

4 – Execute the Burgwald Data Pipeline

Clone · Configure · Run · Verify

Chris Reudenbach

Goals

- Clone the course repository and open the RStudio project.
- Run all core pipeline scripts **unchanged** on your own machine.
- Verify that raw data, processed data, and predictor stacks are created.
- Document any problems that prevent full reproducibility.

Written output...

Please export all written answers as a PDF and save it in the project directory **docs/** using the filename convention **<Name1_NameN>_ws-04.pdf**. This helps you maintain a clean and consistent documentation of your work.

You are welcome to use ChatGPT for wording support, but the important part is that you genuinely understand the tasks and the underlying concepts — the AI is only meant to reduce typing effort, not replace your comprehension.

How to use ChatGPT for this assignment

You may use ChatGPT **only** to refine the wording of your own ideas.
Please follow this workflow:

1. **Write your own bullet points first** (even rough or incomplete).
2. Paste your bullets into ChatGPT and ask for **clarification, shortening, or better phrasing**.
3. ChatGPT will only help with **language and structure**, not with content.
4. If you are unsure about something, write: *“I’m not sure about X”* — ChatGPT may then give **hints or guiding questions**, but **no solutions**.
5. Answers must remain **ultra-short and clear** (1–3 bullets or 1–2 sentences).
6. Do **not** ask ChatGPT for full answers; the goal is that **you understand the task**, not that AI solves it.

This ensures that the work remains genuinely yours while reducing unnecessary typing.

Repository

Clone this repository locally and open the .Rproj:

<https://github.com/gisma/LV-19-d19-006-25-R.git>

Task 1 – Run the setup

Script: 01_setup-burgwald.R

- Open the script.
- Run it from top to bottom **without any modifications**.
- After successful execution, the following must exist:

```
data/raw/AOI_Burgwald/  
data/processed/aoi_burgwald.gpkg  
src/01-fun-data-retrieval.R (loaded without error)
```

Minimal check (optional in console):

```
fs::dir_ls("data", recurse = TRUE)
```

Task 2 – Run generic data retrieval

Script: 01-data-retrieval.R

- Run the script and change the time-period to something like 1 month.
- Afterwards, the following data must physically exist (content not relevant, existence is):
 - DEM (GeoTIFF)
 - OSM-by-key files
 - CLC raster / GeoPackage
 - DWD station data in data/raw/dwd-stations/
 - processed DWD files in data/processed/dwd/

Suggested quick check:

```
fs::dir_ls("data", recurse = TRUE)
```

Task 3 – Test external API (SAGA)

Script: 01-04-1-SAGA.R

- Adjust the SAGA binary path if needed (only this, nothing else).
- Run the script.
- At least **one new terrain raster** (e.g. slope, aspect) has to be created.
- Note the file name and location.

Task 4 – Sentinel-2 via CDSE

Script: 01-2-CDSE-sentinel-data-retrieval.R

- Run the script unchanged.
- Expected result (folder may vary slightly, adapt to repo):

```
data/raw/s2/  
  <Sentinel-2 scenes or COG references>
```

- Count/list how many items were downloaded or referenced.

Task 5 – Sentinel-2 via gdalcubes

Script: 01-3-gdalcubes-sentinel-data-retrieval.R

- Run the script unchanged.
- Expected output:

```
data/predictor/  
  <NetCDF file>    # e.g. s2_predictor_stack_summer.nc
```

- Confirm that the NetCDF file exists and can be opened, e.g.:

```
stars::read_stars("data/predictor/<your-file>.nc")
```

Task 6 – Pipeline check

Run this final check:

```
list.files("data/raw", recurse = TRUE)
list.files("data/processed", recurse = TRUE)
list.files("data/predictor", recurse = TRUE)
```

Verify that all three stages exist on your system:

- RAW DATA
- PROCESSED DATA
- PREDICTOR STACK(S)

If any stage is missing, re-run the corresponding script or document the error.

Task 8 – Short reproducibility log

Write **one short paragraph** (max. 5–6 sentences) for your own log:

- Did the full pipeline run on your machine?
- Which scripts worked without issues?
- Where did you have to change paths or credentials (e.g. SAGA, CDSE)?
- Which errors (if any) stopped the pipeline?

You will use this later to compare reproducibility between systems.

Expected final structure

00-setup-burgwald.R • pacman::p_load() (zentral) • Projektstruktur / Pfade / AOI • Tempdirs (R/GDAL/terra) • source("src/01-fun-data-retrieval.R")

00-setup-burgwald.R...

src/01-fun-data-retrieval.R (Helper) • aoi_with_buffer() • get_osm_burgwald_by_key() • run_if_missing(), download_if_missing(), write_nc_if_missing() • burgwald_get_hourly_dwd(), burgwald_get_subhourly_precip(), burgwald_get_radolan_rw_recent() • mapview_clc()

src/01-fun-data-retrieval.R (Helper)...

01-1_get_base-geodata.R Base Geodata • DGM1 mosaic+clip • CLC5 clip • OSM by key Output: data/raw/AOI_Burgwald/{dem,clc,osm_by_key}/...

01-1_get_base-geodata.R...

02-2_processing_s2-predictor-stacks-summer.R Sentinel-2 Jahresstacks (CDSE) • STAC search + daily cloud stats • Best summer day je Jahr • RawBands + kNDVI/SAVI/EVI via CDSE::GetImageOutput: data/pred_stack_.tif

02-2_processing_s2-predictor-stacks-summer.R...

01-3_get-gdalcubes-sentinel-data.R Sentinel-2 Databcube (STAC → gdalcubes) • Monatsmedian 2018-06...2022-09 • Wolkenmaskierung via SCLOutput: data/burgwald_2018_2022_all.nc

01-3_get-gdalcubes-sentinel-data.R...

01-4_get_DWD-data.R DWD Forcing (Orchestrator) • Zeitraum setzen • Calls: burgwald_get_hourly_dwd() + burgwald_get_subhourly_precip() Output: data/raw/dwd-stations/ + data/processed/dwd-stations/

01-4_get_DWD-data.R...

source()

source()

uses helpers

uses helpers

uses helpers

uses helpers

uses write_nc_if_missing()

uses write_nc_if_missing()

optional context (AOI/CLC)

optional context (AOI/CLC)

optional context (AOI/CLC)

optional context (AOI/CLC)

Text is not SVG - cannot display

The expected directory tree at the end of this worksheet should be more or less like this:

```
data/
  processed
    dwd-stations
    osm_by_key
    radolan-rw
    relief_1m
  raw
    AOI_Burgwald
    clc
    dem
    osm_by_key
    clc5_2018_copernicus
    Results
    dgm1_burgwald
    dgm1_coelbe
    dgm1_gemuenden
    dgm1_haina
    dgm1_lahntal
    dgm1_muenchhausen
    dgm1_rauschenberg
    dgm1_rosenthal
    dgm1_wetter
    dgm1_wohra
    dwd-stations
      unzipped_10min_precip
      unzipped_5min_precip
      unzipped_hourly_
        precipitation
        wind
      unzipped_hourly_precipitation
      unzipped_hourly_wind
    radolan-rw
    u2018_clc2018_v2020_20u1_raster100m
      DATA
        French_DOMs
        Documents
        French_DOMs
        Legend
        Legend
        Metadata predictor/
  predictos
src/
```

```
outputs/  
  figures/  
docs  
metadata  
renv
```

Take-home summary

This pipeline forms the foundation for all later modelling tasks in the course. By running it successfully, you ensure that:

- all Burgwald base layers (DEM, OSM, CLC, DWD) are available,
- all Sentinel-2 sources (CDSE + gdalcubes) are standardised and reproducible,
- the project contains a clean, FAIR-aligned data structure, and
- downstream scripts (classification, predictors, modelling) can run without modification.

In short: if this pipeline runs, every later analysis in the course becomes plug-and-play.