EL5: Analysis pipelines

{targets} and reproducible analysis

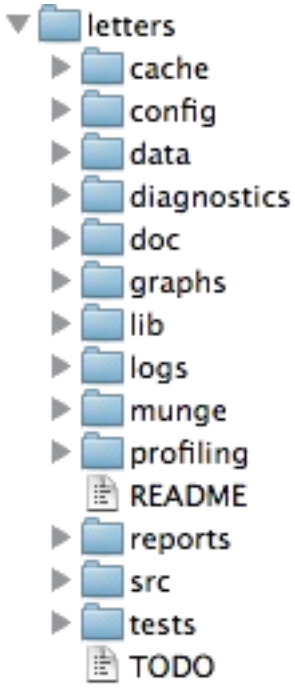
Not a single R script

- Real projects are more complex than a single R script!
- Multiple scripts
- Data files
- Documentation
- Reports
- Version control
- Reproducible workflows

Project structure

Common file structures

- Help you organize your thoughts
- Help others to collaborate
- simplifies paths used in your code



Example structure

```
/.../my_project/
  ├── README.md        - project documentation
  ├── TODO             - what should be done next?
  ├── .git             - handled by git (hidden folder)
  ├── .gitignore       - used by git but your responsibility!
  ├── data/            - your data files (not under version control!)
      ├── cancer.csv
      └── patients.qs
  ├── R/               - your saved R functions
      ├── function1.R
      └── function2.R
  ├── ...
  ├── ...
  └── _targets.R       - targets pipeline script (WHAT??? :-))
```

`README.md`

- Document the purpose of the project
- What is it about?
- What is the aim?
- Who to contact for questions?
- In what circumstances was it created?

Markdown format (simple text with some possible formatting)

Markdown

TODO: intro and give examples + link etc

`data` folder

- Store your data files here as they are when you get them
- Avoid any modifications to the raw files!
- It is very easy to forget what you do if it can not be traced by code
- Do NOT include this folder in version control!
- Git is not good at handling large files
- Sensitive data should not be shared!
- Add `data/*` to your `.gitignore` file
- In realistic projects, data might come in varying formats
  - csv, txt, xlsx, sas7bdat, sav, dta, etc (we will cover some of these later)
  - some files might be very big (gigabytes not uncommon)

`.gitignore` fiile

TODO

`R` folder

- Store your R functions here
- You have learned about R functions in the earlier R course
- We will cover some more of that later
- Document their purpose inline!
- Helps you to reuse code
- Easier to read main scripts if functions are defined elsewhere
- Easier to test and debug code
- we eill not cover testing but some about debugging later
- Easier to share code between projects

`reports`

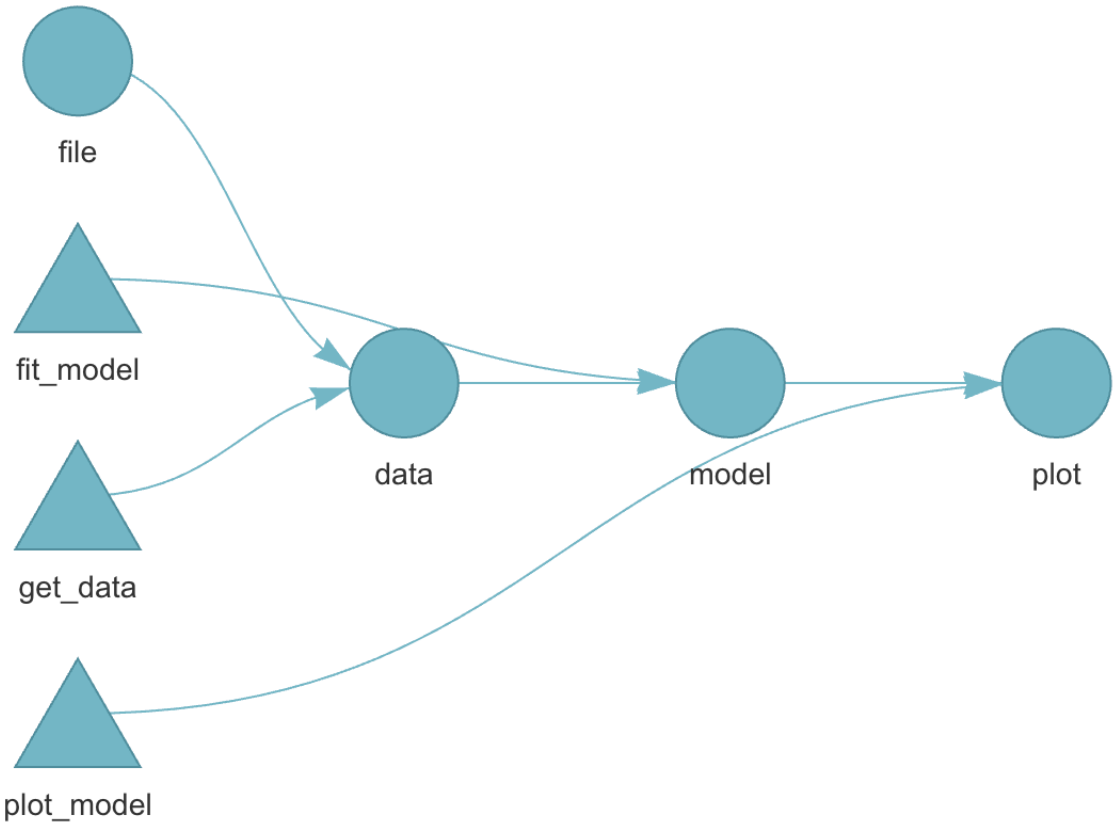
- Separate reports from analysis/data management but integrate them with the analysis pipeline
- Quarto documents
- Documemnt your analysis and thoughts/decisions along the way
- Include figures and tables
- To share with external collaborators

`_targets.R` file

Use the `targets` package to create a reproducible data analysis pipeline

Pipeline

- Define steps in your analysis as targets
- Define dependencies between targets
- Automatically track changes and rerun only necessary parts



Why?

- You do not want to rerun everything all the time!
- You want to keep track of what you have done
- You want to be able to reproduce your results later
- You want to share your workflow with others

Reading

- Targets overview
- Target manual