ECS4: Data project

2026-03-14

Table of contents

All remaining computer sessions, as well as the final seminar, will be devoted to a mini-project.

The formal examination consists of:

- A presentation of your work during the final seminar
- A review of your Git repository

> **Note**
>
> The GitHub repository itself constitutes the submission. No additional files need to be uploaded to Canvas.
>
> When your work is ready for review, first push all commits to GitHub and then submit the Canvas assignment.

The scheduled computer sessions are primarily intended to give groups an opportunity to work together in person and to allow the teacher to be available for questions and guidance.
You should expect that additional work on the project will be required outside these sessions.

The project

Your task is to conduct a small **health data project** using the datasets from the previous computer exercises.

You should formulate a research question that can be explored using the available datasets (for example through data linkage). The question does **not** need to be novel, but it should be clear and interesting enough to investigate.

During the project it is perfectly acceptable to:

- refine your question
- narrow or broaden the scope
- adapt the analysis based on intermediate findings

Exploratory analysis is also acceptable.

Your project should involve **more than one dataset** and may include additional contextual data (for example aggregated state-level information, translations of medical codes, or other external sources).

Typical steps in the project include:

- understanding the available data
- selecting relevant datasets
- cleaning and validating the data
- standardising variables and labels where necessary
- deriving new variables
- linking datasets
- analysing the research question
- presenting results using figures and tables

More technical information about the data can be found in
this GitHub wiki.
The variable lists and data scans introduced in ECS2 should already provide most of the information you need.

> **i** Statistical analysis
>
> The main focus of this project is not advanced statistical modelling, but rather the earlier stages of the data analysis process: understanding the data, preparing and cleaning it, and organising it in a reproducible workflow.
>
> You may use hypothesis tests, regression models, machine learning methods, or other statistical techniques if they are relevant to your question. However, this is **not required**. Well-structured descriptive analyses (tables, figures, and clear interpretation) will be sufficient.

Group work

For this assignment you will collaborate through GitHub in **groups of two students**.

When accepting the assignment in GitHub Classroom (using the link found in Canvas), you will either **create a team** or **join an existing team**.

Please follow **one of the two options below**.

If you already have a partner

1. Decide who will create the team.
2. That person accepts the assignment and chooses **Create a new team**.
3. Choose any team name.
4. The other student accepts the assignment and chooses **Join team**, selecting the same team.
5. Both of you clone the repository locally.

Important: **Only one person should create the team.**

If you do not yet have a partner

1. Open the assignment link in Canvas.
2. Check if there is a team whose name starts with `join-`.
3. If such a team exists, **join that team**.
4. If no such team exists, **create a team whose name starts with `join-`** (for example `join-1`, `join-2` etc).
5. Clone the repository locally and wait for another student to join.

If no one joins your team (for example because the number of students is odd), **please inform the teacher**.

Suggested workflow

Before writing any code, discuss with your partner how you want to organise the project and what direction it should take.

To help the project get started efficiently, each group may initially divide responsibilities between two roles.

Workflow Lead

Responsible for the technical setup:

- creating the repository structure
- setting up the computational workflow (for example a `{targets}` pipeline)
- implementing the first steps such as loading the data

Project Lead

Responsible for planning and coordination:

- clarifying the research question
- outlining the analysis strategy
- documenting the roadmap in `README.md`
- creating GitHub issues to organise tasks

These roles are mainly intended to help the project get started and may change later during the project.

Initial setup

After agreeing on both the project direction and the technical structure:

1. The Workflow Lead creates the basic repository structure.
2. Add essential files such as `README.md` and `.gitignore`.
3. Commit the structure and push it to GitHub.
4. The other student pulls the repository so both work from the same structure.

During this early stage it is often helpful if **one person manages structural changes** to reduce merge conflicts.

Collaborative development

Once the structure is established, **both students should contribute actively to the analysis and code development**.

To keep the collaboration organised:

- discuss tasks using **GitHub issues**
- assign issues where appropriate
- commit small changes frequently
- push updates regularly
- pull the latest version before starting work

Draft scripts

To experiment without disrupting the main workflow, you may create draft scripts such as `draft-your-name.R` within the repository (for both of you to see).

These can be used to test ideas before integrating code into the main workflow.

If you are using `{targets}`, place such files **outside the `R/` folder** so that they are not executed automatically.

It might be easier if both students develop draft code in parallel and that you use pair programming (see below) to later copy and past different parts of your draft code into your formal pipeline. You may need to experiment and see what works best for you!

Working with workflows

Using `{targets}` to organise the workflow is recommended but not required.

If you want to ensure identical package versions, you may also use `renv`.

Otherwise, verify that you are using the same R version:

```
R.version.string # run it in your R console
```

and update packages if needed (in a fresh R sessin without any packages loaded):

```
update.packages(ask = FALSE)
```

Merge conflicts

Since two people may modify the same files simultaneously, try to work from the **latest version of the repository** (sync often).

If conflicting changes occur, you will need to resolve merge conflicts.

Instructions

Pair programming

It can sometimes be useful to work together at the same computer. In such cases only one group member may commit the changes, even though both contributed.
Indicate in the commit message that the commit was made jointly.

Repository requirements

The cloned assignment repository is initially empty. When the project is complete it should contain at least:

- A project description (`README.md`)
- Analysis code (typically in an `R/` folder)
- A reproducible workflow (e.g. `{targets}` with `_targets.R`)
- Quarto documents presenting the results (introduced in EL9 so don't think about this yet)

Individual grading

! Important

Although this is a group project, the work will be graded **individually** (pass or fail).

Individual assessment will be based on:

- your contributions to the repository
- your ability to explain the project during the final seminar

During the seminar one student from each group will present during the first hour and the other during the second hour. You should therefore both be prepared to present the project independently.

Example projects

The examples below illustrate possible research questions and analyses based on the data.
They are intended **only as inspiration**.

You are free to choose your own research question and approach.

- Cardiovascular risk analysis
- Healthcare cost analysis

- Healthcare trends analysis
- Social determinants