Programming for Economist

PS 6

Class 5 & 9

Matias B.F. Hall,
Institute of Economics,
November 2025

KØBENHAVNS UNIVERSITET

Plan for today

1. Data project
2. Classes crash course
3. Hints for PS6

1. My groups

- 345
- 404-not-found
- AKathrine-Elisa-Caroline
- cool-grapes
- data-crushers
- elisabeth
- emma-og-luna
- fred-og-alex
- gala
- gfs189rtil595
- jmva
- kasper
- lena
- mamahael
- miami-vicer
- mountain-dew-code-review
- ren-hygge
- slangetaemmer
- smoothiesour
- snakes-experts
- sof
- sofie-ck
- teamcoding
- ultrasonic
- wee

1. Feedback

- Clean up your notebooks – example
- Have a README
  - Make it correct!
- Implement functions and python script file
- Create headlines/toc
- Use of AI
  - Try having an idea before asking.
  - A lot better code than just pasting a whole assignment question into gpt.

2. Classes

- Classes are blueprints for creating **objects**
- They contain data accessed as **attributes** and class specific functions called **methods**.
- You can create multiple **instances** of a class with their own set of attributes and methods.

- Today you will see, how we through one class, can solve different economic models through the use of multiple instances.
- For now we will only work with classes and objects, but not create our own classes.

2. Classes

- Define a class by `class ClassName`:
  - Initialize the class by `def __init__(self, attributes*)`
    - 'self' represents the instance of the class
    - Set attributes by `self.AttributeName = attribute`
  - Define one or more methods by
    - `def MethodName(self,args*):`
- Create an instance of the class
  - `ObjectName = ClassName(attributes)`
- Access attributes and methods of the specific class instance
  - `ObjectName.AtttributeName`
  - `ObjectName.MethodName(args*)`

3. Hints for PS6

- General
  - When choosing methods for optimisers, try different one. Some will fail, other differ in speed/precision, but most are fine for this course. Look up scipy documentation or ask gpt☺
  - All fill in questions draw on the math/info given – use it!

- Q2
  - 2.1 It is the utility and indifference methods not the indifference_curves methods you need to change
  - 2.2c try changing the max_iter or tol attributes
  - 2.3
    - Modify the 'solve_dictator_B' method in 'ExchangeEconomyModel.py'
    - Solve using xB as input or keep xA as input and get xB from 1-xA
- Q3
  - In the consumption method 0.1 is added, this messes up the solution so remove it.
  - Q3.1 Use a bounded method to ensure l>0
  - Q3.2 Use any 2 of the 3 clearing conditions at zero as errors
  - Q3.3 Use a root optimiser to solve for the errors