

**MBA  
USP  
ESALQ**

## **DATA WRANGLING**

Prof. Dr. Wilson Tarantin Junior

**\*A responsabilidade pela idoneidade, originalidade e licitude dos conteúdos didáticos apresentados, é do professor.**

**Proibida a reprodução total ou parcial, sem autorização. Lei nº 9610/98**

# Projetos de Data Science & Analytics no GitHub

# Git

- Software utilizado para o controle de versões
- Registra as alterações feitas nos arquivos
- Vamos utilizá-lo em conjunto com o Github
- Para instalar o Git no computador, siga o tutorial disponibilizado

# Github

- Site utilizado para hospedar os arquivos
- Organizado em repositórios (pastas) que podem ser compartilhados e, inclusive, podem ser publicados
  - Útil para armazenar e compartilhar seu portfólio de projetos
- Os arquivos do computador podem ser enviados ao Github (pelo Git)

# Git: configuração inicial

- Na primeira vez em que utiliza o Git, há um cadastro inicial
  - Crie uma pasta na área de trabalho de seu computador
  - Dentro da pasta, com o botão direito do mouse, acesse **Git Bash Here**
  - Faça o seguinte comando e clique enter
  - **git config --global user.email “seu email”**
  - Na sequência, faça o seguinte comando e clique enter
  - **git config --global user.name “seu nome”**

# Git e Github

- Add e Commit

- Abra o RStudio, crie um novo script e escreva apenas # Versão 1
- Salve este arquivo na pasta com o nome Exemplo.R
- Dentro da pasta, com o botão direito do mouse, acesse **Git Bash Here**
- No Git, escreva **git init** (inicializa o Git na pasta selecionada)
- Na sequência, troque para a seguinte branch: **git branch -M main**
- Escreva **git add 'Exemplo.R'** (adiciona o arquivo para o índice)
- Para gerar versões utilize o comando **git commit -m 'título'** (são as versões)

O nome do commit, exemplo: 'Primeira\_Versão'

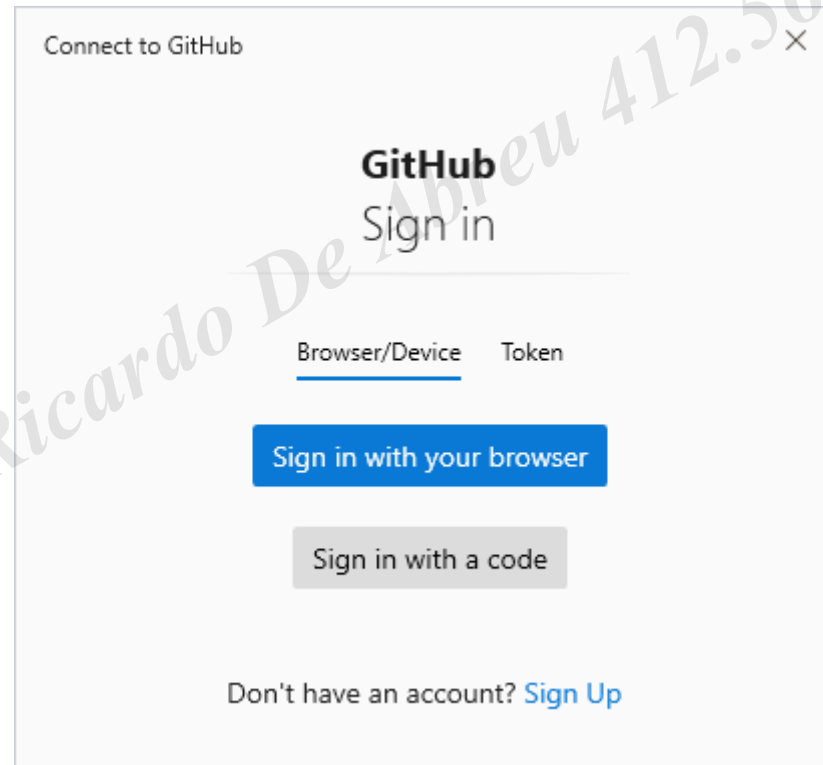
# Git e Github

- Push
  - Em seu Github, crie um novo repositório e nomeie como preferir
  - Copie o link do repositório criado
  - No Git, escreva **git remote add origin link de sua pasta**
  - Por fim, digite **git push -u origin main** (envia o arquivo para o repositório, ficando na ramificação principal)
    - Na primeira vez que for feito, solicitará a conexão inicial com o Github!
- Após atualizar, é possível verificar que o arquivo já está em seu Github!



# Git e Github: conexão inicial

- Caso seja a primeira vez que utiliza o Git, há um login



# Git e Github: conexão inicial

- Opção 1: Autenticando por meio do browser

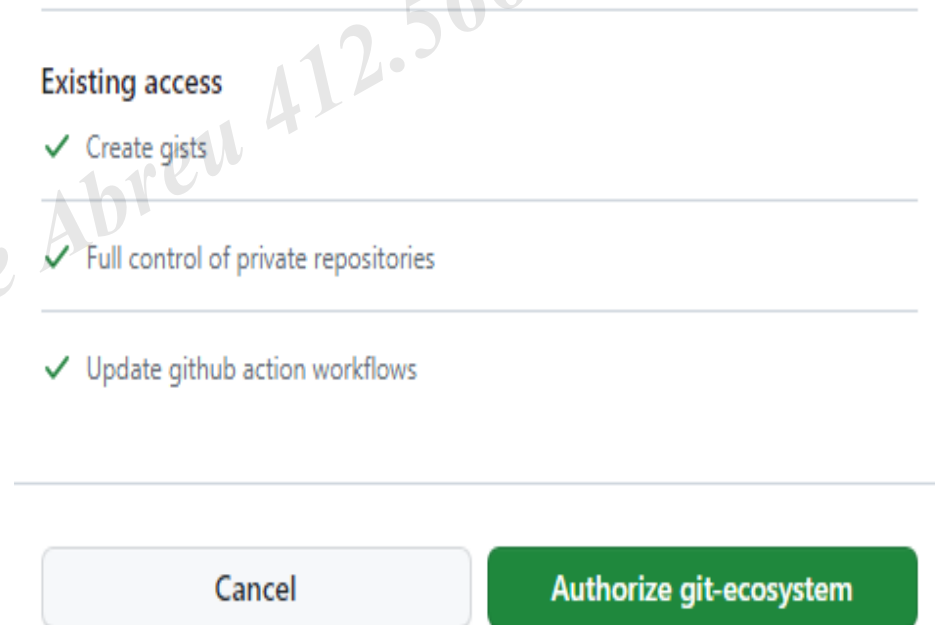


# Git e Github: conexão inicial

- Opção 2: Autenticando por meio de código



Insira o código  
fornecido neste link  
e avance



Autorize e, na  
sequência, informe  
a senha do GitHub

# Git e Github

- Criando e comparando versões
  - Abra o arquivo Exemplo.R e escreva mais uma linha: # Versão 2
  - Após salvar, feche e com o botão direito abra **Git Bash Here** na pasta
  - Utilize os mesmos procedimentos:
    - **git add 'Exemplo.R'**
    - **git commit -m 'Segunda\_Versão'**
    - **git push -u origin main**
- No Github, a nova versão já está disponível e podemos compará-las!

Note que não foi  
necessário informar  
novamente o endereço

# Git e Github

- Criando ramificações no repositório
  - Nos comandos anteriores, alteramos a ramificação principal do repositório
  - Poderíamos criar ramificações novas no Github
  - **git checkout -b 'nova'**
  - No Git, já há a indicação de mudança da “main” para a “nova”
  - Abra o arquivo Exemplo.R e escreva mais uma linha: # Versão 3
  - **git add 'Exemplo.R'**
  - **git commit -m 'Terceira\_Versão'**
  - **git push -u origin nova**

**nova** é o nome  
que demos para  
a branch

# Git e Github

- Importando repositórios (Clone e Pull)
  - Pode ser útil trazer para seu computador arquivos que estão no Github
  - Uma forma de “baixar” tais arquivos é por meio da função clone
  - Crie uma pasta em seu computador
  - Dentro da pasta, com o botão direito do mouse, abra o **Git Bash Here**
  - No Github, no repositório de interesse, clique em **code** e copie o link
  - No Git, digite **git clone link do repositório**
  - Para baixar novamente, após alterações no Github, indique **cd 'repositório'**
  - Na sequência, digite **git pull** (o arquivo foi atualizado no computador)

# Git e Github

- Importando repositórios (Clone e Pull)
  - Também podemos fazer o pull de outras branches
  - Digite: **git branch -a** (mostra as branches disponíveis)
  - Na sequência: **git checkout 'nome da outra branch'**
  - Para realizar o pull da outra branch: **git pull**
  - Note que os arquivos mudaram na pasta em seu computador

# Git e Github

- Copiando repositórios públicos (Fork)
  - É possível copiar repositórios que estão publicados no Github
  - Procure por algum tema de interesse
  - Acesse o repositório
  - No canto superior direito, existe o botão **Fork**
  - Após clicar, poderá ver o repositório em sua lista (em seu perfil)



# Git, Github e RStudio

- É possível integrar o Git, Github e RStudio
- No RStudio, clique em File → New Project → Version Control → Git
  - Em “Repository URL” basta indicar o link do repositório no Github
- Após criar um documento (R Script, R Markdown), clique em Git e faça o **commit** e, em seguida, o **push**
  - Também é possível fazer o **pull** dos arquivos do repositório que foi indicado

# Criação de Projects e Scripts R Markdown

# R Markdown

- **Introdução ao R Markdown**
- **Formatação básica do texto**
- **Inserção de fórmulas**
- **Chunks**
- **Gerando outputs (HTML; PDF, DOC)**
- **Material para referência:**
  - <https://rmarkdown.rstudio.com/index.html>

# OBRIGADO!

[linkedin.com/in/wilson-tarantin-junior-359476190/](https://www.linkedin.com/in/wilson-tarantin-junior-359476190/)