

# PROJECT 2: MODEL ANALYSIS PROJECT

Christian Langholz Carstensen

**Vision:** Programming is more than writing code. The ultimate goal of the projects in this course is that you learn to formulate a programming problem of your own choice, and find your own way to solve it, and present the results. The bullets below are minimum requirements, but otherwise it is very much up to you, what you will like to do with your project. I hope to see some creative ideas!

- **Objectives:** In your model analysis project, you should show that you can:
  1. Apply model analysis methods (e.g. `scipy.optimize`, `scipy.linalg`, `sympy`).
  2. Structure a code project (see lecture 05).
  3. Document code.
  4. Present results in text form and in figures (put text in Markdown cells).
- **Content:** In your model analysis project, you should at a minimum:
  1. Describe an algorithm on how to solve a simple economic model (most likely taken from a textbook). See examples of algorithm descriptions in L09 L10.
  2. Solve (and perhaps simulate) a simple economic model.
  3. Visualize how the solution changes across parametrizations. This is an important task (that also works as a debugging device).
  4. Analyze one or more extensions of the baseline model.
- **Limitation:** *sympy* can only play a small role in your project
- **Structure:** Your data analysis project should consist of:
  1. A README.md with a short introduction to your project
  2. A single self-contained notebook (.ipynb) presenting the analysis
  3. (Optionally) Fully documented Python files (.py)

**Example of structure:** [See this repository](#).

- **Size:** *Quality before quantity*. It is better to have a rather simple but well-tested model with nice vizualization and simulation compared to a larger model that does not succeed entirely.
- **Hand-in:** On GitHub by uploading it to the folder:  
`github.com/projects-2022-YOURGROUPNAME/modelproject/`
- **Deadline:** 13th of May 23.59
- **Exam:** Your model analysis project will be a part of your exam portfolio.  
You can incorporate feedback before handing in the final version.
- **General advice:** In the beginning of the project, write your code in the way *you* find most intuitive. Also if that is a slow implementation that does not use optimized routines and such. You can then begin to optimize it when things seem to work the way you intend.