Grading and expectations

- 50% of the grade is based on **weekly exercices**
  - Exercices should be **submitted via github**, by thursday noon the following week
  - The best half of the exercises will make up the grade

- 50% of the grade is based on the **project** (groups of max 3 persons)
  - The project can be either:
    - Re-producing the analyses from a publication (in a critical fashion)
    - Analyzing new data (e.g. yours or in collaboration with a group)
  - The project *must be discussed and approved in advance*
  - The expected outputs of the project are:
    - a report (e.g. ~10-15 pages) with embedded full code and figures, and including an introduction and discussion of the results
  - Deadline: before the end of the day on **July 8th**

Evaluation scheme for the course project

- Format and formal requirements (1/6)
  - Rendered markdown (html or pdf), proper scientific references, figure legends, etc.
- Introduction/conclusion (1/6)
  - More is not necessarily better: ask yourself what background would your fellow students need to understand your problem, analyses and observations
- Analysis
  - **Correctness**, i.e. lack of mistakes (1/6)
  - **Adaptability/creativity**, i.e. whether you could adapt (e.g. the tools seen in class) to your purposes (1/6)
  - **Appropriateness**, i.e. whether you used the right tool/visualization to address a question (1/6)
  - **Interpretation/discussion**, i.e. whether you correctly describe and interpret your figures and analysis results (1/6)
- Extra 1/6 (summing to 7 of max 6 points) for difficulty / going beyond the expectations (to not penalize those who undertook something harder)

A few (optional) tips for preparing your project report

- It's often easier to split a project into different markdown documents for different steps
  - e.g. one that downloads the data, another processes it, another that answers specific questions of makes the figures with which you'll tell your story
  - you may want to number the documents so that they appear in the right order
  - if you want to go deeper, workflowr is a great rmarkdown organization and versioning framework

- If you're wondering how to do something in rmarkdown, consult this online book
  - See especially:
  - code chunk options
  - the section about references and bibliography
    (most reference management software, e.g. Zotero and the likes, can export references in the *bibtex* format required by rmarkdown)