In-Class Problem Set: Environment Setup + Git/GitHub Workflow (R)

Goal. Set up a reproducible workflow on your machine so you can run course code, save outputs, and submit work through GitHub.

What to submit (in your GitHub repo).

- A short write-up: `outputs/setup_check.md`
- A screenshot file: `outputs/terminal_proof.png` (or `.jpg`)
- A text file created by R: `outputs/r_check.txt`

Rules.

- Work inside an **R Project**.
- Use the **Terminal tab** for Git commands (not the R Console).
- If something breaks: read the error, check your paths, ask a neighbor, then ask for help.

Questions

1. **Install / verify your toolchain (proof required).**
   (a) Confirm you can open **RStudio** and run R code in the **Console**.
   (b) In the Console, run:

   ```
   R.version.string
   ```

   (c) **Proof:** Create a file `outputs/r_check.txt` that contains your R version string.

   Hint: Use `writeLines(...)` to write text to a file.
2. **Create an R Project (proof required).**
   (a) Create an R Project folder for this course on your computer.
   (b) Open the `.Rproj` file so RStudio is working inside that project.
   (c) In the Console, run:

   ```
   getwd()
   ```

   (d) **Proof:** In `outputs/setup_check.md`, paste the output of `getwd()` and explain (1 sentence) what it means.
3. **Create a reproducible folder structure.**
   (a) Inside your R Project, ensure these folders exist:
   - `data/`
   - `scripts/`
   - `outputs/`
   - `figures/`

   (b) Write R code (in the Console or a script) that creates these folders without errors.
   (c) **Proof:** In `outputs/setup_check.md`, include the output of:

   ```
   list.files()
   ```

showing the four folders.

4. **Set up Git and confirm it works (proof required).**
   (a) Open the **Terminal tab** in RStudio.
   (b) Run:

   ```
   git --version
   ```

   (c) Configure your name and email (use the same email as your GitHub account):

   ```
   git config --global user.name "YOUR NAME"
   git config --global user.email "YOUR EMAIL"
   ```

   (d) **Proof:** In `outputs/setup_check.md`, paste the output of:

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

5. **Clone the course repository and run a test script.**
   (a) In the Terminal tab, navigate to where you want the repo to live (or do this in File Explorer/Finder).
   (b) Clone the course repo:

   ```
   git clone <COURSE_REPO_URL>
   ```

   (c) Open the cloned folder as an R Project (if an `.Rproj` exists) or create one.
   (d) Run a simple test in the Console (example):

   ```
   1 + 1
   ```

   (e) **Proof:** Take a screenshot showing the Terminal tab in RStudio and save it as `outputs/terminal_proof.`

6. **Commit and push your first submission (proof required).**
   (a) Create your write-up file: `outputs/setup_check.md`. Include:
   - Your `getwd()` output
   - Your Git name/email confirmation outputs
   - A 2–3 sentence reflection: what broke (if anything) and what step was least intuitive

   (b) In the Terminal tab, run the Git workflow:

   ```
   git status
   git add .
   git commit -m "Setup: environment + project + git"
   git push
   ```

   (c) **Proof:** In `outputs/setup_check.md`, paste:
   - the output of `git status` after your commit (clean working tree), and
   - either (a) a screenshot of GitHub showing the commit, or (b) the commit hash from:

     ```
     git log -1
     ```

Checklist (before you leave)

- RStudio opens and runs R code
- You are working inside an R Project
- `outputs/` and `figures/` exist
- Git works in the Terminal tab
- You cloned the course repo
- Your write-up and screenshot are committed and pushed to GitHub