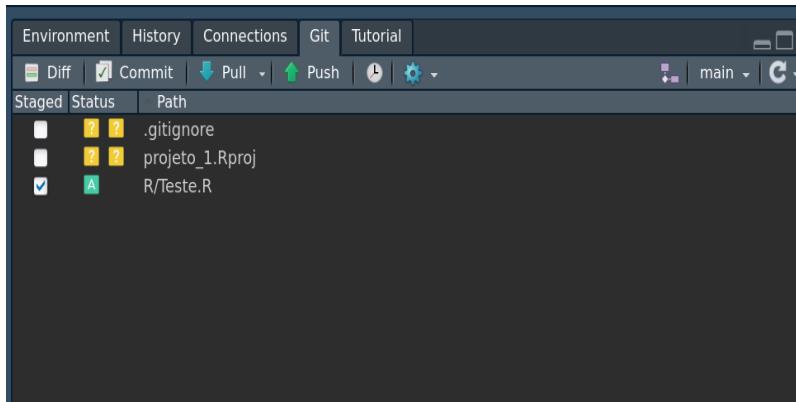


Parte 4

Organização dos projetos

Testando o controle de versão para um script R

Clique em **Diff**

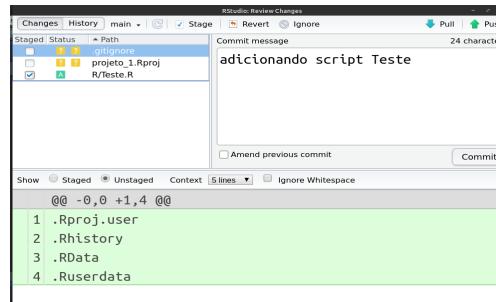


Parte 4

Organização dos projetos

Testando o controle de versão para um script R

Adicione uma mensagem informativa e clique em "Commit" e em seguida "Push".



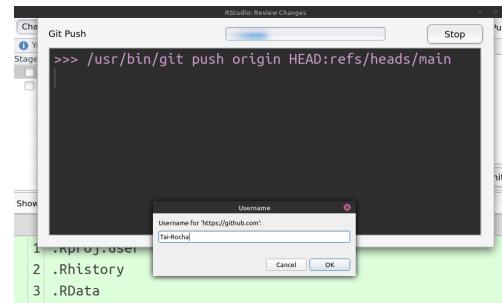
Parte 4

Organização dos projetos

Testando o controle de versão para um script R

Talvez o git/rstudio peça suas credenciais (adicone o nome de usuário e senha que foram cadastradas aqui e no Github)

Usuário:

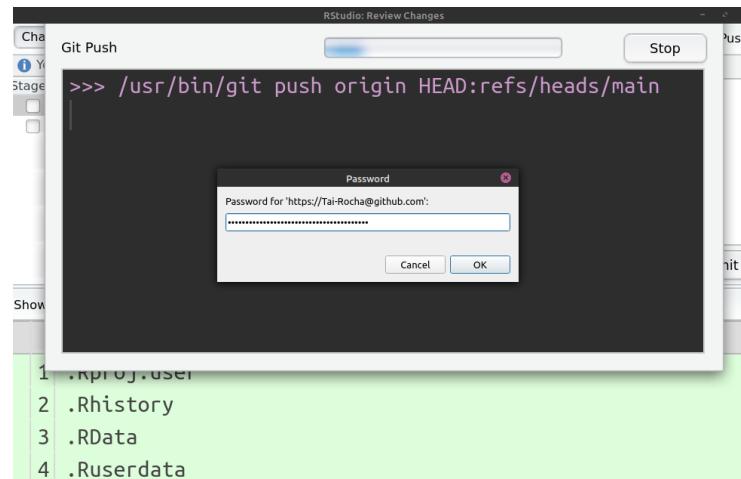


Parte 4

Organização dos projetos

Testando o controle de versão para um script R

Senha:

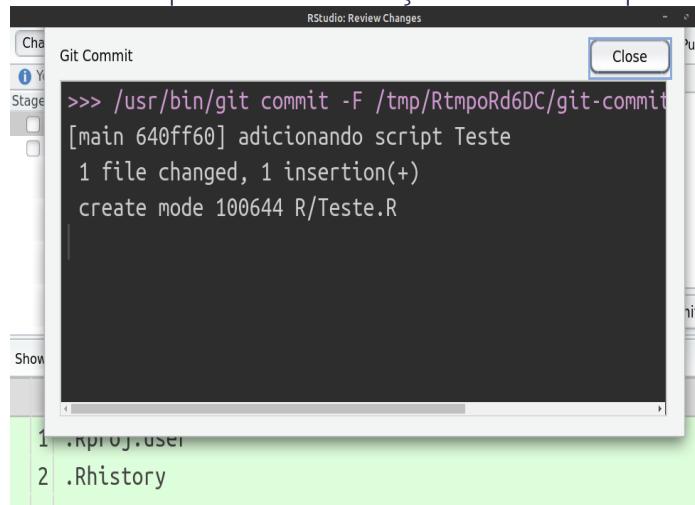


Parte 4

Organização dos projetos

Testando o controle de versão para um script R

Ao finalizar a seguinte mensagem informará quais modificações subiram para o Github.



```
>>> /usr/bin/git commit -F /tmp/RtmpoRd6DC/git-commit
[main 640ff60] adicionando script Teste
1 file changed, 1 insertion(+)
create mode 100644 R/Teste.R
```

Vá até o Github e veja que seu script agora está lá.

Parte 4

Organização dos projetos

Testando o controle de versão para um script R

Adicione qualquer texto ao seu script Teste e salve as alterações.

The screenshot shows the RStudio interface with a project titled "projeto_1". In the left panel, there is a script editor window for "Teste.R" containing the following code:

```
1 Teste
2
3 Agora este script está armazenado na nuvem do Github. Verifique você
mesmo lá no seu repositório do Github.
```

In the top right corner of the RStudio window, there is a GitHub icon indicating that the project is connected to a GitHub repository. The sidebar on the right shows the project structure:

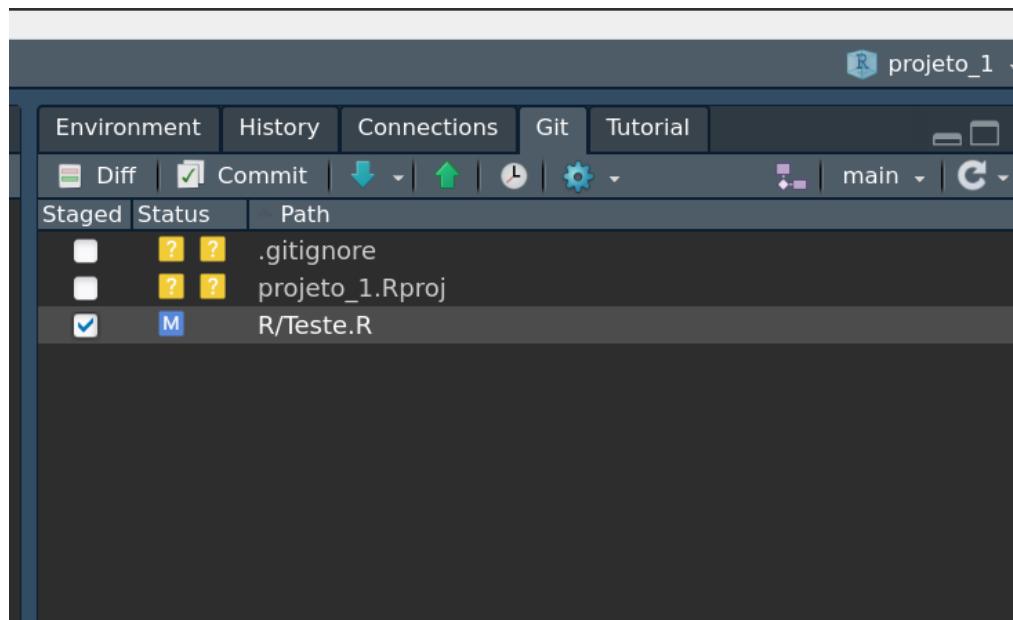
- .gitignore
- projeto_1.Rproj
- R/Teste.R

The bottom right corner of the RStudio window shows the file browser with the path: Home > Documents > Github > Disciplina_PPGBV > projeto_1.

Parte 4

Organização dos projetos

Veja que o Git sinaliza que seu arquivo foi modificado. Clique no **M**. Em seguida repita os passos anteriores clicando em Diff, adicionando uma mensagem informativa e em seguida clicar em Commit e finalizar clicando em Push



Fim da prática !

Pronto, agora você pode trabalhar em seus scripts fazendo controle de versão!

