Econ 204B

Matthew Fitzgerald

UCSB

January 10, 2020

Getting to know you

On notecard please write:

- Name: Matthew Fitzgerad
- Preferred First Name: Matt
- Hometown: Riverside, CA
- Programming language with most experience: R
    - Level: beginner/intermediate/advanced

General Information

Contact:

- Email: mfitzgerald00@ucsb.edu
- Office: North Hall 2048
- Office Hours: Wednesdays 11:00am -1:00pm

Problem Sets:

- Typically will have both written and programming components
- Grades will be on a 10 point scale
- Need to be done in LaTeX

Materials

- There are lots of online materials and books
- One very useful resource are the lectures from QuantEcon
  - Material on topics
  - Material on Julia
- For Julia specifically:
  - Julia documentation page
  - Learn Julia the Hard Way

General Cont.

- I will be using Julia with Atom as my text editor
- You can use any language that you'd like, however there are benefits to using Julia
  - You have resources to help you learn (Professor Rupert and me)
  - You get experience early in an up and coming language
  - Even if you know other languages, it's always good to be familiar with as many as possible
- For the first assignment you will need to use Julia, and later I will create Julia scripts for you to complete
- If you had trouble installing, let me know and we can get you set up

Overview

- This quarter we will learn a new tool: **dynamic programming**
- Allows us to take into consideration the dynamics of economic decision making
  - In simple static optimization problems individuals consider how much of each good they'd like to consume
  - In dynamic optimization we take into account that consumption today comes at the expense of savings, and thus consumption tomorrow
  - Can also take into account uncertainty in the future which influences decisions today

Examples of Dynamic Problems

Dynamic optimization important for many problems in both micro and macro economics:

- General consumption problems
  - How much to consume vs. save tomorrow (can add shocks to future income stream)
  - Decision to invest in human capital (take lower wage in the present to attain a higher wage in the future)
- Resource Extraction problems (water, fish, timber harvest, etc.)
- Firm entry (or exit) decision
- Job serach

Dynamic Programming Intuition

Recall the definition of indirect utility

$$V(\boldsymbol{p}, M) = \max_{\boldsymbol{x}} u(\boldsymbol{x})$$
$$\text{s.t. } \boldsymbol{p'x} = M$$

First order conditions:

$$\frac{u_i(\boldsymbol{x})}{p_i} = \lambda \text{ for i = 1, 2, ..., I}$$

Dynamic Programming Intuition Cont.

What is the interpretation of $V(\boldsymbol{p}, M)$?

- Maximized level of utility given current state $(\boldsymbol{p}, M)$
- Given $(\boldsymbol{p}, M)$ consumer will have $V(\boldsymbol{p}, M)$ level of utility

What is the interpretation of λ?

$$\begin{aligned} \lambda &= V_M(\boldsymbol{p}, M) \\ \implies \frac{u_i(\boldsymbol{x})}{p_i} &= V_M(\boldsymbol{p}, M) \text{ for i} = 1, 2, ..., \text{I} \end{aligned}$$

If we knew $V(\boldsymbol{p}, M)$ at every $(\boldsymbol{p}, M)$, don't need to know how consumer will allocate M just need to know she's acting optimally

Multi-Period Problems

- While this was a static optimization problem, when we look at dynamic optimization problems the intuition will be similar
- Consider the optimization problem faced by an infinitely-lived agent with an endowment stream $\{e_t\}_{t=0}^{\infty}$ (assume complete markets). Agent maximizes present discounted value of lifetime utility:

$$\max_{\{c_t\}_{t=0}^{\infty}} \sum_{t=0}^{\infty} \beta^t u(c_t)$$

$$\text{s.t. } \sum_{t=0}^{\infty} c_t \leq \sum_{t=0}^{\infty} e_t$$

Multi-Period Problems Cont.

- Solution to this problem is sequence of consumption decisions $\{c_t\}_{t=0}^{\infty}$
- If we add uncertainty we would need an optimal sequence for every possible sequence of states
- Instead we can write the problem recursively, making the assumption that the agent will make optimal decisions in the future:

$$V(a) = \max_{c}\{u(c) + \beta V(a')\}$$

s.t. c + a' = (1+r)a + e

- Solution will be a value function V, or equivalently a policy function that maps states to outcomes

Where we're going

- Next week we'll talk more about solving these dynamic problems
- We'll also talk more about recursive formulation of the problem
- These types of problems are often difficult or impossible to solve analytically, and thus we need to learn computational methods to find solution
- The recursive formulation of these types of problems makes them easy to set up on the computer

Latex in Atom

- Install the following packages:
  - language-latex
  - latex
  - latex-completions
  - latex-plus
- Potential issue with keyboard shortcut to compile
  - One solution is to go into settings in latex package and check the box for "Build on Save"
  - This will compile pdf every time you save, thus cmd+s/cntrl+s will compile and save

Homework 1

- Homework 1 will be posted after section today
  - Four questions: Variable assignment, Functions, Loops, and Plotting
  - Once you know how to do these 4 things in a language, you're 'conversational' (i.e. you now know enough to Google questions)
- First try to do the problems on your own, but certainly feel free to ask me questions
- Learning a new language can be daunting but once you get the basics down it becomes much easier!

Let's take a look at Atom!