BioR - Class 2 Challenge (Field Campaign)

Tidyr • Relational data (joins) • Hybrid objects (list-columns)

Scenario. You collected data during a field campaign. Biomass was entered in a spreadsheet in wide format (one column per day), while temperature was recorded separately by loggers. Sample IDs contain metadata (site, treatment, replicate) but use inconsistent separators. Your task is to create an analysis-ready tidy dataset and a hybrid tibble with one linear model per site relating biomass to temperature.

Tasks

A. Inspect + types

Convert site_meta$start_date to Date. Use a quick inspection step (e.g., glimpse/skim) and comment on one issue you notice.

B. Tidy biomass + parse IDs

Convert the biomass spreadsheet into tidy format. First, standardize `sample_id` by replacing - with _. Then pivot the day columns (`day_0`, `day_7`, `day_14`) into two columns (`day`, `biomass`), convert `day` to numeric (0/7/14), and split `sample_id` into `site`, `treatment`, and `rep`.

C. Duplicate keys + QC flag

Check whether any sample-day combination appears more than once (same site, treatment, replicate, and day). If duplicates exist, collapse them into a single value (e.g., take the mean) and add a qc_flag column to record whether the row was 'OK' or created by merging duplicates.

D. Temperature aggregation + joins

Compute the mean temperature for each site and day (because there are multiple logger readings). Then join temperature onto the biomass table using (site, day). Compare inner_join vs left_join by reporting the number of rows produced and explain which rows are removed (inner) or kept with NA temperature (left).

E. Make missing combos explicit

Some sample-day measurements are missing and therefore do not appear as rows. Use `complete()` to add the missing rows so that each `(site, treatment, rep)` has day 0, 7, and 14. Keep missing biomass values as `NA` and create `missing_biomass_flag` to mark them.

F. Hybrid dataset: one lm per site

Group the dataset by site, then nest so each site becomes one row with a `data` list-column. Fit one linear model per site (`biomass ~ temp_c`) and store each model in a `fit` list-column. Finally, extract slope, intercept, R^2, and the number of observations used (n) into standard numeric columns for reporting.

Export outputs

Write outputs/biomass_temp_tidy.csv and outputs/site_models_summary.csv.

Hints (do not copy-paste as a full solution)

- Standardize separators first: sample_id |> str_replace_all('-', '_')

- Use parse_number() to extract day from 'day_14'

- If pivot_wider complains about duplicates, your key is not unique

- For modeling per site: group_by(site) |> nest() |> mutate(fit = map(data, ~ lm(...)))