

# Introdução ao RStudio + GitHub

ESTAT0109 – Mineração de Dados em Estatística

Prof. Dr. Sadraque E. F. Lucena

sadraquelucena@academico.ufs.br

<http://sadraquelucena.github.io/mineracao>

# Uso na Disciplina

Cenário:

- Você está cursando *Mineração de Dados em Estatística*.
- 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 acessa de qualquer máquina o que você fez exatamente de onde parou.

# Você pode fazer tudo isso com **git** e







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 Mineração de Dados!

# 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  **R** Studio®.

# Breve introdução ao



# 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

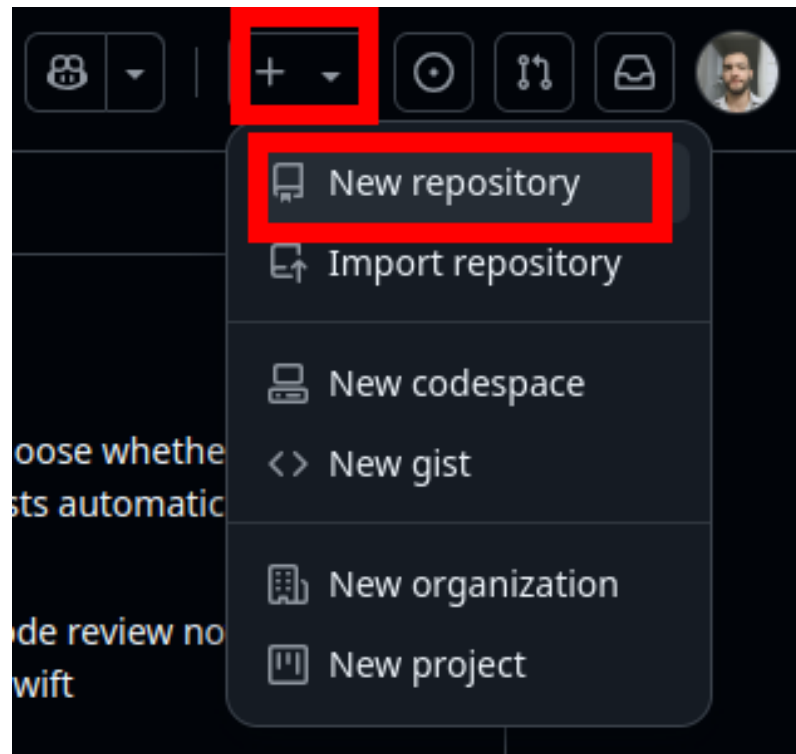
# Primeiros passos

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.

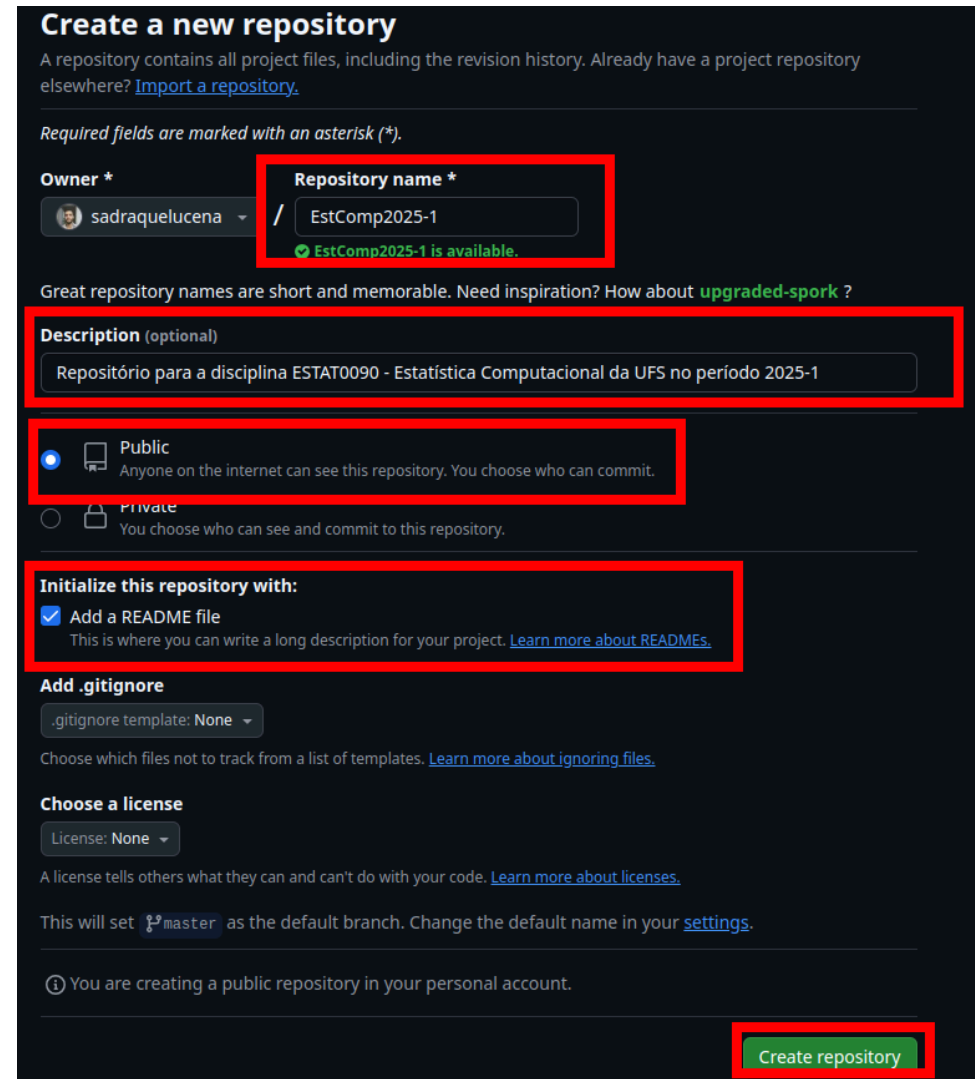
# Criando um repositório **git** no **GitHub**

- Após fazer login no GitHub, Clique em **+** no canto superior direito
- Em seguida, clique em **New repository**



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

1. Em **Repository name** dê um anome ao repositório
2. Em **Description** faça uma descrição do repositório
3. Marque a opção **Public** ou **Private**
4. Em **Initialize this repository with:** marque a opção **Add a README file**
5. 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 `main` as the default branch. Change the default name in your [settings](#).

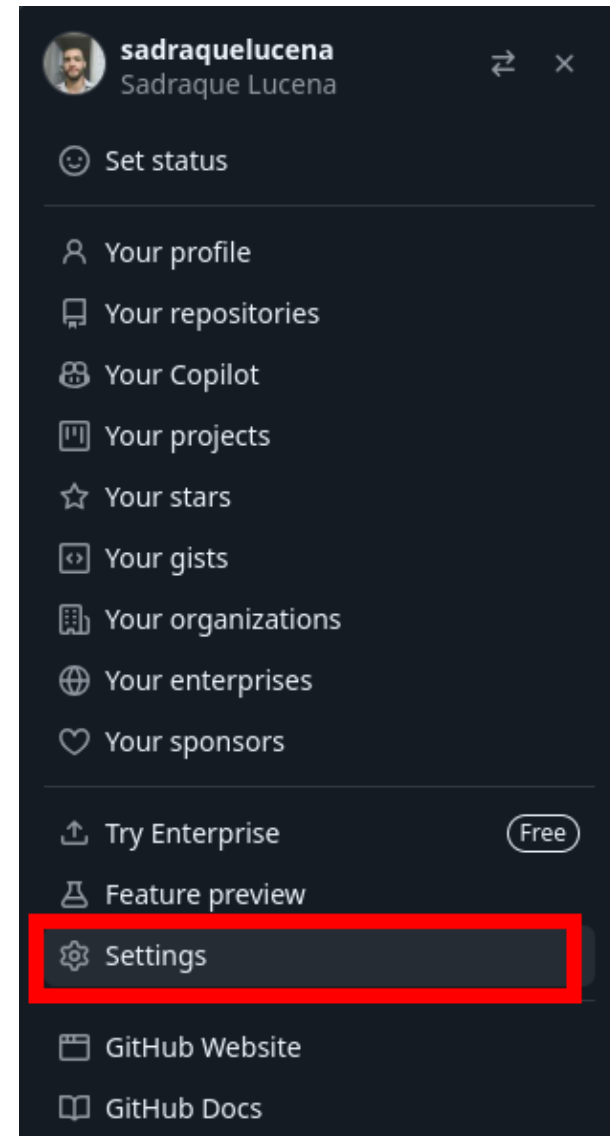
📄 You are creating a public repository in your personal account.

**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.

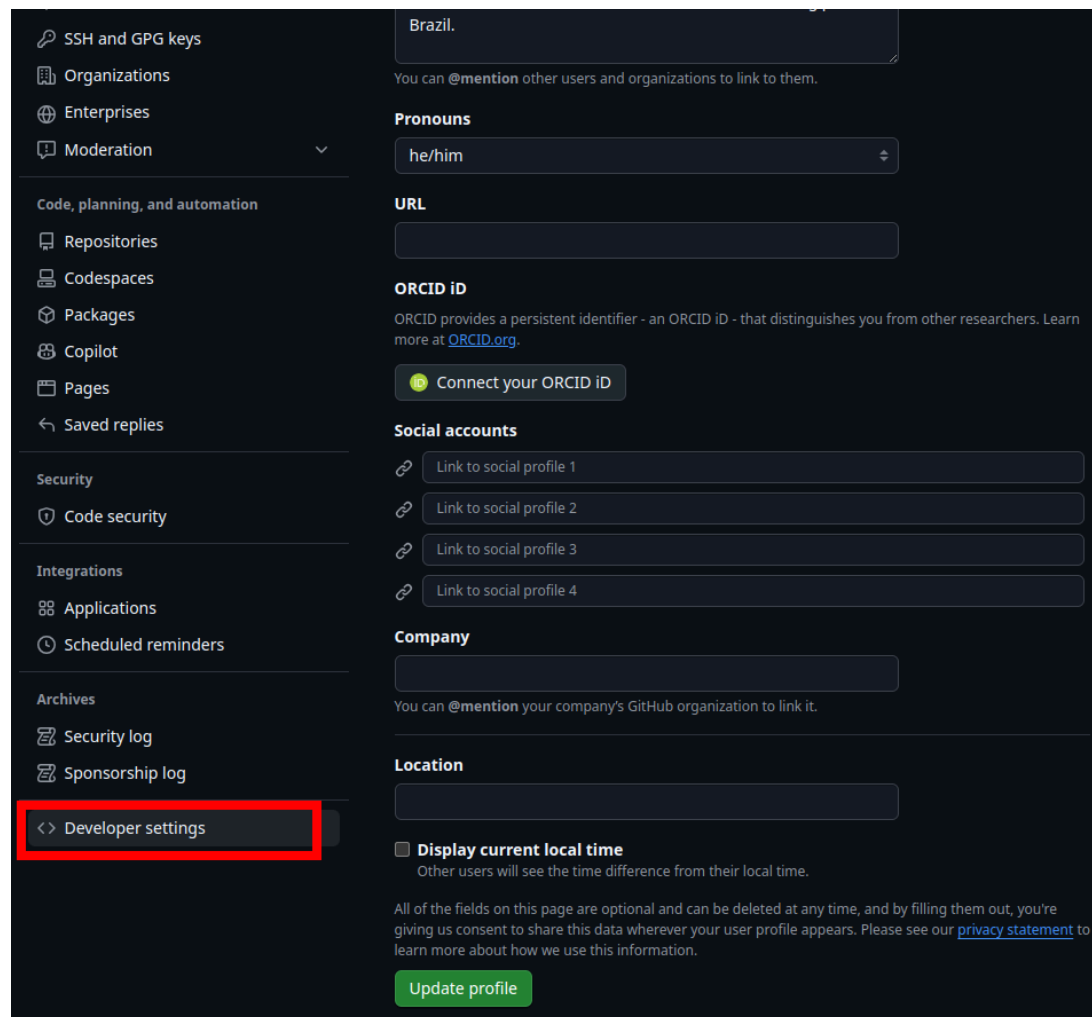
# Criando um token de acesso no GitHub

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**



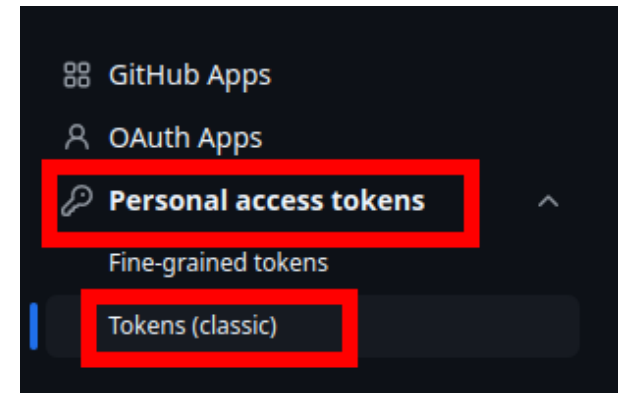
The screenshot shows the GitHub Developer settings page. The left sidebar contains various settings categories, with 'Developer settings' highlighted by a red rectangle. The main content area includes fields for 'Pronouns' (set to 'he/him'), 'URL', 'ORCID iD' (with a 'Connect your ORCID iD' button), 'Social accounts' (four 'Link to social profile' buttons), 'Company', 'Location', and a checkbox for 'Display current local time'. At the bottom, there is an 'Update profile' button.



# 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**

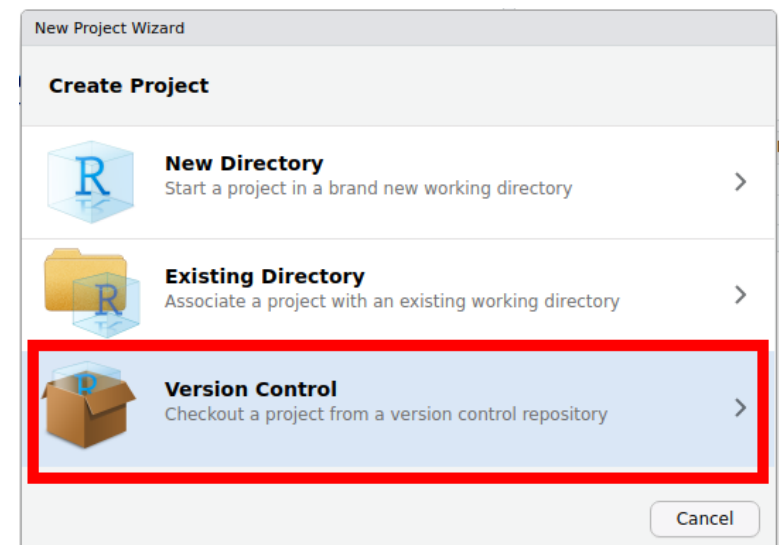
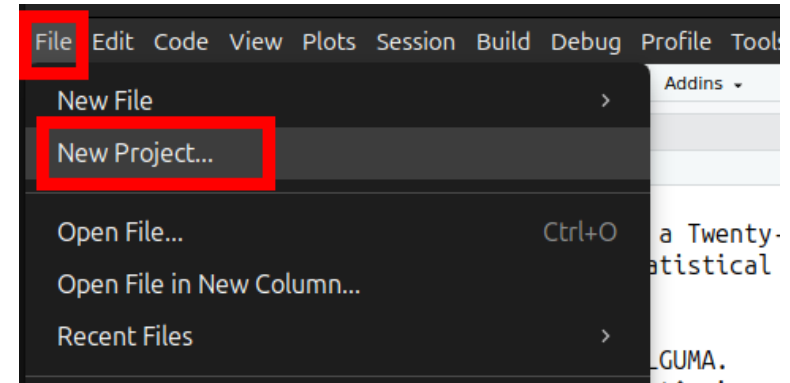
- O **RStudio** possui integração nativa com o **Git** e **GitHub**
  - Ou seja, é possível sincronizar um repositório GitHub a um repositório local
  - Isso significa que você pode ligar o repositório do GitHub (na nuvem) ao seu projeto no computador. Assim, o que você altera localmente pode ser enviado para o GitHub — e vice-versa.
- Para isso, seguimos os seguintes passos:

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

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**).

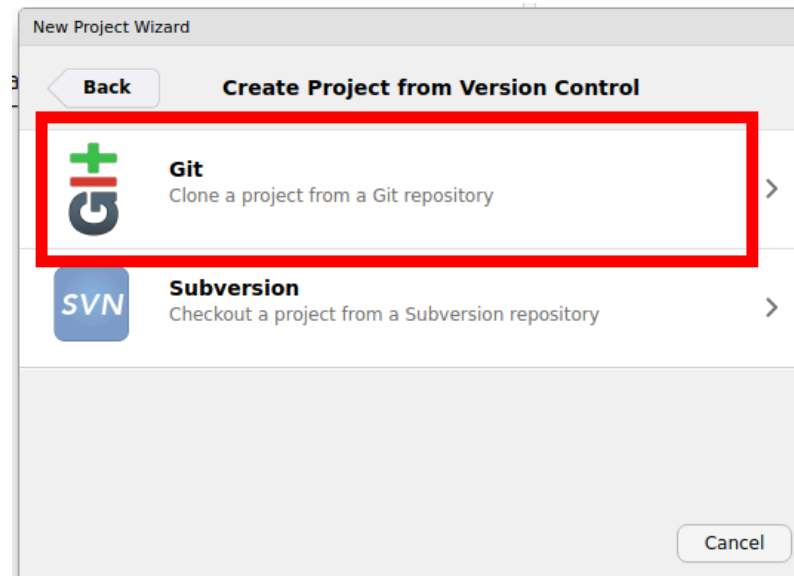
# R Studio® + GitHub: criando o projeto local

1. No canto superior direito do RStudio clique em **File > New Project**
2. Clique em **Version control**



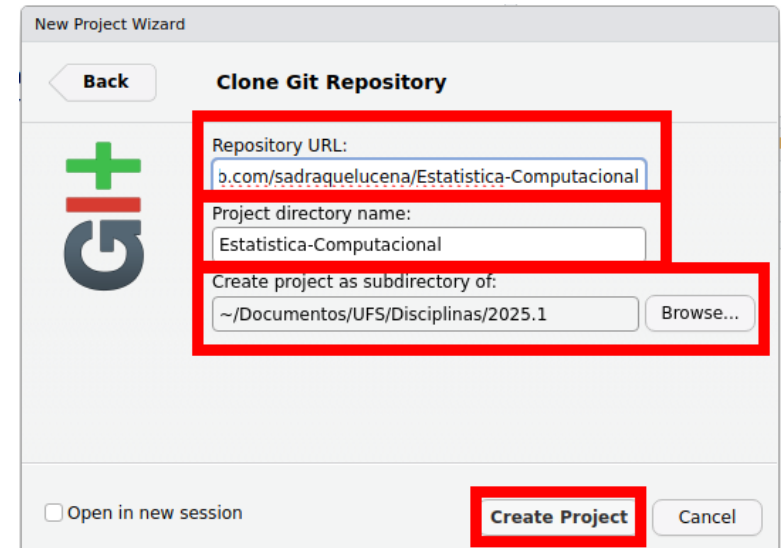
# R Studio® + GitHub: criando o projeto local

3. Clique em **Git**



# GitHub + R Studio: criando o projeto local

4. No campo **Repository URL**, cole a URL do repositório que você criou no GitHub
5. Em **Create project as subdirectory of**, escolha o diretório em que o repositório do GitHub será copiado na máquina local
6. 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).
- Uma vez que já existe o projeto na máquina local, você só precisa acessar o diretório local (pasta) e clicar no arquivo com extensão **.Rproj** para abrir o projeto no Rstudio, sem necessidade de clonar o projeto novamente na máquina.

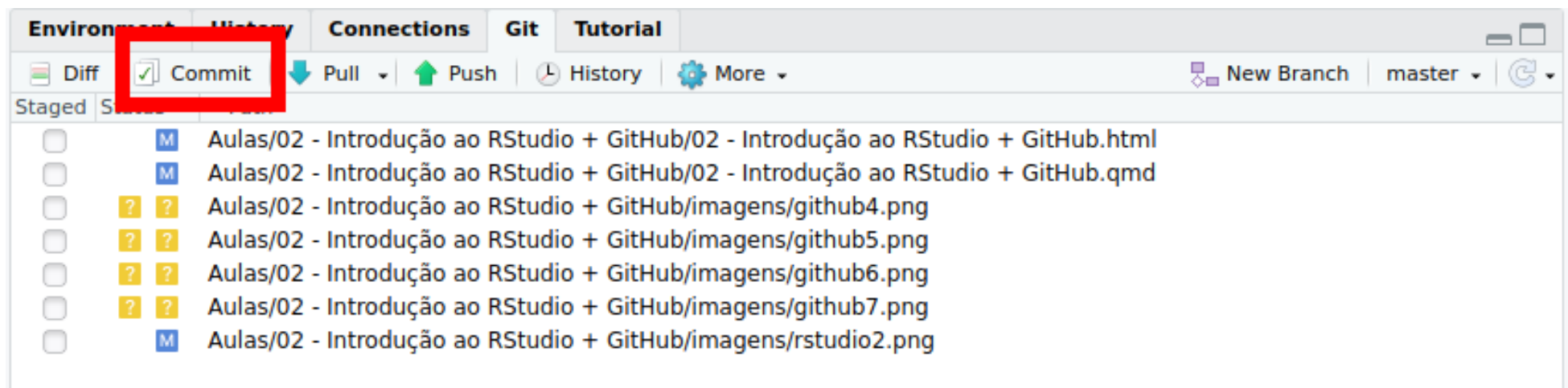


# Atualizando o repositório no GitHub

# 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:

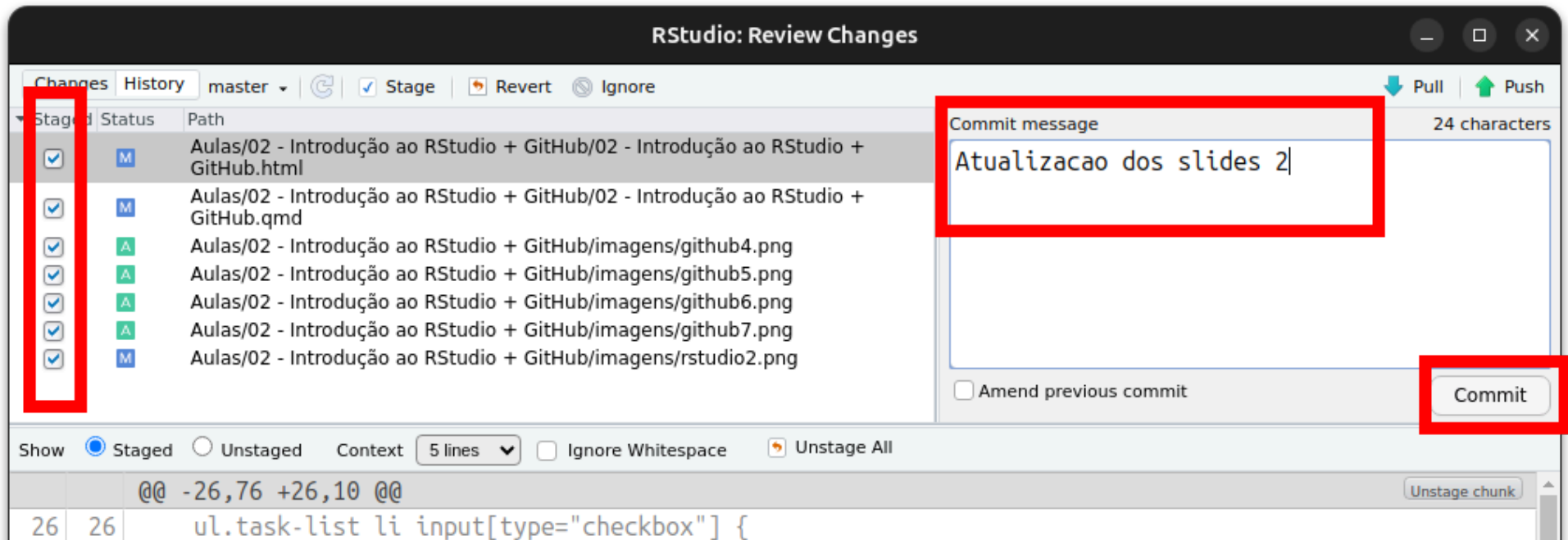
1. No quadrante superior direito clique em **Commit**



# Enviando alterações para o GitHub via RStudio®

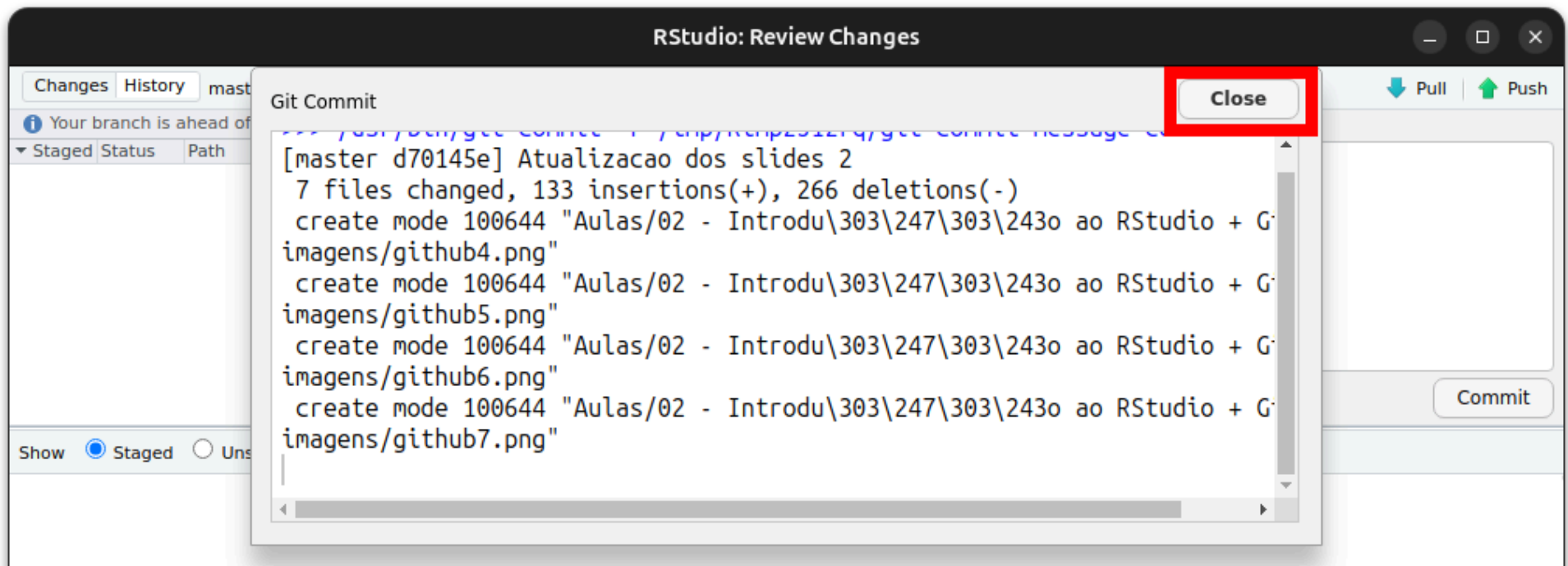
2. RStudio mostra os arquivos que foram alterados. Selecione-os.
3. No campo `Commit message` escreva um comentário contendo o que foi atualizado (sugestão: não use caracteres especiais ou acentos)
4. Clique em `Commit`

# Enviando alterações para o GitHub via RStudio®



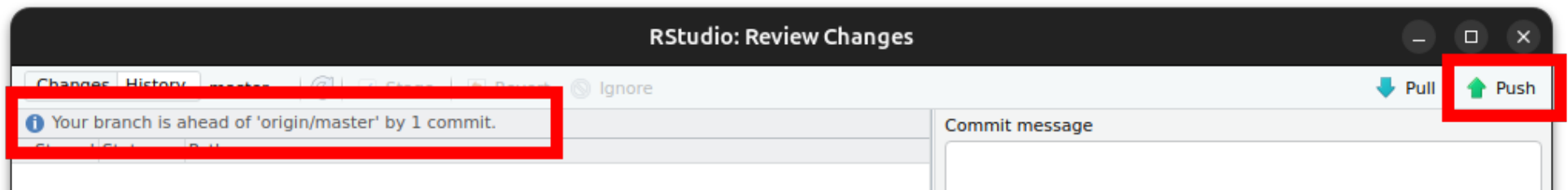
# Enviando alterações para o GitHub via RStudio®

5. Após finalizado o envio, clique em **Close**



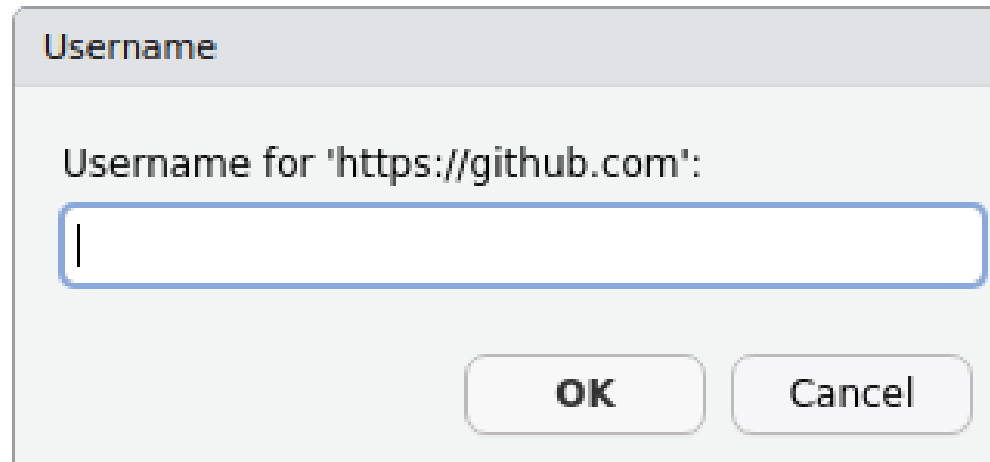
# Enviando alterações para o GitHub via RStudio®

- Note que irá aparecer a mensagem `Your branch is ahead of 'origin/master' by 1 commit` (isto indica que você tem alterações ainda não enviadas ao GitHub)
- Clique em `Push`



# Enviando alterações para o GitHub via Studio®

8. Em máquinas com Windows o Rstudio redirecionará para você fazer login na sua conta GitHub no seu navegador. Em Linux preencha o campo **Username for 'https://github.com'** coloque login e clique em **OK**



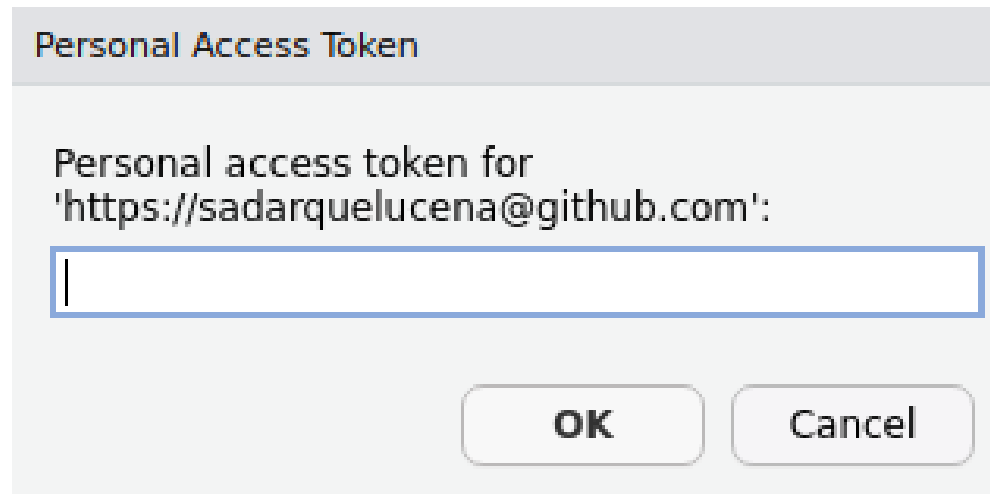
Username

Username for 'https://github.com':

OK Cancel

# Enviando alterações para o GitHub via Studio®

9. Ainda em Linux, no campo **Personal Access Token** insira o token criado no GitHub
10. Clique em **OK**



Personal Access Token

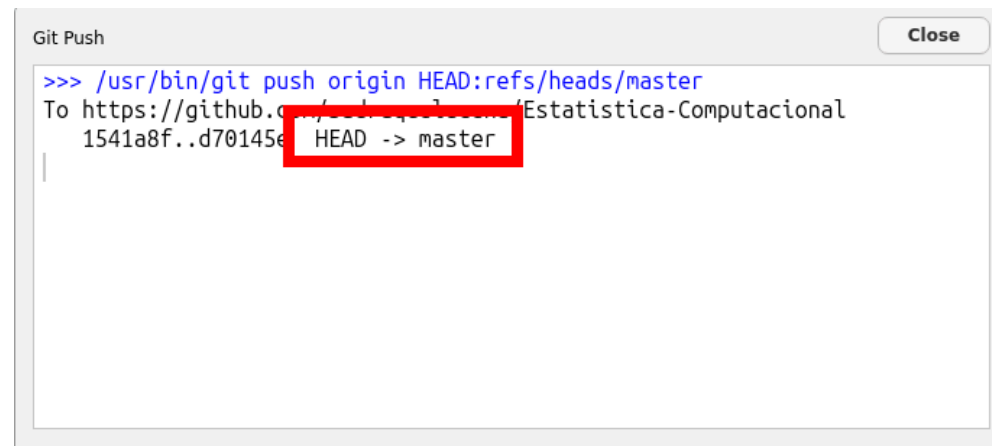
Personal access token for  
'https://sadarqu Lucena@github.com':

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/sergiopereira/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>

Criar um **repositório da disciplina do GitHub**, clonar via RStudio para a sua máquina local e fazer seu **primeiro commit**.

# Fim