Econ 712: Problem set 3

Q1

(a)

I use Arrow-Debreu market structure with single market at date zero with claims to consumption in any future date and any state to solve Q1. A CE is an allocation $\{c_t^1, c_t^2\}_{t=1}^{\infty}$ and a set of prices $\{q_t(s^t)\}_{t=1}^{\infty}$ such that:

1. Agent $i \in \{1,2\}$ maximizes his utility by solving the problem:

$$\max_{c_t^i(s^t)} \sum_{t=0}^{\infty} \sum_{s^t} \beta^t \pi_t(s^t) \log(c_t^i(s^t))$$

$$s.t. \qquad \sum_{t=0}^{\infty} \sum_{s^t} q_t(s^t) c_t^i(s^t) = \sum_{t=0}^{\infty} \sum_{s^t} q_t(s^t) e_t^i(s^t) \tag{1}$$

2. Markets clear:

$$s.t. \qquad \sum_i c_t^i(s^t) = 1, \qquad \forall t, s^t \tag{2}$$

Where $\pi_t(s^t) = \delta$ for any history of endowments $\{1, N_1, 0_t\}$ or $\{0, N_0, 1_t\}$, $\pi_t(s^t) = 1 - \delta$ for any history of endowments $\{1, N_1, 1_t\}$ or $\{0, N_0, 0_t\}$ such that the number of zeros and ones in histories of endowments is nonnegative: $N_0, N_1 \geq 0$; $\pi_t(s^t) = 1$ for $\{1, N_1, N_0\}$ such that $N_1 \geq 0$ and $N_0 \geq 2$ and $\{0, N_0, N_1\}$ such that $N_1 \geq 2$ and $N_0 \geq 0$; $\pi_t(s^t) = 0$ for any other history of endowments.

(b)

From f.o.c. we have that

$$q_t(s^t) = \beta^t \pi_t(s^t) \frac{u'(c_t^i(s^t))}{\mu_i}$$

Given that date $t = s$ is random, the permanent income stream of agent 1 can be presented as initial endowment plus discounted expected value of income tomorrow: $e^1 = \sum_{t=0}^{\infty} \sum_{s^t} \beta^t \pi_t(s^t) e_t^1(s^t) = 1 + \beta((1-\delta)e^1 + \delta \times 0)$. Similarly, for agent 2 we have $e^2 = \sum_{t=0}^{\infty} \sum_{s^t} \beta^t \pi_t(s^t) e_t^2(s^t) = 0 + \beta((1-\delta)e^2 + \delta \times \frac{1}{1-\beta})$. Solving for e^1 and e^2 gives $e^1 = \frac{1}{1-\beta(1-\delta)}$ and $e^2 = \frac{\beta\delta}{(1-\beta(1-\delta))(1-\beta)}$

No aggregate uncertainty implies that $c_t^i(s^t) = \bar{c}^i$. Substitute $q_t(s^t)$ into (1) and solve for $\bar{c}^i$,

$$\bar{c}^1 = (1-\beta) \sum_{t=0}^{\infty} \sum_{s^t} \beta^t \pi_t(s^t) e_t^1(s^t) = \frac{1-\beta}{1-\beta(1-\delta)}$$

$$\bar{c}^2 = (1-\beta) \sum_{t=0}^{\infty} \sum_{s^t} \beta^t \pi_t(s^t) e_t^2(s^t) = \frac{\beta\delta}{(1-\beta(1-\delta))}$$

The price at date zero: $q_t(s^t) = \beta^t \pi_t(s^t)$. The prices of claims to each consumer's endowment process are given by,

$$p_{e1} = \sum_{t=0}^{\infty} \sum_{s^t} q_t(s^t) e_t^1(s^t) = e^1 = \frac{1}{1 - \beta(1-\delta)}$$

$$p_{e2} = \sum_{t=0}^{\infty} \sum_{s^t} q_t(s^t) e_t^2(s^t) = e^2 = \frac{\beta\delta}{(1 - \beta(1-\delta))(1-\beta)}$$

(c)

Price of a claim to the aggregate endowment:

$$p_{ea} = \sum_{t=0}^{\infty} \sum_{s^t} q_t(s^t) \times 1 = \sum_{t=0}^{\infty} \sum_{s^t} \beta^t \pi_t(s^t) = \sum_{t=0}^{\infty} \beta^t \sum_{s^t} \pi_t(s^t) = \frac{1}{1-\beta}$$

The risk-free interest rate is the inverse of a one period bond price,

$$p_b = \beta \sum_{s_{t+1}} \pi_{t+1} = \beta$$

$$R = \frac{1}{\beta}$$

(d)

Lagrangian of Social Planner's

$$L = \sum_{t=0}^{\infty} \sum_{s^t} \lambda \beta^t \pi_t(s^t) \log(c_t^1(s^t)) + (1-\lambda) \beta^t \pi_t(s^t) \log(c_t^2(s^t)) + \theta_t(s^t)(1 - \sum_i c_t^i(s^t))$$

The f.o.c. for agent 1,

$$\beta^t \pi_t(s^t) \frac{1}{c_t^1(s^t)} = -\lambda^{-1} \theta_t(s^t)$$

The f.o.c. for agent 2,

$$\beta^t \pi_t(s^t) \frac{1}{c_t^2(s^t)} = -(1-\lambda)^{-1} \theta_t(s^t)$$

Combine f.o.c.-s for agent 1 and agent 2,

$$\frac{c_t^2(s^t)}{c_t^1(s^t)} = \frac{1-\lambda}{\lambda}$$

Substitute $c_t^1(s^t) = \frac{\lambda}{1-\lambda} c_t^2(s^t)$ into the resource constraint $1 = \sum_i c_t^i(s^t)$ and solve for $c_t^2(s^t)$. The solution to SP's problem is constant consumption allocation $\bar{c}^2 = 1 - \lambda$ and $\bar{c}^1 = \lambda$.

(e)

Yes, the competitive equilibrium is Pareto optimal. Set λ such that,

$$\lambda = \frac{1-\beta}{1-\beta(1-\delta)}$$

In order to to decentralize the Pareto optimum as a competitive equilibrium for a given λ, set lump-sum transfer τ such that,

$$\lambda = \frac{(1-\beta)(1+\tau)}{1-\beta(1-\delta)}$$

$$1-\lambda = \frac{\beta\delta - \tau(1-\beta)}{(1-\beta(1-\delta))(1-\beta)}$$

Q2

(a)

Rep agent's problem:

$$V(a,s) = \max_{a',c}\{u(c) + \beta E[V(a',s')|s]\}$$

s.t. $a' \in [0,1], c \geq 0, c + p(s)a' = (p(s)+s)a$ where $E[V(a',s')|s] = \int V(a',s')F(ds',s)$.

RCE: Value function $V(a,s)$ and pricing function $p(s)$ s.t. V solves the Bellman equation (household optimize) and $V(1,s)$ is optimized by $a'=1$,$c=s$ (market clears).

(b)

Solving:

$$u'(c) = \lambda$$
$$p(s)\lambda = \beta E[V_1(a',s')|s]$$

Envelope condition:

$$V_1(a,s) = (p(s)+s)\lambda$$

Combining to get the Euler eq, and setting $a'=1, c=s$:

$$u'(s)p(s) = \beta E[(p(s')+s')u'(s')|s]$$
$$p(s) = E[\beta(p(s')+s')\frac{u'(s')}{u'(s)}|s]$$

Re-introducing time subscripts:

$$\begin{aligned}
p_t &= E_t[(p_{t+1}+s_{t+1})\frac{u'(s_{t+1})}{u'(s_t)}] \\
&= E_t[s_{t+1}\beta\frac{u'(s_{t+1})}{u'(s_t)}] + E_t[p_{t+1}\beta\frac{u'(s_{t+1})}{u'(s_t)}] \\
&= E_t[s_{t+1}\beta\frac{u'(s_{t+1})}{u'(s_t)}] + E_t[E_{t+1}[(p_{t+2}+s_{t+2})\beta\frac{u'(s_{t+2})}{u'(s_{t+1})}]\beta\frac{u'(s_{t+1})}{u'(s_t)}]
\end{aligned}$$

Recursively subbing in p_{t+j} and using the law of iterated expectation will give us

$$p_t = E_t[\sum_{j=1}^{\infty} \beta^j s_{t+j} \frac{u'(s_{t+j})}{u'(s_t)}]$$

For log utility:

$$\frac{p_t}{s_t} = E_t[\sum_{j=1}^{\infty} \beta^j s_{t+j} \frac{s_t}{s_{t+j} s_t}] = \frac{\beta}{1-\beta}$$

which doesn't depend on the distribution of consumption growth.

(c)

For CRRA:

$$\begin{aligned} p_t &= E_t[\sum_{j=1}^{\infty} \beta^j s_{t+j} (\frac{s_t}{s_{t+j}})^{\gamma}] \\ &= E_t[\sum_{j=1}^{\infty} \beta^j s_{t+j}^{(1-\gamma)} s_t^{\gamma}] \end{aligned}$$

If $\gamma < 1$, higher s_{t+j} means higher p_t, whereas if $\gamma > 1$, higher s_{t+j} means lower p_t. As $1/\gamma$ is the intertemporal elasticity of consumption substitution, high γ implies that agents prefer a more smooth consumption path. News of higher consumption growth tomorrow then means that agents would want to save less/borrow today to smooth consumption. For equilibrium a' to equal 1, the returns to saving need to increase, hence p_t decreases.

(d)

The price of a tree at date $t+1$ is $p_{t+1} = E_{t+1}[\sum_{j=1}^{\infty} \beta^j s_{t+1+j}^{(1-\gamma)} s_{t+1}^{\gamma}]$. The option will be exercised if $p_{t+1} \geq \bar{p}$, so that the returns to the option is

$$\max\{0, E_{t+1}[\sum_{j=1}^{\infty} \beta^j s_{t+1+j}^{(1-\gamma)} s_{t+1}^{\gamma}] - \bar{p}\}$$

The price of the option at time t is then

$$p = E_t\{\beta s_{t+1}^{-\gamma} s_t^{\gamma} * \max\{0, E_{t+1}[\sum_{j=1}^{\infty} \beta^j s_{t+1+j}^{(1-\gamma)} s_{t+1}^{\gamma}] - \bar{p}\}\}$$

Q3

(a)

Bellman eq:

$$V(a,l) = \max_{c,a'} \{\frac{c^{1-\gamma}}{1-\gamma} + \beta E[V(a',l')|l]\}$$

s.t. $c + a' = wl + (1+r)a$, $a' \geq 0, c \geq 0$ where $E[V(a',l')|l] = \sum_{l'} V(a',l')Q(l',l)$.

Optimality:

$$c^{-\gamma} = \lambda$$
$$\lambda = \beta E[V_1(a', l')|l]$$

Envelope condition:

$$V_1(a, l) = (1 + r)\lambda$$

Combining gives the Euler eq:

$$c^{-\gamma} = \beta(1 + r)E[c'(a', l')^{-\gamma}|l]$$

See codes and inlined comments for (b), (c), (d).

In [7]:
```
using Plots, LinearAlgebra
```

In [2]:
```
### Params
const L = [0.7, 1.1]
const Q = [0.85 0.15
           0.05 0.95]
const β, γ, r, w = 0.95, 3.0, 0.03, 1.1
;
```

In [53]:
```
### Discretize asset grid
amin = 1e-10
amax = 3.0
na = 500
agrid = collect(range(amin, amax, length = na))
;
```

In [33]:
```
function u(c)
    return c^(1-γ) / (1-γ)
end
;
```

In [30]:
```
### Coding up my own function for max and argmax
function mymax(x)
    opt = -1e10
    arg = 0.0
    for i in eachindex(x)
        if x[i] >= opt
            opt = x[i]
            arg = i
        end
    end
    return opt, arg
end
;
```

(b)

In [25]:
```
### Function to find ergodic distro of a transition matrix
function ergodic(P)
    A = P - I + ones(size(P))
    B = ones(size(P)[1])
    X = A' \ B
    return X
end
;
```

Ergodic (stationary) distro:

In [27]:
```
P = ergodic(Q)
P
```

Out[27]:
```
2-element Array{Float64,1}:
 0.2500000000000001
 0.7499999999999999
```

Mean of labour:

In [29]:
```
L' * P
```

Out[29]:
```
1.0
```

(c)

In [46]:

```
function T(v, grid)
    vnext = zero(v) ### Placeholders. v is a na X 2 matrix, since we have 2 discrete states for shocks
    pol = zero(v)
    pol_arg = zero(v) ### This will be useful for the large transition matrix for (d)
    for (il, l) in enumerate(L) ### Loop for each shock state
        prob = Q[il, :]
        for (ia, a) in enumerate(grid) ### Loop for each asset state
            ynow = w * l + (1+r) * a
            val = zero(grid[ynow .- grid .> 0]) ### Only considering feasible a_next (which i call a_p)
            for (ia_p, a_p) in enumerate(grid[ynow .- grid .> 0])
                val[ia_p] = u(ynow - a_p) + β * v[ia_p, :]' * prob ### Note the expectation of future values here
            end
            opt, arg = mymax(val)
            vnext[ia, il] = opt
            pol[ia, il] = grid[arg]
            pol_arg[ia, il] = arg
        end
    end
    return vnext, pol, pol_arg
end

function VFI(vguess, grid; tol = 1e-4, maxiter = 1000)
    err = 1.0
    i = 0
    vnow = vguess
    pol, pol_arg = zero(vguess), zero(vguess)
    while err > tol && i < maxiter
        vnext, pol, pol_arg = T(vnow, grid)
        err = maximum(abs.(vnext - vnow))
        i += 1
        vnow = vnext
        if i % 80 == 1
            println("iter: ", i, " error: ", err) ### Print some stuff so we dont get impatient
        end
    end
    return vnow, pol, pol_arg
end
;
```

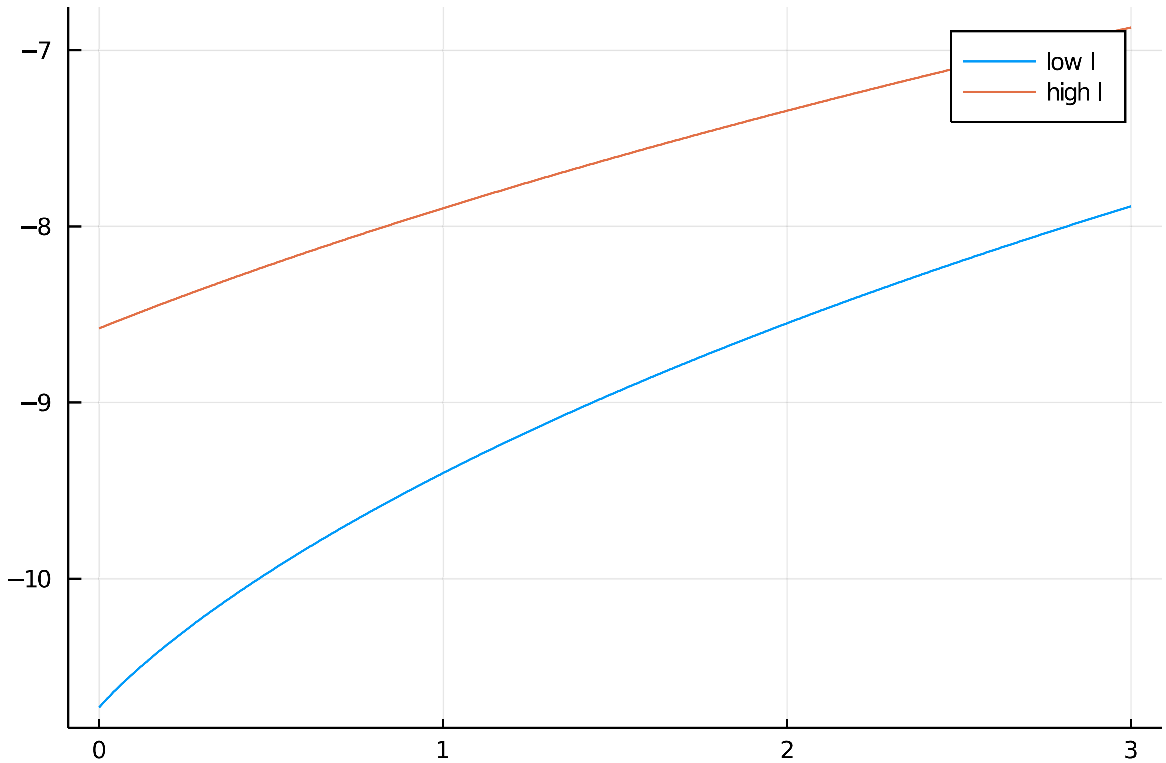
In [54]:

```
vguess = hcat(u.(agrid), u.(agrid))
val, pol, pol_arg = VFI(vguess, agrid)
;
```

```
iter: 1 error: 5.0e19
iter: 81 error: 0.0062882951788392205
iter: 161 error: 0.00010380815927746312
```

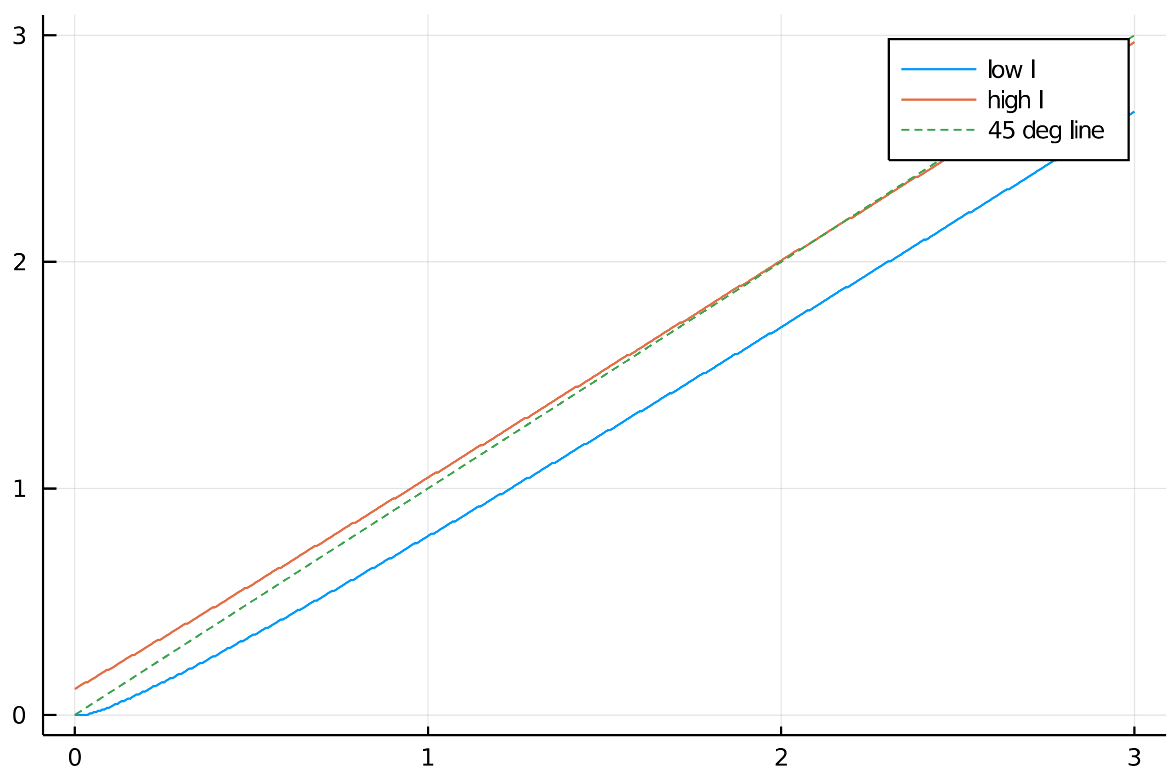
In [59]:
```
plot(agrid, val[:, 1], label = "low l")
plot!(agrid, val[:, 2], label = "high l")
```

Out[59]:



In [60]:
```
plot(agrid, pol[:, 1], label = "low l")
plot!(agrid, pol[:, 2], label = "high l")
plot!(agrid, agrid, l = :dash, label = "45 deg line")
```

Out[60]:



Note that for a > 2.2 (or something), both policy functions lie below the 45 deg line. This means that a_p(a) < a for a > 2.2. This gives us the upper bound on our asset.

(d)

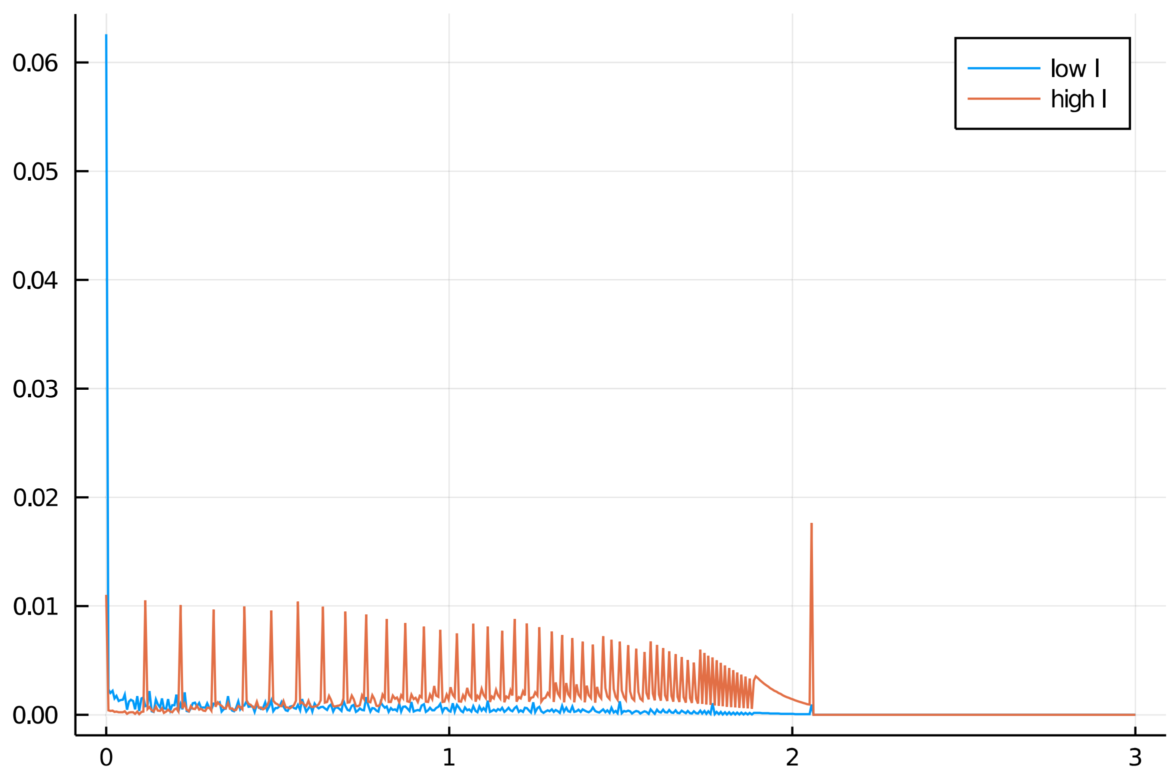
In [65]:
```
function get_trans(pol_arg, grid)
    P = zeros((2 * na, 2 * na)) ### A square matrix for total states (which is 2*na)
    for il in eachindex(L)
        prob = Q[il, :]
        for ia in eachindex(grid)
            index = Int(pol_arg[ia, il]) ### This gives us the index of the optimal anext, given our state (a, l)
            P[na * (il-1) + ia, index] += prob[1] ### A fraction goes to (anext, l = l_low)
            P[na * (il-1) + ia, na + index] += prob[2] ### A fraction goes to (anext, l = l_high)
        end
    end
    return P
end
;
```

In [66]:
```
P = get_trans(pol_arg, agrid)
;
```

In [74]:
```
ivd = ergodic(P)
;
```

In [75]:
```
plot(agrid, ivd[1:na], label = "low l")
plot!(agrid, ivd[na+1:end], label = "high l")
```

Out[75]:



Mean of assets:

In [77]:
```
ivd' * vcat(agrid, agrid)
```
Out[77]:
```
1.0221158265030454
```
In [ ]:
```

```