

Getting Started with Git and GitHub

A Complete Beginner's Guide for RStudio & VS Code

MBAN 5560 - Machine Learning & AI - Winter 2026

Dr. Aydede

Welcome!

This guide will walk you through everything you need to know to start using Git and GitHub with both **RStudio** and **VS Code**. Don't worry if you've never used these tools before - we'll go through each step together, with plenty of explanations along the way!

Take your time. There's no rush. If you get stuck, that's completely normal! Check the troubleshooting section at the end, or ask for help.

Part 1: Understanding Git and GitHub

What is Git?

Think of Git as a **super-powered "Track Changes"** feature (like in Microsoft Word), but for all your files. It keeps a complete history of every change you make, lets you go back to any previous version of your work, and helps you understand what changed, when, and why.

Important: Git runs on your computer. It's a separate program that both RStudio and VS Code use behind the scenes.

What is GitHub?

GitHub is a **website that stores your Git projects in the cloud**. Think of it like Google Drive or Dropbox, but specifically designed for code. It's also a social network for programmers where you can share and collaborate, and a backup system that keeps your work safe online.

Why Should You Care?

- **Backup:** Your work is safely stored in the cloud - never lose your homework again!
- **Version History:** Made a mistake? No problem! Go back to any previous version
- **Collaboration:** Work with classmates without emailing files back and forth
- **Professional Skill:** Every tech company uses Git - it's a must-have skill!
- **Portfolio:** Your GitHub profile becomes a portfolio showing your work to future employers

Part 2: Setting Up (One-Time Setup)

Step 1: Create a GitHub Account

1. Open your web browser and go to github.com
2. Click the big "**Sign up**" button in the top-right corner
3. Enter your email address (use your university email for free Pro features!)
4. Create a strong password
5. Choose a username - **pick something professional!** Future employers will see this
Good examples: jane-smith, jsmith2026
Avoid: xXx_CodingNinja_xXx, partyanimal99
6. Complete the verification puzzle (prove you're human!)
7. Click "**Create account**"
8. Check your email inbox and click the verification link GitHub sends you

You now have a GitHub account!

Step 2: Install Git on Your Computer

Remember: **Git is NOT the same as RStudio or VS Code.** It's a separate program that must be installed first.

For Windows Users:

1. Go to git-scm.com/downloads
2. Click "**Windows**" - the download starts automatically
3. Find the downloaded file (usually in Downloads folder) - called something like Git-2.x.x-64-bit.exe
4. Double-click to run the installer
5. **Click "Next" through ALL the screens** - the default options are perfect for beginners!
6. Click "**Install**"
7. When it's done, click "**Finish**"

For Mac Users:

1. Go to git-scm.com/downloads
2. Click "**macOS**"
3. The easiest method is using **Homebrew**. Open Terminal (press Cmd+Space, type "Terminal", press Enter)
4. Copy and paste this command, then press Enter:

```
/bin/bash -c "$(curl -fsSL  
https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

5. After Homebrew finishes installing, type this and press Enter:

```
brew install git
```

Verify Git is Installed:

Open a terminal (Git Bash on Windows, Terminal on Mac) and type:

```
git --version
```

If you see a version number (like git version 2.43.0), Git is installed correctly!

Step 3: Tell Git Who You Are

IMPORTANT: This step is REQUIRED! Without it, you'll get errors later. Git needs to know your name and email to track who made each change.

1. Open a terminal (Git Bash on Windows, Terminal on Mac)
2. Type this command (replace with YOUR name) and press Enter:

```
git config --global user.name "Your Full Name"
```

For example: git config --global user.name "Jane Smith"

3. Type this command (replace with YOUR email) and press Enter:

```
git config --global user.email "your.email@example.com"
```

Important: Use the SAME email address you used to create your GitHub account!

Verify your identity is set by typing:

```
git config --global user.name git config --global user.email
```

You should see your name and email displayed back to you.

Step 4: Create a Personal Access Token (PAT)

A Personal Access Token is like a **special password** that allows RStudio and VS Code to talk to your GitHub account. GitHub no longer accepts regular passwords for this.

Generate Your Token:

1. Log in to GitHub at github.com
2. Click your profile picture in the top-right corner
3. Click "**Settings**" from the dropdown menu
4. Scroll ALL the way down on the left sidebar and click "**Developer settings**" (at the very bottom!)
5. Click "**Personal access tokens**" in the left sidebar
6. Click "**Tokens (classic)**"
7. Click "**Generate new token**" button, then "**Generate new token (classic)**"
8. You might need to enter your GitHub password again
9. Fill in the token settings:
 - Note:** Give it a name like "RStudio-VSCode-Access" or "My-Laptop-Git"
 - Expiration:** Choose **90 days** (or "Custom" to match your course duration)
 - Select scopes:** Check the box next to **repo** (this will auto-select all sub-options)
 - Also check **user:email** (scroll down to find this under "user")
10. Scroll down and click "**Generate token**"

CRITICAL - COPY YOUR TOKEN RIGHT NOW! GitHub will show you the token **ONLY ONCE**. It looks like a long string starting with ghp_. Click the copy button, then paste it somewhere safe IMMEDIATELY (password manager, secure note, etc.). If you lose this token, you can always generate a new one.

Part 3: Connecting RStudio to GitHub

Step 1: Tell RStudio Where Git Is

1. Open RStudio
2. Go to **Tools -> Global Options** (on Mac: **RStudio -> Preferences**)
3. Click "**Git/SVN**" in the left sidebar
4. Make sure "**Enable version control interface for RStudio projects**" is checked
5. Check that the "**Git executable**" field shows a path:
 - Windows: Usually C:/Program Files/Git/bin/git.exe
 - Mac: Usually /usr/bin/git or /usr/local/bin/git

If it's empty, click "Browse" and navigate to find the git file.

6. Click "**OK**"

Step 2: Store Your PAT (Recommended Method)

1. In the RStudio **Console** (bottom-left panel), install the required packages:

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

Press Enter and wait for them to install (this might take a minute).

2. Now store your PAT by running:

```
gitcreds::gitcreds_set()
```

3. When prompted, **paste your Personal Access Token** and press Enter

Your PAT is now saved! RStudio will use it automatically.

Step 3: Verify Everything Works

Run this in the Console to check your setup:

```
library(usethis) usethis::git_sitrep()
```

Look for: Name and Email showing your information, GitHub user showing your username, and Personal access token showing .

Part 4: Connecting VS Code to GitHub

VS Code has excellent built-in Git support and makes connecting to GitHub very easy!

Step 1: Open VS Code

1. Download VS Code from code.visualstudio.com if you haven't already
2. Install and open VS Code

Step 2: Install the GitHub Extension (Recommended)

1. Click the **Extensions** icon in the left sidebar (it looks like 4 squares)
2. Search for "**GitHub Pull Requests and Issues**"
3. Click "**Install**" on the extension published by GitHub

This extension provides much richer GitHub integration including managing pull requests and issues directly within VS Code!

Step 3: Sign In to GitHub

VS Code makes this easy with automatic sign-in:

1. Look at the bottom-left corner of VS Code - you should see a **person icon**
2. Click on it and select "**Sign in to Sync Settings**" or "**Turn on GitHub Pull Requests...**"
3. A browser window will open asking you to **authorize VS Code**
4. Click "**Authorize**"
5. You'll be redirected back to VS Code - you're now connected!

Alternative method: If the automatic sign-in doesn't work, VS Code will prompt you for your GitHub username and password when you try to clone or push. Enter your username and use your **PAT** as the password.

VS Code is now connected to GitHub!

Part 5: Working with Repositories

Option A: Clone an Existing Repository (Most Common)

"**Cloning**" means downloading a copy of a repository from GitHub to your computer.

First, Get the Repository URL:

1. Go to the GitHub repository page (your instructor will give you the link)
2. Click the green "**Code**" button
3. Make sure "**HTTPS**" is selected (not SSH)
4. Click the **copy button** to copy the URL

The URL looks like: <https://github.com/username/repository-name.git>

Clone in RStudio:

1. Go to **File -> New Project**
2. Click "**Version Control**"
3. Click "**Git**"
4. Paste the URL into "**Repository URL**"
5. Choose where to save it (e.g., Documents folder)
6. Click "**Create Project**"

Look for the "**Git**" tab in the top-right panel - that's your Git control center!

Clone in VS Code:

1. Press **Ctrl+Shift+P** (or Cmd+Shift+P on Mac)
2. Type "**Git: Clone**" and select it
3. Paste the URL and press Enter
4. Choose where to save it
5. Click "**Open**" when prompted

The **Source Control** panel (left sidebar) is your Git control center!

Part 6: The Daily Workflow

The Git Workflow Cycle

This is what you'll do every time you work on your project: **PULL -> EDIT -> STAGE -> COMMIT -> PUSH**

Step	What It Does	When to Do It
PULL	Download latest changes from GitHub	Always before you start working!
EDIT	Do your actual work (write code, edit files)	This is your normal work
STAGE	Select which changes you want to save	After making changes you want to keep
COMMIT	Save a snapshot with a description	After staging - like saving a checkpoint
PUSH	Upload your commits to GitHub	When ready to share/backup your work

How to Do Each Step in RStudio:

- 1. PULL:** Click the "Git" tab (top-right panel) -> Click the blue down arrow labeled "Pull"
- 2. EDIT:** Edit your files normally and save them (Ctrl+S or Cmd+S)
- 3. STAGE:** Go to the "Git" tab -> Check the box next to each file you want to include
- 4. COMMIT:** Click the "Commit" button -> Type a message describing what you changed -> Click "Commit"
- 5. PUSH:** Click the green up arrow labeled "Push"

How to Do Each Step in VS Code:

- 1. PULL:** Click the Source Control icon (left sidebar) -> Click the three dots menu -> "Pull"
- 2. EDIT:** Edit your files normally and save them (Ctrl+S or Cmd+S)
- 3. STAGE:** Go to Source Control panel -> Click the + icon next to each file to stage it
- 4. COMMIT:** Type your message in the text box at the top -> Click the checkmark or press Ctrl+Enter
- 5. PUSH:** Click the three dots menu -> "Push" (or click "Sync Changes" button)

Writing Good Commit Messages:

Your commit message should briefly describe what you changed:

- GOOD: "Added analysis for Question 3"
- GOOD: "Fixed bug in data cleaning script"
- BAD: "stuff" (too vague!)
- BAD: "asdfasdfs" (meaningless!)

Quick Reference Card

Action	What It Does	RStudio	VS Code
Clone	Download a repo	File -> New Project -> Version Control	Ctrl+Shift+P -> "Git: Clone"
Pull	Get latest changes	Git tab -> Blue arrow down	Source Control -> Menu -> Pull
Stage	Select files	Git tab -> Check boxes	Source Control -> Click + icon
Commit	Save snapshot	Commit button -> Type message	Type message -> Click checkmark
Push	Upload to GitHub	Git tab -> Green arrow up	Menu -> Push (or Sync Changes)

Part 7: Troubleshooting Common Issues

"Git executable not found"

Problem: Git may not be installed, or your IDE can't find it.

Solution:

1. Reinstall Git from git-scm.com
2. Restart your computer
3. In RStudio: Tools -> Global Options -> Git/SVN -> Browse to find git
4. In VS Code: It usually finds Git automatically after restart

"Authentication failed" or "Invalid credentials"

Problem: Your PAT is missing, expired, or incorrect.

Solution:

1. Generate a new PAT on GitHub (follow Step 4 again)
2. In RStudio: Run `gitcreds::gitcreds_set()` and paste the new token
3. In VS Code: When prompted, enter your username and use the PAT as password

"Repository not found" when cloning

Problem: The URL is wrong, or you don't have access.

Solution:

- Check that the URL is correct (copy it again from GitHub)
- Make sure you have access to the repository (it might be private)
- Verify you're logged into the correct GitHub account

"Updates were rejected" when pushing

Problem: Someone else made changes, or you made changes on another computer.

Solution:

1. Click **Pull** first to get the latest changes
2. Then try **Push** again

Can't see the Git tab/panel

Problem: Your project isn't linked to Git.

Solution:

- RStudio: Make sure you opened an RStudio Project, not just a file. The project must be cloned from GitHub.

- VS Code: Make sure you opened a folder that contains a Git repository (has a hidden .git folder).

Getting More Help

Helpful Resources:

- GitHub Documentation: docs.github.com
- Happy Git with R (excellent resource for R users): happygitwithr.com
- VS Code Git Documentation: code.visualstudio.com/docs/sourcecontrol
- Git Cheat Sheet: education.github.com/git-cheat-sheet

Don't forget: Ask your instructor or TA if you get stuck! That's what they're there for.

You're Ready!

You now have all the tools you need to start using Git and GitHub with both RStudio and VS Code.

Remember: Practice makes perfect. The more you use these tools, the more natural they'll become!

Last updated: January 2026