

# Como manter seu projeto organizado, seguro e acessível?

## Introdução ao RStudio + GitHub

ESTAT0090 – Estatística Computacional

Prof. Dr. Sadraque E. F. Lucena

sadraquelucena@academico.ufs.br

# Cenário profissional

## Uma História Real (ou quase)

- Imagine que você é contratado por uma empresa para analisar dados de clientes e construir relatórios periódicos.
- Você começa empolgado, faz seu código em R, roda tudo no seu computador e entrega um relatório lindo.
- Um mês depois, o gerente pede que você refaça a análise com os novos dados.
- Só tem um problema:
  - você não lembra qual script usou.
  - Os arquivos estão todos em “versão final\_v3\_final\_real.R”.
  - E o relatório? Foi feito no Word e agora está com outro estagiário.

# Cenário profissional

## O Problema

Você percebe que:

- Seus scripts estão bagunçados.
- O histórico do que foi feito se perdeu.
- O relatório não é reproduzível.
- Ninguém (nem você mesmo) entende o que foi feito.

## A Motivação

Como resolver isso? Você precisa de:

- Um ambiente organizado para escrever e executar código (RStudio).
- Um jeito de acompanhar o histórico do seu trabalho (Git).
- Uma forma de colaborar e compartilhar com outras pessoas (GitHub).
- Relatórios que podem ser reproduzidos a qualquer momento (Quarto).

# Um Exemplo da Nossa Própria Disciplina

Cenário:

- Você está cursando Estatística Computacional.
- A cada aula, o professor entrega:
  - Slides da aula com explicações e um código em R.
  - Um problema para você resolver, adaptando e expandindo o código.
- Você modifica, testa, ajusta e responde o problema.
- Na aula seguinte, você parte exatamente de onde parou.

# E se você fizer tudo isso com **git** e **GitHub**?

Você terá:

- Um histórico completo de aprendizagem.
- Um repositório que mostra sua evolução.
- Uma base reutilizável para projetos futuros.

É como criar seu próprio material interativo de Estatística Computacional!

# Objetivo da aula

- Entender o que é e para que serve o Git e o GitHub.
- Saber como criar um repositório de projeto.
- Atualizar repositório no GitHub via Studio.

# Breve introdução ao GitHub

# O que é o git?

- **Git** é uma ferramenta que **ajuda a controlar e gerenciar mudanças em arquivos** ao longo do tempo.
- Ele permite que você **salve versões** diferentes de um trabalho à medida que faz alterações, de modo que possa **voltar para versões anteriores** se algo der errado ou se precisar revisar mudanças feitas.

## Por que o git é importante?

- **Evita perda de trabalho:** Se você estiver escrevendo código ou criando qualquer tipo de documento, o Git permite que você **salve diferentes versões** do seu trabalho. Assim, se algo der errado, você pode voltar a uma versão anterior.
- **Facilita o trabalho em equipe:** Quando várias pessoas estão trabalhando no mesmo projeto, o Git permite que cada uma **trabalhe de forma independente** e depois **una os trabalhos de maneira organizada**. Isso evita que as alterações de uma pessoa sobrescrevam as de outra.
- **Organização e rastreamento:** O Git mantém um **histórico completo** de todas as mudanças feitas em um projeto, permitindo saber **quem fez o quê e quando**.

# O que é o GitHub?

- **GitHub** é uma plataforma online que **armazena e organiza projetos que utilizam Git**.
- Ele permite que você **publique seu código, compartilhe arquivos e colabore com outras pessoas** em projetos de forma fácil e eficiente.

## Por que o **GitHub** é importante?

- **Armazenamento seguro:** Com o GitHub, seus projetos ficam **seguramente armazenados na nuvem**. Isso significa que você pode acessar seu trabalho de qualquer lugar e sempre terá uma cópia segura.
- **Colaboração em equipe:** GitHub permite que várias pessoas trabalhem no mesmo projeto ao mesmo tempo. Cada pessoa pode fazer mudanças no código, e o GitHub ajuda a **gerenciar essas mudanças** sem que uma sobrescreva a outra.
- **Histórico e transparência:** O GitHub mantém um **histórico completo** de todas as alterações feitas no seu projeto. Isso permite ver **quem fez o quê e quando**, facilitando o acompanhamento e revisão do trabalho de equipe.

# Como e por que usar o GitHub na disciplina?

- Você receberá um script em R a cada aula
- Durante a aula, vai editar esse script no RStudio, testando e resolvendo problemas
- No final da aula, envia suas alterações para seu repositório (push) — tudo salvo e organizado
- Pode acessar seu trabalho de qualquer lugar, com segurança e histórico garantido

GitHub será seu **caderno digital de códigos** — inteligente, seguro e acessível para a disciplina

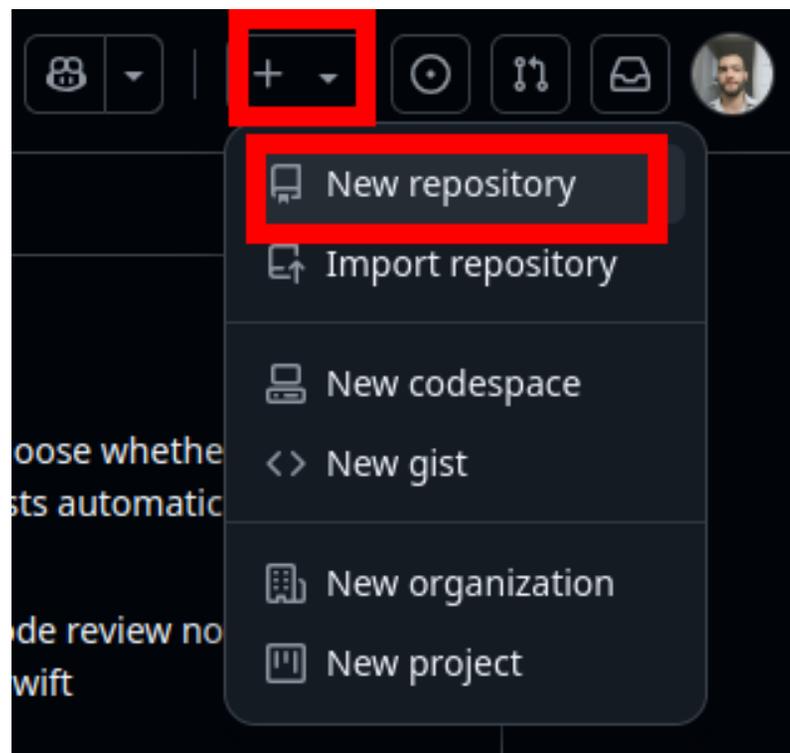
# Primeiros passos

1. Baixar e instalar o Git: <https://happygitwithr.com/install-git.html>
2. Criar uma conta no GitHub: <https://github.com/>
3. Criar um repositório Git
4. Clone esse repositório para sua máquina usando o RStudio
5. Trabalhe no projeto e envie as atualizações de volta ao GitHub

Desenvolva o hábito de buscar soluções por conta própria. Isso faz parte do dia a dia de quem trabalha com dados. Comece agora a desenvolver essa autonomia.

# Criando um repositório **git** no **GitHub**

- Após fazer login no GitHub, Clique em + no canto superior direito
- Em seguida, clique em [New repository](#)



# Criando um repositório **git** no **GitHub**

- Em **Repository name** dê um nome ao repositório
- Em **Description** faça uma descrição do repositório
- Marque a opção **Public** ou **Private**
- Em **Initialize this repository with:** marque a opção **Add a README file**
- Clique em **Create repository**

## Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

*Required fields are marked with an asterisk (\*).*

**Owner \*** sadraquelucena / **Repository name \*** EstComp2025-1  
EstComp2025-1 is available.

Great repository names are short and memorable. Need inspiration? How about [upgraded-spork](#) ?

**Description (optional)**  
 Repositório para a disciplina ESTAT0090 - Estatística Computacional da UFS no período 2025-1

**Public**  
 Anyone on the internet can see this repository. You choose who can commit.

**Private**  
 You choose who can see and commit to this repository.

**Initialize this repository with:**

**Add a README file**  
 This is where you can write a long description for your project. [Learn more about READMEs.](#)

**Add .gitignore**  
 .gitignore template: None  
 Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

**Choose a license**  
 License: None  
 A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

This will set `master` as the default branch. Change the default name in your [settings](#).

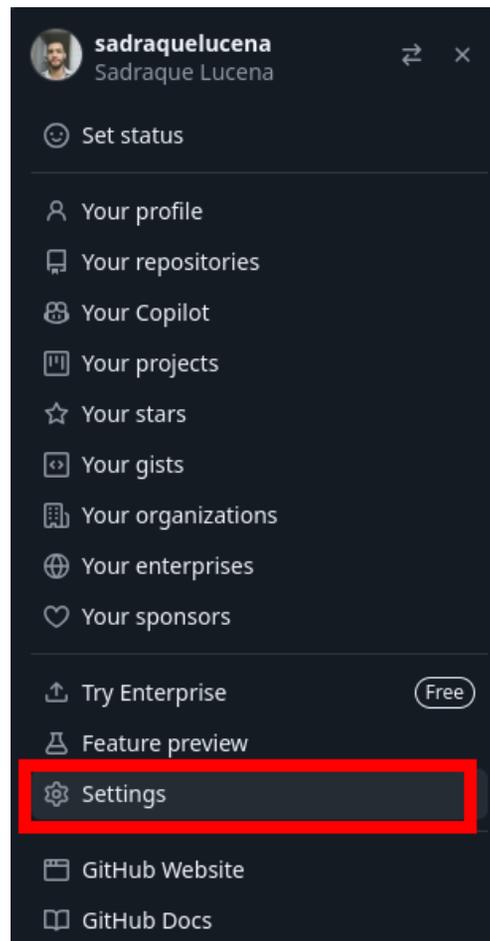
📄 You are creating a public repository in your personal account.

**Create repository**

- Com o repositório já criado no GitHub, agora vamos usar o RStudio para **ligar o projeto local ao repositório remoto**.
- Assim, todas as alterações feitas no RStudio poderão ser **salvas na nuvem e versionadas automaticamente**.
- Para enviar essas alterações ao GitHub, será necessário se autenticar — com **login e senha** ou com um **token de acesso**.
  - Vamos ver como criar um token de acesso no GitHub.

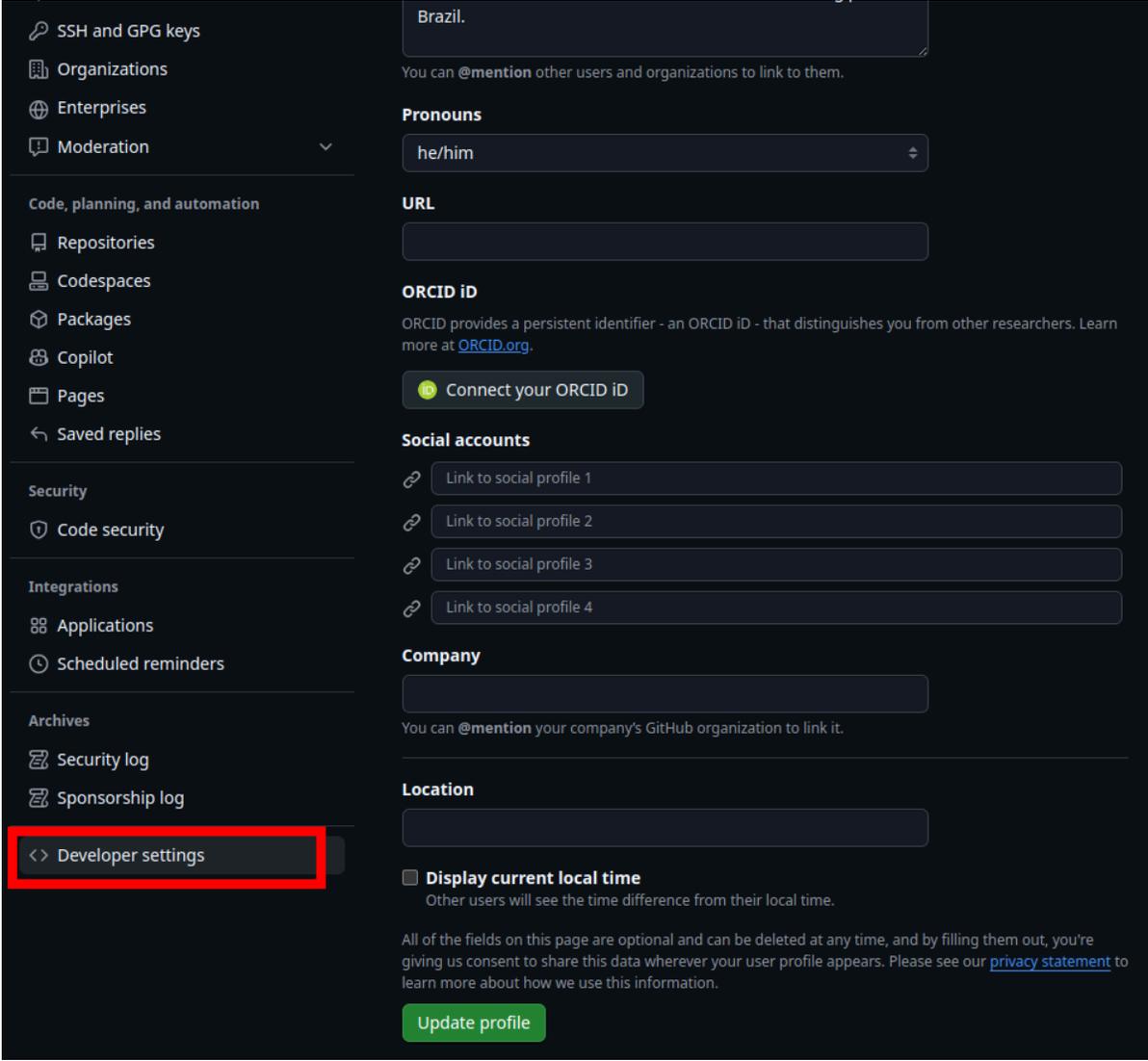
# Criando um token de acesso no GitHub

1. Estando logado no GitHub, clique na sua **foto de perfil** no canto superior direito
2. Clique em [Settings](#)



# Criando um token de acesso no GitHub

3. No canto inferior esquerdo da tela clique em `Developer settings`



The screenshot shows the GitHub Developer settings page. The left sidebar contains a list of settings categories, with 'Developer settings' highlighted by a red box. The main content area is dark-themed and contains several sections: 'Brazil.' (location), 'Pronouns' (set to 'he/him'), 'URL', 'ORCID iD' (with a 'Connect your ORCID iD' button), 'Social accounts' (four 'Link to social profile' buttons), 'Company', 'Location', and 'Display current local time' (unchecked). At the bottom, there is a green 'Update profile' button and a disclaimer about optional fields.

SSH and GPG keys

Organizations

Enterprises

Moderation

Code, planning, and automation

Repositories

Codespaces

Packages

Copilot

Pages

Saved replies

Security

Code security

Integrations

Applications

Scheduled reminders

Archives

Security log

Sponsorship log

<> Developer settings

Brazil.

You can @mention other users and organizations to link to them.

**Pronouns**

he/him

**URL**

**ORCID iD**

ORCID provides a persistent identifier - an ORCID iD - that distinguishes you from other researchers. Learn more at [ORCID.org](https://orcid.org).

Connect your ORCID iD

**Social accounts**

Link to social profile 1

Link to social profile 2

Link to social profile 3

Link to social profile 4

**Company**

You can @mention your company's GitHub organization to link it.

**Location**

**Display current local time**

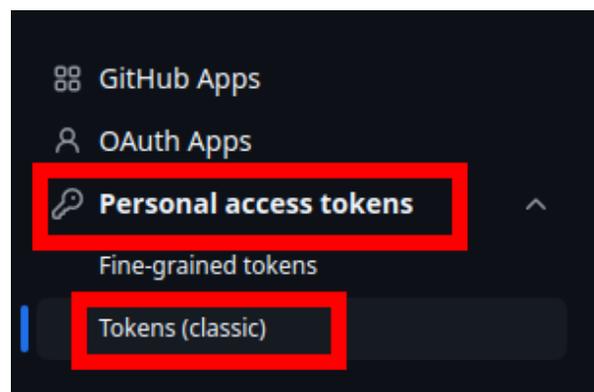
Other users will see the time difference from their local time.

All of the fields on this page are optional and can be deleted at any time, and by filling them out, you're giving us consent to share this data wherever your user profile appears. Please see our [privacy statement](#) to learn more about how we use this information.

Update profile

# Criando um token de acesso no GitHub

4. No canto superior esquerdo da tela clique em [Personal access tokens](#)
5. Clique em [Tokens \(classic\)](#)



6. Em [Expiration](#) selecione a data em que o token irá expirar
7. Marque todas as opções em [Select scopes](#)
8. Clique em [Generate token](#)

O token será gerado uma única vez. Guarde-o com cuidado, pois não será possível visualizá-lo novamente no GitHub. Você usará esse token quando for solicitada autenticação.

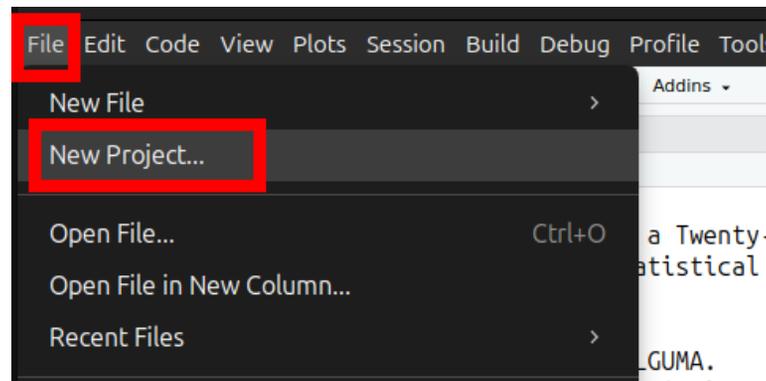
# Integrando Studio<sup>®</sup> e GitHub

# Studio® + **GitHub: integração prática**

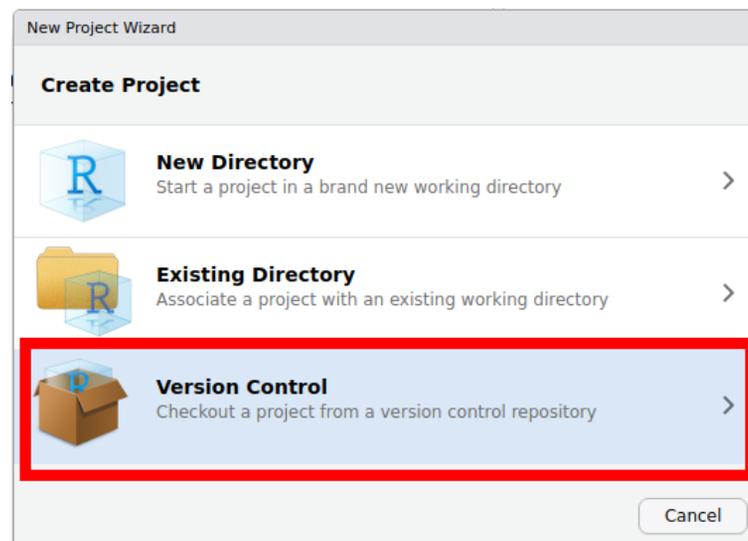
- O **RStudio** possui integração nativa com o **Git** e **GitHub**
  - Ou seja, é possível sincronizar um repositório GitHub a um repositório local
  - Isso significa que você pode ligar o repositório do GitHub (na nuvem) ao seu projeto no computador. Assim, o que você altera localmente pode ser enviado para o GitHub — e vice-versa.
- Para isso, seguimos os seguintes passos:
  1. Fazemos uma cópia do repositório do GitHub na máquina local usando o RStudio.
    - Quando já há uma cópia na máquina, começamos o trabalho atualizando o projeto local com as alterações que estão no GitHub (**pull**).
  2. Trabalhamos normalmente no projeto: scripts, análises, relatórios...
  3. Usamos o Git para **registrar as alterações** (**commit**) e **enviar para o GitHub** (**push**).

# GitHub + R Studio: criando o projeto local

- No canto superior direito do RStudio clique em **File > New Project**

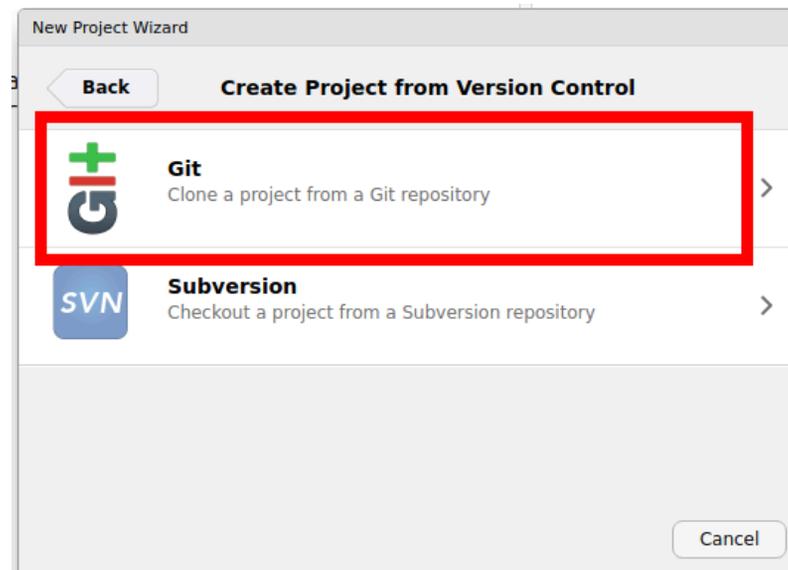


- Clique em **Version control**



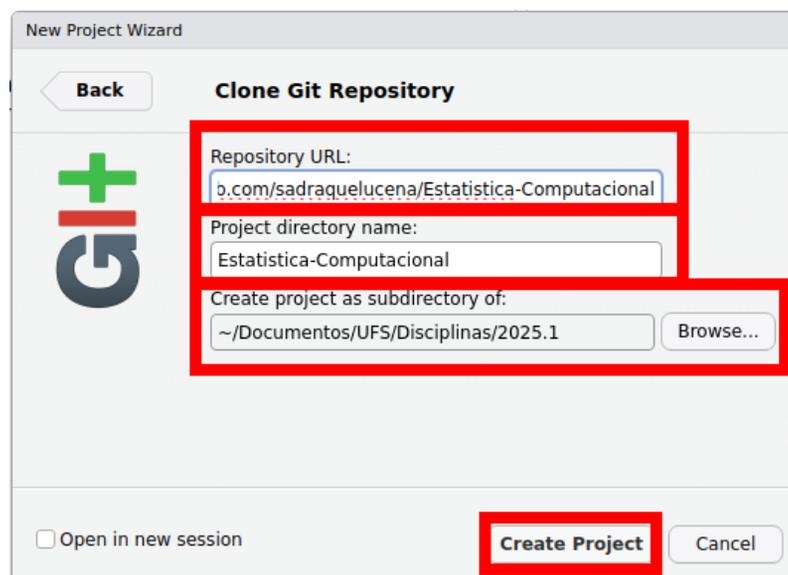
# GitHub + R Studio: criando o projeto local

- Clique em [Git](#)



# GitHub + R Studio: criando o projeto local

- No campo `Repository URL`, cole a URL do repositório que você criou no GitHub
- Em `Create project as subdirectory of`, escolha o diretório em que o repositório do GitHub será copiado na máquina local
- Clique em `Create Project`



# **GitHub** + **Studio®: criando o projeto local**

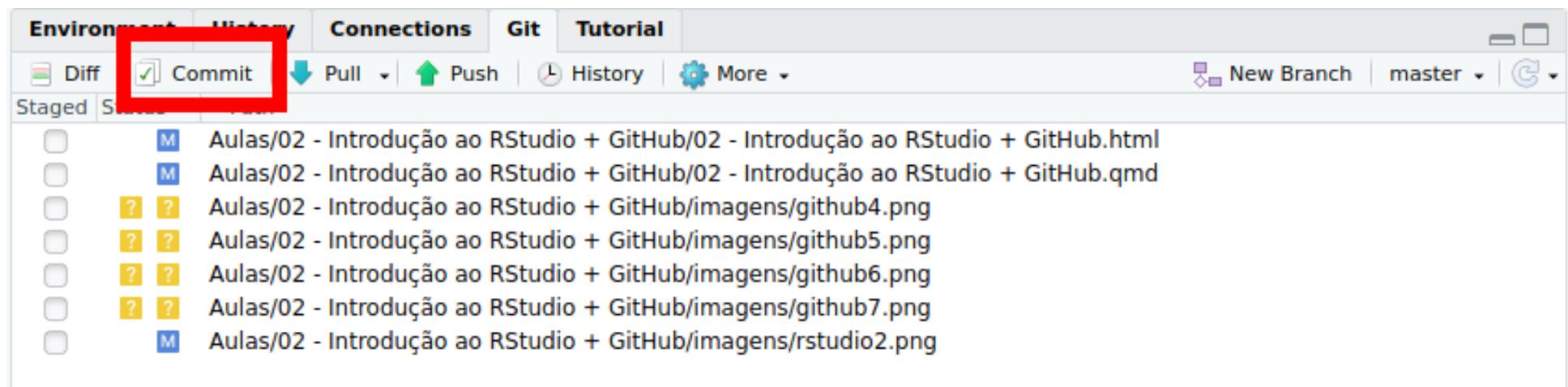
- Se você estiver clonando um **repositório público**, o RStudio irá criar uma cópia do projeto localmente, sem exigir login.
- Se o repositório for **privado**, o GitHub pedirá que você se autentique (login e senha ou token).

# Atualizando repositório no



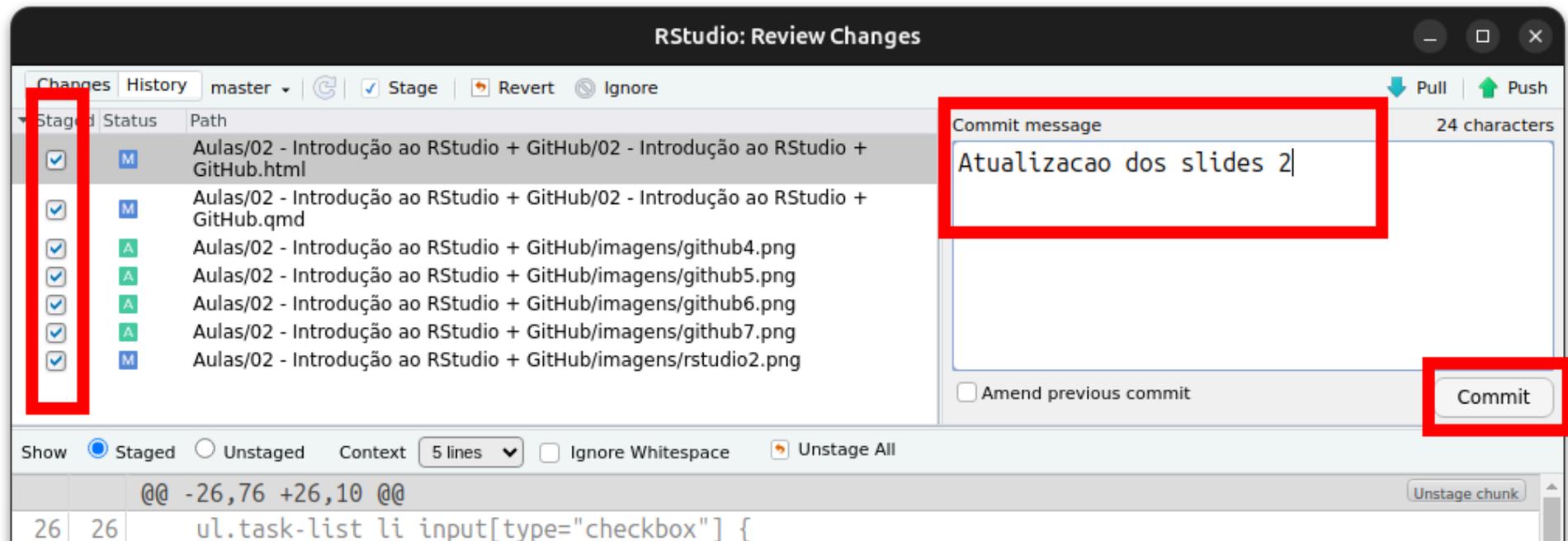
# Enviando alterações para o GitHub via RStudio®

- Depois de salvar as atualizações do seu projeto local, você pode **enviar essas alterações para o repositório no GitHub diretamente pelo RStudio**.
- Você deve fazer:
  1. No quadrante superior direito clique em **Commit**



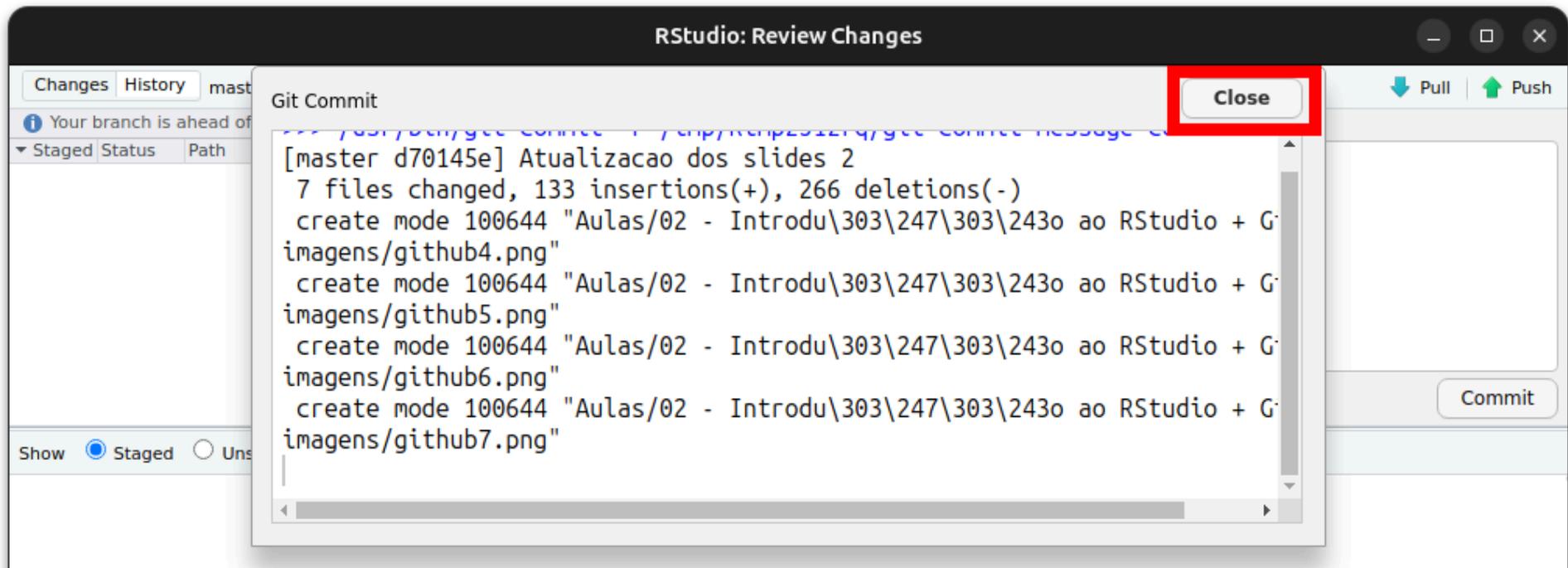
# Enviando alterações para o GitHub via RStudio®

2. RStudio mostra os arquivos que foram alterados. Selecione-os
3. No campo **Commit message** escreva um comentário contendo o que foi atualizado
4. Clique em **Commit**



# Enviando alterações para o GitHub via RStudio®

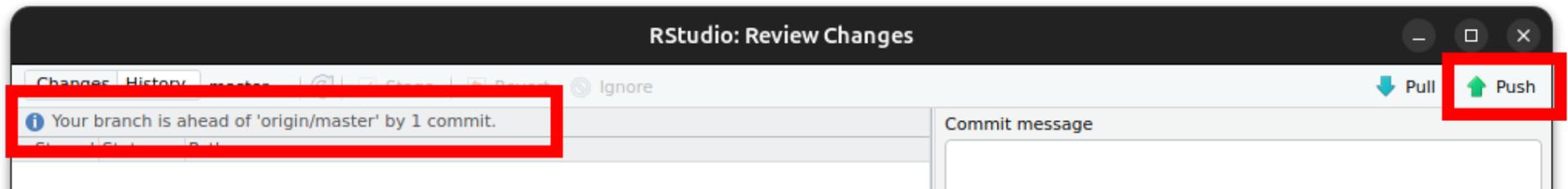
## 5. Clique em [Close](#)



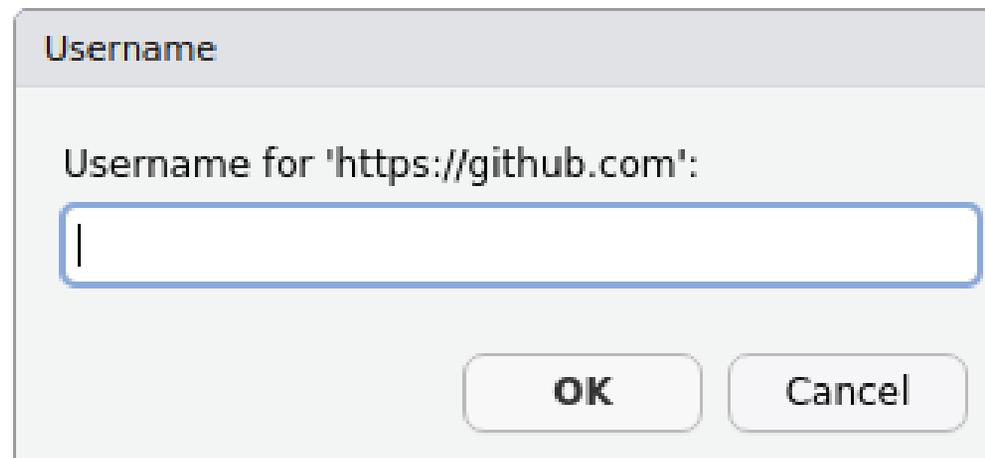
# Enviando alterações para o GitHub via RStudio®

6. Note que irá aparecer a mensagem `Your branch is ahead of 'origin/master' by 1 commit`

7. Clique em `Push`



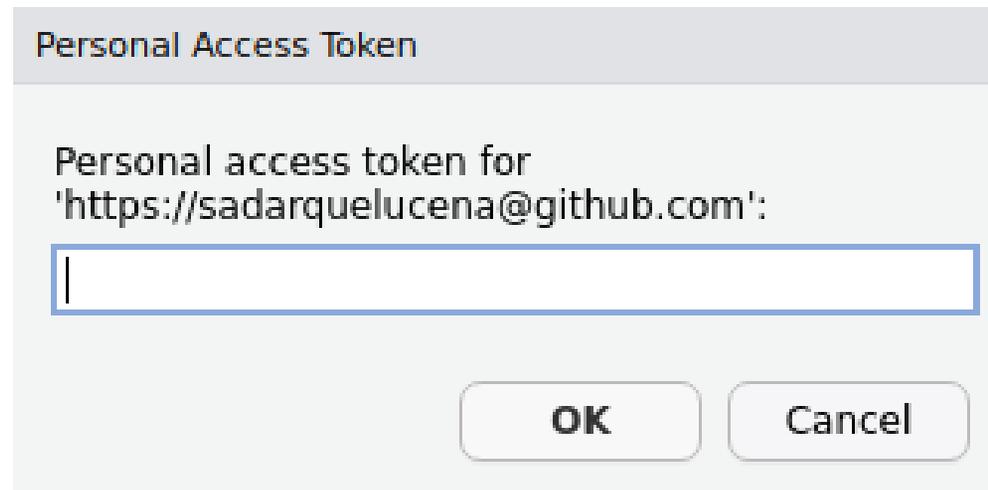
8. No campo `Username for 'https://github.com'` coloque login e clique em `OK`



# Enviando alterações para o GitHub via Studio<sup>®</sup>

9. No campo **Personal Access Token** insira o token criado no GitHub

10. Clique em **OK**



Personal Access Token

Personal access token for  
'https://sadarqu Lucena@github.com':

OK Cancel

# Enviando alterações para o GitHub via Studio®

11. Caso apareça a mensagem abaixo, os arquivos foram atualizados no repositório do GitHub.

```
Git Push Close  
>>> /usr/bin/git push origin HEAD:refs/heads/master  
To https://github.com/sergipe/Estadistica-Computacional  
1541a8f..d70145e HEAD -> master
```

# Material Extra

Aprofunde o que vimos em aula com esses vídeos no YouTube:

- Curso completo de Git e GitHub: <<http://tiny.cc/GitGitHub>:
- Integração do RStudio com o GitHub:
  - Parte 1: <http://tiny.cc/RStudioGitHub1>
  - Parte 2: <http://tiny.cc/RStudioGitHub2>

# Ganhos da aula

- Versionamento de código e arquivos com GitHub
- Integração do RStudio com GitHub
- Experiência com ferramentas do mercado

# Atividade extraclasse

## Configure seu ambiente de trabalho pessoal

### Objetivo

Deixar seu computador pessoal pronto para continuar os trabalhos da disciplina fora do laboratório, de forma independente.

### Etapas:

- Instalar o Git: <https://happygitwithr.com/install-git.html>
- Instalar o R: <https://cran.r-project.org>
- Instalar o RStudio: <https://posit.co/download/rstudio-desktop>
- Fazer uma cópia (clone) do **repositório da disciplina no GitHub** para o seu computador.

Criar um **repositório de teste**, clonar no RStudio e fazer seu **primeiro commit**.

**Fim**