Project 0: Inaugural Project

Christian Langholz Carstensen

Vision: The inaugural project teaches you to solve a simple economic model and present the results.

- **Objectives:** In your inaugural project, you should show that you can:
  1. Apply simple numerical solution and simulation methods
  2. Structure a code project
  3. Document code
  4. Present results in text form and in figures
- **Content:** In your inaugural project, you should:
  1. Solve and simulate a pre-specified economic model (see next page)
  2. Visualize results

  Example of structure: See this repository.
- **Structure:** Your inaugural 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)
- **Hand-in:** On GitHub by uploading it to the subfolder *inaugralproject*, which is located in:

  github.com/NumEconCopenhagen/projects-2022-YOURGROUPNAME

  1. Create your GitHub repository if you have not already done so. Go to Github and follow step 1 in this guide.
  2. **Test** if your notebook **runs:** Restart the notebook kernel and run all cells.
  3. Stage, commit and push all your relevant files to the repository on Github.
- **Deadline:** 27th of March 23.59
  - **Peer feedback:** After handing in, you will be asked to give peer feedback on the projects of two other groups.
    The solution will be made available after the peer feedback round has ended.
- **Exam:** Your inaugural project will be a part of your exam portfolio.
  You can incorporate feedback before handing in the final version.

Getting an insurance

We will here work with a benchmark model of insurance policies, Mossin's model. We are dealing with an agent wanting to get an insurance. The agent holds assets y and may suffer a monetary loss x. For now we assume that the loss is incurred with probability p. Utility of assets is given by $u(\cdot)$.
If the agent is not insured, expected value is

$$V_0 = pu(y - x) + (1 - p)u(y)$$

The agent may however obtain an insurance contract to mitigate a loss. The contract consists of a coverage amount q and a premium π. The coverage cannot exceed the loss so $q \in [0, x]$ and π must be paid irrespective of whether a loss was incurred. To not go broke, the insurance company must require higher premium for better coverage. It therefore implements a premium policy

$$\pi(p, q) = pq,$$

An insured agent will thus have expected utility

$$V(q; \pi) = pu(y - x + q - \pi(p, q)) + (1 - p)u(y - \pi(p, q))$$

To solve the model numerically, we let u be given by the function

$$u(z) = \frac{z^{1+\vartheta}}{1 + \vartheta}$$

And use the following parameterization.

$$y = 1, \ p = 0.2, \ \vartheta = -2$$

Questions (sub-questions in roman numbers):

1. *i*) Construct a function that takes (x, y, p) as arguments and returns the agents' optimal insurance coverage:
$$q^\star = \text{argmax}_{q \in [0,x]} V(q; \pi)$$

   ii) Make a grid of x in the range $[0.01, 0.9]$.

   iii) For each x, calculate $q^\star$.

   iv) Plot the xs and $q^\star$s and write what the graph suggests about the general solution to the agent's problem? (You can verify the finding analytically if you will).

2. We would like to find the set of *acceptable* contracts from the agent's point of view. This would be all the contracts (q, π) which ensure an expected value at least as good as if not having an insurance, $V(q; \pi) = V_0$.

   i) Let the loss be x= 0.6 and construct a grid of q over $[0.01, 0.6]$.

ii) Think now of π as a *variabel* that has to be solved for (instead of it being a function of p and q as before). Loop over each element in the grid of qs and find the corresponding $\tilde{\pi}$ such that $V(q;\tilde{\pi}) = V_0$.

iii) Make a plot of the acceptable premiums $\tilde{\pi}$ (the q grid on the x-axis). Add to the diagram the function $\pi(p,q) = pq$ from before, which gives the premiums that the insurance company need at least in order to break even at each q. Taken together, these two graphs map out the set of feasible premiums for a given q.
Try to make the graph pretty and informative (as if it was going into a text book or paper). That is, use shading, labels, titles etc.

3. We will now consider a modification to the setup where the loss x is drawn from a beta distribution and the coverage q is a *fraction* of x. Specifically, we'll set

$$q = \gamma x, \quad \gamma \in [0,1]$$

$$x \sim \text{Beta}(\alpha, \beta), \quad \alpha = 2, \ \beta = 7$$

which means that the agent's value is written as

$$V(\gamma, \pi) = \int_0^1 u(y - (1-\gamma)x - \pi) f(x) dx$$

where $f(\cdot)$ is the density of the Beta distribution. Note that the Beta distribution has support in $[0,1]$ which explains the limits in the integral.

i) Create a function that computes $V(\gamma,\pi)$ by Monte Carlo integration using at least 10,000 draws.

ii) Consider the two following insurance policies:

$(\gamma,\pi)^1 = (0.9, 0.2)$
$(\gamma,\pi)^2 = (0.45, 0.1)$

Which one is preferable to the agent?

4. We now consider the situation from the insurance company's viewpoint. They know the parameters of the loss distribution. Their policy is to let customers set a coverage ratio γ and then return an offer on the premium π to maximize profits.

i) Given a customer wanting $\gamma = 0.95$, what is the profit maximizing premium, $\pi^\star$?

Hint 1: the insurance company acts as a monopolist, so the offer is only just acceptable to the client.

Hint 2: you will probably want to use the 'broyden1' method when calling scipy's root finder.