

PROJECT 2: MODEL ANALYSIS PROJECT

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. You can use pretty much any economic model that you have encountered during your studies. It is not a bad idea to do further work on a model you already have a good understanding of; for instance what you have been working at in a seminar or in your bachelor thesis. You can also pick one of the models listed in the last section.

Note: the Solow model was demonstrated in lectures as well as in exercise class. The requirements for doing a passable project with that model are therefore higher than if you start from scratch with another model.

- **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 your economic model (most likely taken from a textbook). See examples of algorithm descriptions in L09 L10.
 2. Solve the model using an optimization routine or an equation solver and simulate it.
 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. 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-YEAR-YOURGROUPNAME/modelproject/`

- **Deadline:** See Absalon.
- **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.

Examples of models you could implement in the project

You have most likely encountered some of these models during your core BA courses or in the elective courses. You can look them up in standard textbooks or google them. Also, these are just ideas, you are absolutely free to come up with something else.

- The CAPM model.
- The monocentric city model for city structure.
- The market for lemons.
- The Lucas islands model.
- The consumption-savings model from L11 (but with a new feature).
- Stackelberg competition.
- The Heckscher-Ohlin model.
- Optimal extraction of non-renewable resources.
- VaR (value at risk) as a tool for regulating capital requirements of banks.