

## Problem Set #9

ECON 815, Prof. Jason DeBacker  
Due Friday, December 13, 5:00 p.m.

This problem set will have you working *collaboratively* to extend the basic overlapping generations models we worked with in class. I want you to make the following extensions to the model:

1. Add endogenous labor supply
2. Allow for  $S$ -period lived agents (where  $S$  and integer that represents is the maximum number of model periods a household lives)
3. Add realistic demographics to the model, which includes mortality risk and demographic transitions based on mortality, fertility, and immigration rates

You all will work together on these modifications. You will collaborate on GitHub. Questions can be posted as Issues on the class repository. Code additions will made through pull requests to the PS9 branch: [https://github.com/jdebacker/CompEcon\\_Fall19/tree/PS9](https://github.com/jdebacker/CompEcon_Fall19/tree/PS9).

### DELIVERABLES

You will have the following deliverables, all of which will be placed in the `./CompEcon_Fall19/OverlappingGenerations/ProblemSet9` directory:

1. Working code. The OG model with the above additions should solve. This code should be contained in the following `*.py` files:
  - `execute.py` – sets parameters, solves the model
  - `households.py` – contains functions related to the households' problems
  - `firm.py` – contains functions related to the firm's problem
  - `aggregates.py` – contains functions related to the aggregate variables used in the market clearing conditions
  - `SS.py` – contains functions to solve the steady-state of the model
  - `TPI.py` – contains functions to solve for the transition path of the model
2. A a tex file and a pdf compiled from TeX that includes (a) a plot of interest rates and wage rates over the transition path, (b) plots of the aggregate capital stock and labor supply over the transition path, (c) a plot of the lifecycle profiles of savings consumption, and labor supply from the steady-state. Please give a brief description of each of these figures. Please name the pdf "ProblemSet9.pdf".

Your grade will reflect a common component based on the above deliverables as well as an individual component based on what I observe regarding your contributions on GitHub (which includes additions of source code, discussion of issues, and comments to PRs).

### HELPFUL TIPS

1. Checking out and working with a remote branch from a GitHub repo:
  - Do: `git checkout -b LocalName upstream/upstreambranchname` will create a new branch on your local machine named `LocalName` that is based on the upstream repo's `upstreambranchname` branch.
  - E.g., to checkout the PS9 branch and give it the name PS9 locally do: `git checkout -b PS9 upstream/PS9`
  - Do your work on a branch off of the PS9 branch. Do this by checking out a new branch *when you are on the PS9 branch*:

- e.g., `git checkout -b ps9_hh` will create a new branch called `ps9_hh`
  - To keep your PS9 branch up-to-date, you'll fetch from the upstream branch and merge in any changes *while on your PS9 or PS9 working branch*:
    - `git fetch upstream`
    - `git merge upstream/PS9`
2. Start early!
  3. Agree on variable and parameter naming conventions and stay consistent
  4. Be consistent with what is in each dimension of NumPy arrays and the order items are passed through tuples
  5. Document your code so others can understand it
  6. Consider testing your code to make sure each piece works as expected before trying to put them together
    - I've put an example test in the `CompEcon_Fall19/OverlappingGenerations/tests`
    - You can run these tests by typing `python -m pytest tests/test_name.py` (or `python -m pytest` to run all tests) from the `./CompEcon_Fall19/OverlappingGenerations/ProblemSet9` directory in your terminal window (Mac/Unix/Linux) or your Anaconda Command Prompt (Windows)
  7. Google is your friend – as am I and your classmates – search the internet for solutions and post any unanswered questions as GitHub issues.