3 - Core Module 1: Project Organization & Reproducible Setup

Chris Reudenbach

Goals: - Structure spatial analysis projects for long-term reproducibility - Organize folders, scripts, metadata and versioning workflows - Integrate external GIS tools via `link2GI` **without** using `setwd()` - Connect local projects with GitHub for collaboration and backup

FAIR Principles First

Ensuring FAIR (Findable, Accessible, Interoperable, Reusable) data and project workflows is a priority. This is achieved through:

- **Project structure** with consistent folder and script conventions
- **Dynamic data acquisition** using `download.file()` or APIs
- **Scripted preprocessing pipelines** (`targets`, `renv`) for reproducibility
- **Metadata recording** via YAML, README files, and code headers
- **Versioning** with Git/GitHub to track changes and share openly
- **Optional use of `link2GI`** for GIS project structure bootstrapping and integration with GRASS, SAGA, and QGIS

Checklist Example

Principle	Applied to…	Example Tool/Action
Findable	Source datasets	`metadata/project_info.yaml`
Accessible	Public download scripts	`download.file()` / manual links
Interoperable	CRS, format harmonization	`terra::project()`, GRASS/RSAGA
Reusable	Clear scripts & metadata	`scripts/*.R`, YAML, Git commits

Working Directory Best Practices

- **Always use RStudio Projects** (`.Rproj`) to define the project root.
- **Never use** `setwd()` inside scripts – it breaks reproducibility across machines.
- Use `here::here("subfolder","file.ext")` to build paths relative to the project root.
- Keep all data and outputs **inside** the project folder (no absolute `C:/...` paths).
- Document required folders once (e.g. `data/raw`, `data/processed`) and create them via script.

Example:

```
library(here)

# points to the project root (where the .Rproj lives)
here::here()

# path to raw DGM file (no setwd needed)
dgm_path  <- here::here("data", "raw", "dgm1_burgwald.tif")
radolan_path <- here::here("data", "raw", "radolan_rw_20250101.tif")
```

Using this pattern ensures that **all groups** can clone a GitHub repo, run the scripts, and obtain the same results without manual path tweaking.

Best-practice Project Setup (RStudio Project):

> Assumption: You create a new RStudio Project first
> *File → New Project → New Directory → New Project*
> This defines the project root and makes `here::here()` work. No `setwd()` is used.

```
# 1. Initialize reproducible environment
renv::init()

# 2. Initialize link2GI project structure at the current project root
library(here)
library(link2GI)

proj <- initProj(
  projRootDir = here::here(),
  projFolders = c("data/raw", "data/processed", "outputs/figures", "scripts", "metadata", "do
  path_check  = TRUE
)

# 3. Ensure folders exist (idempotent)
```

```r
library(fs)
fs::dir_create(c(
  here::here("data/raw"),
  here::here("data/processed"),
  here::here("outputs/figures"),
  here::here("scripts"),
  here::here("metadata"),
  here::here("docs")
))

# 4. Save project metadata (YAML)
cat("""
title: "Rainfall Stratification - Group X"
authors: ["Name1", "Name2"]
date_started: 2025-11-15
source_datasets:
  - DGM1 Hesse 10m
  - CORINE Land Cover 2018
  - RADOLAN RW 1h
""", file = here::here("metadata", "project_info.yaml"))

# 5. Basic Git/GitHub integration (no setwd needed)
# A) Command line (inside project folder):
#   git init
#   git add .
#   git commit -m "Initial project setup"
#   git remote add origin https://github.com/<user>/burgwald_rainnet.git
#   git branch -M main
#   git push -u origin main
#
# B) RStudio Git tab:
#   - Enable Git when creating the project
#   - Use *Commit* and *Push* buttons to sync with GitHub
```

Data Retrieval Template (Open Sources): - DGM: https://gdz.bkg.bund.de (DGM1-DE, GeoTIFF) - CORINE: https://land.copernicus.eu - RADOLAN: https://opendata.dwd.de (RW-composites) - Sentinel-2: via `sen2r` or ESA Copernicus Browser - Stream gauges: Hessisches Landesamt für Naturschutz (HLNUG)

Example scripted download (to be adapted by students):

```r
# Example: download one DGM tile (students replace with actual URL)

# URL placeholders - to be replaced with real links from the data portals
url_dgm  <- "https://example.org/path/to/dgm1_burgwald.tif"
url_rad  <- "https://example.org/path/to/radolan_rw_20250101.gz"

# Local paths in the project structure
dgm_file <- here::here("data", "raw", "dgm1_burgwald.tif")
rad_file <- here::here("data", "raw", "radolan_rw_20250101.gz")

# Download (only if file does not yet exist)
if (!file.exists(dgm_file)) download.file(url_dgm, destfile = dgm_file, mode = "wb")
if (!file.exists(rad_file)) download.file(url_rad, destfile = rad_file, mode = "wb")
```

Homework after Session 1

Each group must:

1. **Create a new RStudio Project** for their rainfall network topic.
2. **Initialize `renv`** and commit the lockfile to Git.
3. **Run the `link2GI::initProj()`-based setup** (or equivalent) to create the folder structure.
4. **Create a metadata file** (`metadata/project_info.yaml`) with:
   - project title, authors, start date,
   - planned datasets (DGM, LC, RADOLAN, etc.),
   - main research question in 2–3 sentences.
5. **Create a script** `scripts/02_download_data.R` that:
   - defines URLs for at least one DGM tile and one RADOLAN product,
   - uses `here::here()` to define local file paths in `data/raw`,
   - uses `download.file()` with an `if (!file.exists(...))` guard,
   - is fully commented (what is downloaded, from where, and why).
6. **Initialize a GitHub repository** (private or public) and push:
   - the project structure,
   - `renv.lock`,
   - the metadata file,
   - the setup and download scripts.

The next session (Module 2) will assume that **all groups can load at least their DGM file and one specific other data set from `data/raw`** using `here::here()`.

Module 2: Geodata Preprocessing

(... continues ...)