Getting Started with GitHub & Git Bash — A First-Timer's Guide

---

Part 1: Create a GitHub Account

1. **Go to GitHub** Open your browser and navigate to https://github.com.
2. **Click "Sign up"** (top-right corner).
3. **Fill in your details:**
   - Enter your email address
   - Create a password (at least 15 characters, or 8 with a number + lowercase letter)
   - Choose a username — this becomes your public profile name (e.g. `tina-analytix`)
   - Verify you're a human (solve the puzzle)
4. **Verify your email** GitHub will send a code to your email — enter it to activate your account.
5. **Choose the Free plan** (it's more than enough to start with).

You now have a GitHub account! Your profile lives at `https://github.com/your-username`.

Part 2: Install Git Bash (Windows)

> If you're on Mac or Linux, Git is usually pre-installed. Open Terminal and type `git --version` to check.

1. **Download Git for Windows** Go to https://git-scm.com/downloads/win and download the installer.
2. **Run the installer**
   - Accept the defaults for most screens
   - **Important screen:** When asked about the default editor, you can select "Use Visual Studio Code" or "Use Nano" (easier than Vim for beginners)
   - When asked about PATH, choose **"Git from the command line and also from 3rd-party software"**
   - Finish the installation
3. **Open Git Bash**
   - Press the Windows key, type `Git Bash`, and open it
   - You should see a terminal window with a blinking cursor
4. **Verify the installation** Type the following and press Enter:

```bash
git --version
```

You should see something like `git version 2.47.1` (the number may differ).

Part 3: Configure Git (One-Time Setup)

Before using Git, tell it who you are. This info gets attached to every change you make.

1. **Set your name** (use the same name as your GitHub profile):

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

2. **Set your email** (use the same email you registered on GitHub):

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

3. **Verify your config:**

```bash
git config --list
```

You should see your name and email listed.

Part 4: Clone a Repo FROM GitHub TO Your Computer

"Cloning" means downloading a complete copy of a repository (project) from GitHub to your local machine.

Step-by-step:

1. **Find the repo you want to clone on GitHub** For example, go to any public repo like: `https://github.com/hadley/r4ds`
2. **Copy the repo URL**
   - Click the green **"<> Code"** button
   - Make sure **HTTPS** is selected (the default tab)
   - Click the clipboard icon to copy the URL
   - It will look like: `https://github.com/hadley/r4ds.git`
3. **Open Git Bash**
4. **Navigate to where you want the project to live** For example, to put it on your Desktop:

```bash
cd ~/Desktop
```

> **Tip:** `cd` means "change directory". The `~` symbol refers to your home folder (e.g. `C:/Users/Tina`).

5. **Clone the repo:**

```bash
git clone https://github.com/hadley/r4ds.git
```

6. **Wait for the download to finish.** You'll see output like:

```
Cloning into 'r4ds'...
remote: Enumerating objects: 15432, done.
remote: Total 15432 (delta 0), reused 0 (delta 0)
Receiving objects: 100% (15432/15432), 48.21 MiB | 5.12 MiB/s, done.
```

7. **Enter the project folder:**

```bash
cd r4ds
```

8. **Check that everything is there:**

```bash
ls
```

This lists all the files in the repo. You're done — the project is on your machine!

Part 5: Push a Project FROM Your Computer TO GitHub

This is for when you have a folder on your computer (like your Tutorial 3 project) and you want to upload it to a new GitHub repository.

Step A: Create an empty repo on GitHub

1. Go to https://github.com and log in.
2. Click the **"+"** icon (top-right) → **"New repository"**.

3. Fill in:
    - **Repository name:** e.g. `econometrics-871-tutorial3`
    - **Description:** (optional) e.g. "Unit root testing tutorial in R"
    - **Public** or **Private** — your choice
    - ⚠️ **Do NOT** tick "Add a README file" (we want an empty repo)
4. Click **"Create repository"**.
5. You'll see a page with setup instructions — keep this page open, you'll need the URL.

Step B: Initialise Git in your local project folder

1. **Open Git Bash.**
2. **Navigate to your project folder:**

```bash
cd ~/Desktop/tutorial3_project
```

> Replace the path with wherever your project actually lives.

3. **Initialise a Git repository:**

```bash
git init
```

This creates a hidden `.git` folder that tracks all your changes.

4. **Check the status** (see what files Git has detected):

```bash
git status
```

You'll see all your files listed in red under "Untracked files".

Step C: Stage and commit your files

5. **Stage all files** (tell Git you want to include everything):

```bash
git add .
```

> The `.` means "everything in this folder".

6. **Commit** (save a snapshot with a message):

```bash
git commit -m "Initial commit: Tutorial 3 project files"
```

> The `-m` flag lets you write your commit message inline. Always write something meaningful.

Step D: Connect to GitHub and push

7. **Rename your default branch to `main`** (GitHub's standard):

```bash
git branch -M main
```

8. **Link your local repo to the GitHub repo** (copy the URL from the GitHub page you kept open):

```bash
git remote add origin https://github.com/your-username/econometrics-871-tutorial3.git
```

9. **Push your code to GitHub:**

```bash
git push -u origin main
```

> The `-u` flag sets `origin main` as the default so next time you can just type `git push`.

10. **Authenticate if prompted:**
    - GitHub no longer accepts passwords in Git Bash
    - A browser window will pop up asking you to sign in to GitHub — click **"Authorize"**
    - If that doesn't happen, you'll need a **Personal Access Token** (see the Quick Reference below)
11. **Refresh your GitHub repo page** — your files are now online! 🎉

Quick Reference: Common Git Bash Commands

Command	What it does
`git clone <url>`	Download a repo from GitHub

Command	What it does
`git init`	Turn a folder into a Git repository
`git status`	See what's changed since last commit
`git add .`	Stage all changed files
`git add filename.R`	Stage a specific file
`git commit -m "message"`	Save a snapshot of staged changes
`git push`	Upload commits to GitHub
`git pull`	Download latest changes from GitHub
`git log --oneline`	View commit history (compact)
`cd foldername`	Move into a folder
`cd ..`	Move up one folder
`ls`	List files in the current folder
`pwd`	Print the current folder path

Troubleshooting: Personal Access Token (if browser auth doesn't work)

If Git Bash asks for a password and the browser popup doesn't appear:

1. Go to **GitHub → Settings → Developer settings → Personal access tokens → Tokens (classic)** Direct link: https://github.com/settings/tokens
2. Click **"Generate new token (classic)"**
3. Give it a name like `git-bash-access`
4. Tick the **`repo`** scope (full control of repositories)
5. Click **"Generate token"**
6. **Copy the token immediately** (you won't see it again!)
7. When Git Bash asks for your password, **paste the token** instead of your GitHub password

---

Visual Summary: The Git Workflow

```
YOUR COMPUTER                    GITHUB (cloud)
```

```
──────────────          ──────────────

Working Directory          Remote Repository
(your project files)      (github.com/you/repo)
     │                        ▲
     │  git add .               │
     ▼                        │
Staging Area                    │
     │                       │
     │  git commit -m "msg"           │
     ▼                        │
Local Repository ──── git push ──────────────► │
     ▲                        │
     │ ◄──────────── git pull ───────────────── │
```

You're all set! Start small — commit often, write clear messages, and don't be afraid to experiment.