In the name of God

Sharif University of Thecnology

Department of Economics

Quantitative Economics

Midterm Exam - Question 1

Seyed Mohammad Mahdi Hosseini Gohar

99203047

Importing Libraries

In [1]:
```
using Distributions, Random, Plots, DataFrames, Statistics, Optim, StatsPlots
```

1. Monte-Carlo Simulation of Volunteerly Unemployment

Question's data:

- wage = w
- unemployment benefit = b
- HouseHold Utility = $U(c, l) = \frac{c^{1-\eta}}{1-\eta} - \alpha L$
- $ln(\alpha) \sim N(\mu, \sigma^2)$
- $ln(\eta) \sim N(\beta, \zeta^2)$
- productivity: $ln(z) \sim N(\theta, \Omega^2)$
- HH i receives $w_i = z_i \bar{w}$

1.1 Setup a problem and solving.

**

Since L is a discrete variable, solving the problem as below will not lead to a proper answer.

$$Max \quad U(c, l) = \frac{c^{1-\eta}}{1-\eta} - \alpha L$$

$$S.t. \quad c = (z_i \bar{w}) \times L + b \times (1 - L)$$

Therefore, we must divide the problem into two parts:

**

I) In the first problem, we set L to 1 and solve the problem.

$$Max \quad U(c, 1) = \frac{c^{1-\eta}}{1-\eta} - \alpha$$

$$S.t. \quad c = (z_i \bar{w})$$

$$\Rightarrow U_{Employment} = \frac{(z_i \bar{w})^{1-\eta}}{1-\eta} - \alpha$$

**

II) In the second problem, we set L equal to zero and solve the problem again.

$$Max \quad U(c, 0) = \frac{c^{1-\eta}}{1-\eta}$$

$$S.t. \quad c = b$$

$$\Rightarrow U_{Unemployment} = \frac{b^{1-\eta}}{1-\eta}$$

**

Now, if the utility from the employment is greater than the utility from unemployment, then the person decides to work. Otherwise, the person will not enter the labor market and will remain unemployed.

$$if \quad \frac{(z_i \bar{w})^{1-\eta}}{1-\eta} - \alpha > \frac{b^{1-\eta}}{1-\eta}$$

$$then \quad L = 1$$

$$o.w \quad L = 0$$

1.2 & 1.3 Write a code to simulate the HH's behavior and save their decisions in the memory with given parameters.

- $N = 10^3$
- $\mu = 0.5$
- $\sigma = 1$
- $\theta = 2$
- $\Omega = 3$

First we want to define Utility function for employed person:

In [2]:
```
function U_emp(z,w̄,η,α)
    UE = ( ( (z .* w̄) .^ (1 - η) ) ./ (1-η) ) .- α
    return UE
end
```

```
U_emp (generic function with 1 method)
```

Then we should define Utility function for unemployed person:

In [3]:
```
function U_une(b,η)
    UU = (b^(1-η))/(1-η)
    return UU
end
```

```
U_une (generic function with 1 method)
```

Now we define a function to determine the decision of individuals to participate in the labor market. In this function, the necessary distribution parameters are given as input. $\bar{w}$ and b and the number of simulated households are also entries for function.

In [4]:
```
function HouseHold_Choice(θ, Ω, β, ζ, μ, σ, b, w̄, HouseHold_number)
    Random.seed!(1400)
    L = zeros(HouseHold_number)
    Consumption = zeros(HouseHold_number)
    Production = zeros(HouseHold_number)
    z = rand(LogNormal(θ,Ω), HouseHold_number)
    η = rand(LogNormal(β,ζ), HouseHold_number)
    α = rand(LogNormal(μ,σ), HouseHold_number)

    for n = 1:HouseHold_number
        if U_emp(z[n],w̄,η[n],α[n]) > U_une(b,η[n])
            L[n] = 1
            Consumption[n] = z[n] .* w̄
            Production[n] = z[n]
        else
            L[n] = 0
            Consumption[n] = b
            Production[n] = 0

        end
    end
    return L, Consumption, Production
end
```

```
HouseHold_Choice (generic function with 1 method)
```

In [20]:
```
L, Consumption, Production = HouseHold_Choice(2,3,0.3,1,0.5,1,0.2,0.9,1000);
```

In [21]:
```
D = DataFrame( L = L, Consumption = Consumption, Production =Production)
```

1,000 rows × 3 columns

	L	Consumption	Production
	Float64	Float64	Float64
1	0.0	0.2	0.0
2	1.0	3823.08	4247.86
3	0.0	0.2	0.0
4	1.0	6.94739	7.71932
5	1.0	313.467	348.296
6	1.0	2.27795	2.53106
7	0.0	0.2	0.0
8	0.0	0.2	0.0
9	1.0	61.8006	68.6674
10	0.0	0.2	0.0
11	0.0	0.2	0.0
12	1.0	95.6094	106.233
13	0.0	0.2	0.0
14	1.0	49.6607	55.1786
15	1.0	343.72	381.911
16	1.0	1.44083	1.60092
17	1.0	118.261	131.401
18	1.0	7.29945	8.1105
19	0.0	0.2	0.0
20	0.0	0.2	0.0
21	1.0	350.313	389.237
22	0.0	0.2	0.0
23	0.0	0.2	0.0
24	0.0	0.2	0.0
⋮	⋮	⋮	⋮

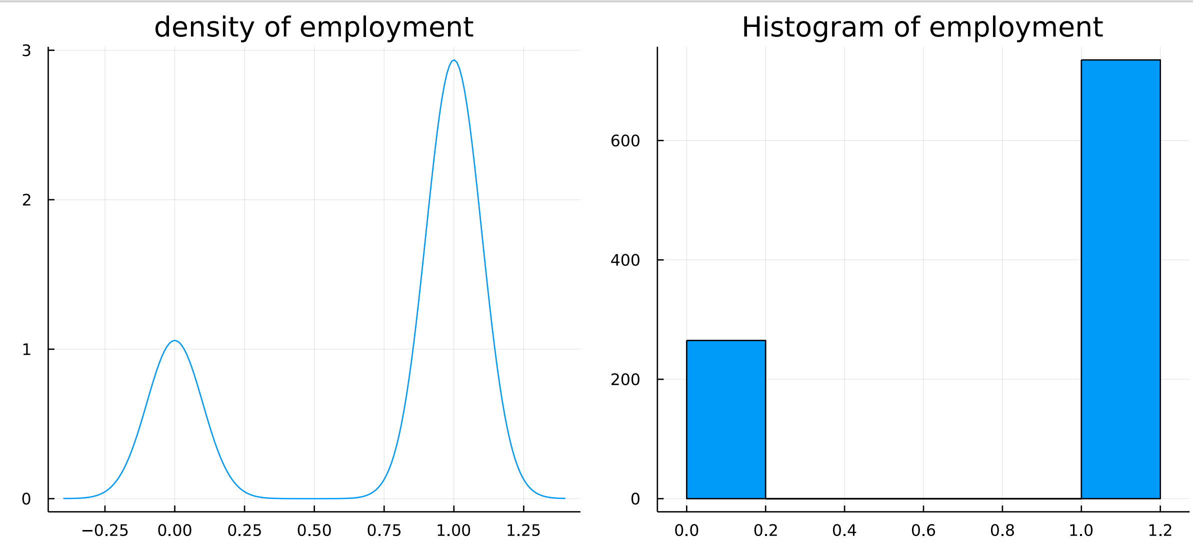
1.4 Calculate the aggregate consumption, employment and prduction for given $\bar{w}$ and b. (production: $\sum z_i L_i$)

In [23]:
```
println("The aggregate consumption is: ", sum(D.Consumption))
println("The employment rate is: ", sum(D.L)/1000)
println("The aggregate production is: ", sum(D.Production))
```

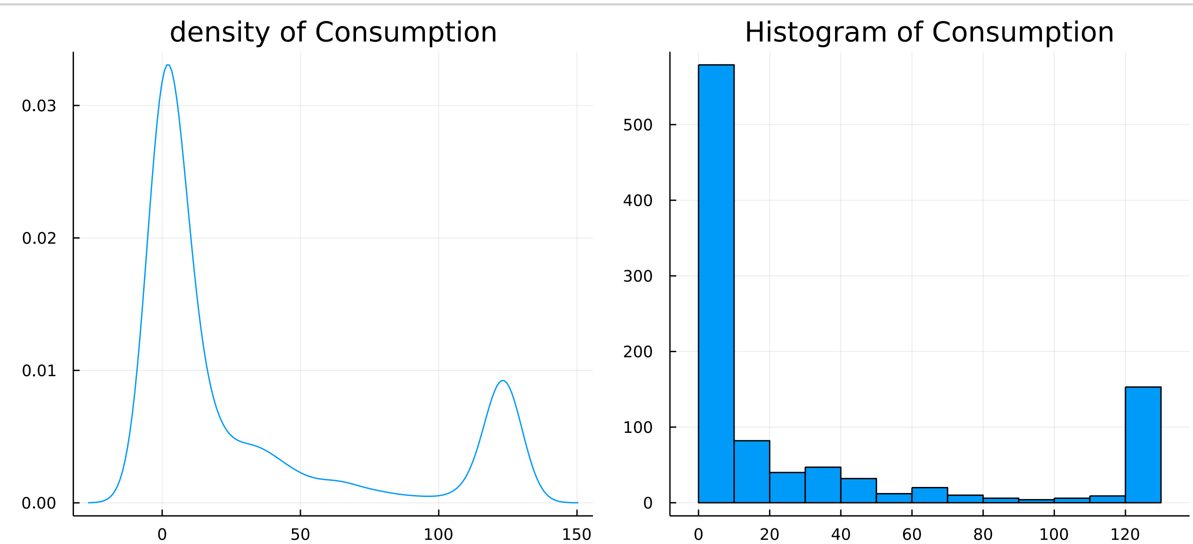
```
The aggregate consumption is: 768742.4946283352
The employment rate is: 0.735
The aggregate production is: 854099.4384759279
```

1.5 Plot the distribution of the consumption and employment

In [28]:
```
p1 = plot(density(clamp.(L, 0, quantile(L, 0.85))), title = "density of employment", legend = false)
p2 = histogram(L, title = "Histogram of employment", legend = false)
plot(p1, p2,size=(900,400))
```



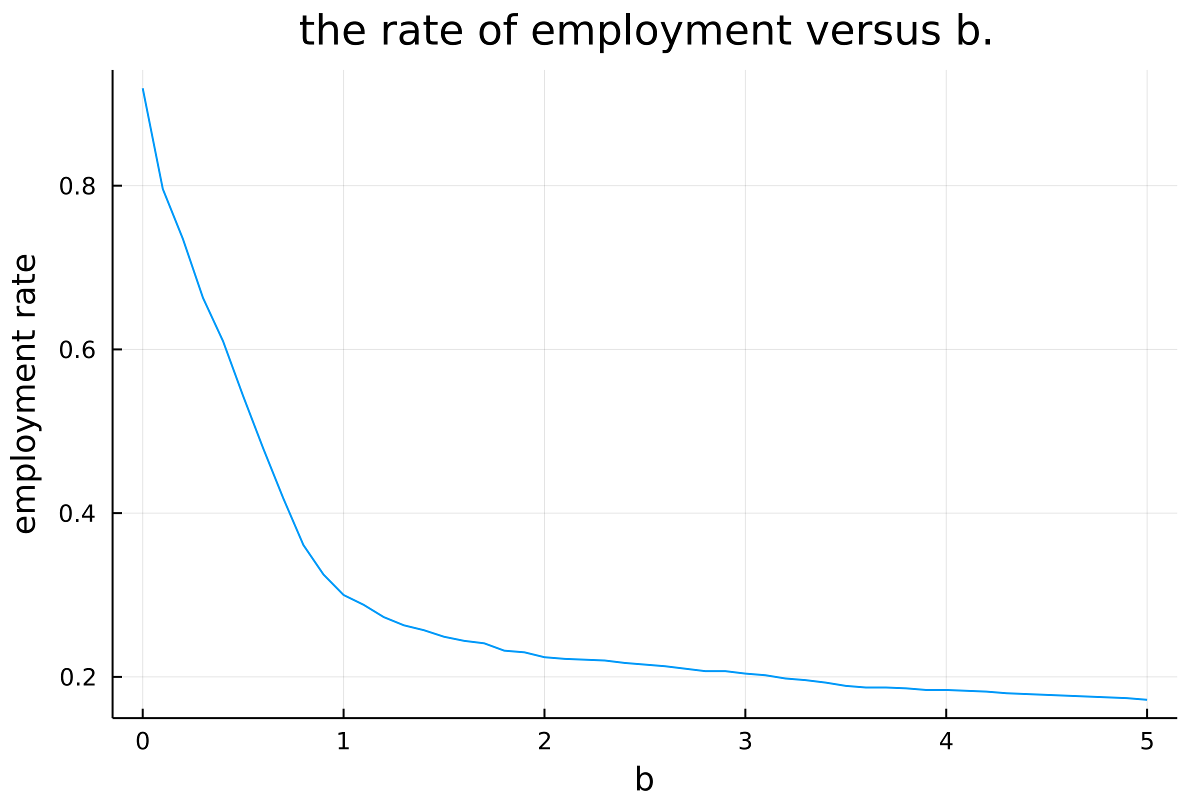
In [30]:
```
p1 = density(clamp.(Consumption, 0, quantile(Consumption, 0.85)),title = "density of Consumption", legend = false)
p2 = histogram(clamp.(Consumption, 0, quantile(Consumption, 0.85)), title = "Histogram of Consumption", legend = false)
plot(p1, p2,size=(900,400))
```



1.6 Plot the rate of employment versus b.

In [42]:
```
mea = zeros(51)
i =1
for b = 0:0.1:5
mea[i] = mean(HouseHold_Choice(2,3,0.3,1,0.5,1,b,0.9,1000)[1])
i = i+1
end
```

In [43]:
```
plot(0:0.1:5 , mea,title = "the rate of employment versus b.", legend = false,
xlabel = "b",
ylabel = "employment rate")
```



as we see above when b increases, the employment decrease. it is obviously because of rational preferences on labour and leisure desicion.

1.7 SMM to estimate θ , Ω

First of all we should define at least two moments from real data! (data that generate before) Moments:

In [44]:
```
moment1 = mean(Production[L .== 1])
moment2 = sum(L) * 1000
moment3 = mean(Consumption[L .== 1])
moment4 = std(Consumption[L .== 1])
M = [moment1, moment2, moment3, moment4]
```

```
4-element Vector{Float64}:
   1162.0400523482015
 735000.0
   1045.8360471133813
  18529.522148201027
```

now we should define simulated moment and define SMM function:

In [45]:
```
function SMM1(B)
    s_L, s_Consumption, s_Production = HouseHold_Choice(B[1], B[2], 0.3,1,0.5,
1,0.2,0.9,1000)
    Simulated_moment1 = mean(s_Production[s_L .== 1])
    Simulated_moment2 = sum(s_L) * 1000
    Simulated_moment3 = mean(s_Consumption[s_L .== 1])
    Simulated_moment4 = std(s_Consumption[s_L .== 1])
    SM = [Simulated_moment1, Simulated_moment2, Simulated_moment3, Simulated_m
oment4]

    return sum((M .- SM).^2)
end
```

```
SMM1 (generic function with 1 method)
```

at the end of the problem we should use OPTIM to minimize the least square of moments difference!

In [46]:
```
optim = optimize(SMM1, [1.99 , 2.99])
optim.minimizer
println("Estimated θ is: ",optim.minimizer[1])
println("Estimated Ω is: ",optim.minimizer[2])
```

```
Estimated θ is: 1.9999999460242965
Estimated Ω is: 3.000000014409726
```

1.8 SMM to estimate all (except θ, Ω and $\bar{w}$, b)

In [47]:
```
m1 = mean(Production[L .== 1])
m2 = mean(L) * 1000
m3 = mean(Consumption[L .== 1])
m4 = std(Consumption[L .== 1])
# m5 = skewness(Production[L .== 1])
# m6 = kurtosis(Production[L .== 1])
# m7 = skewness(Consumption[L .== 1])
# m8 = kurtosis(Consumption[L .== 1])
M2 = [m1, m2, m3, m4]
```

```
4-element Vector{Float64}:
  1162.0400523482015
   735.0
  1045.8360471133813
 18529.522148201027
```

In [48]:
```
function SMM2(Φ)
    s_L, s_Consumption, s_Production = HouseHold_Choice(2, 3, Φ[1], Φ[2], Φ[3
], Φ[4], 0.2, 0.9, 1000)
    Sm1 = mean(s_Production[s_L .== 1])
    Sm2 = mean(s_L) * 1000
    Sm3 = mean(s_Consumption[s_L .== 1])
    Sm4 = std(s_Consumption[s_L .== 1])
    # Sm5 = skewness(s_Production[s_L .== 1])
    # Sm6 = kurtosis(s_Production[s_L .== 1])
    # Sm7 = skewness(s_Consumption[s_L .== 1])
    # Sm8 = kurtosis(s_Consumption[s_L .== 1])
    SM2 = [Sm1, Sm2, Sm3, Sm4]

    return sum((M2 .- SM2).^2)
end
```

```
SMM2 (generic function with 1 method)
```

In [51]:
```
optim = optimize(SMM2, [0.25 ,0.9 ,0.45 ,0.95] )
println("Estimated β is: ",optim.minimizer[1])
println("Estimated ζ is: ",optim.minimizer[2])
println("Estimated μ is: ",optim.minimizer[3])
println("Estimated σ is: ",optim.minimizer[4])
```

```
Estimated β is: 0.18673995428384374
Estimated ζ is: 1.198389386628106
Estimated μ is: 0.38516834428037583
Estimated σ is: 1.1452160488503285
```

9. Solve for the equilibrium

in this part we assume τ as a tax ratio. with this assumption our utility of employed person change to

$$\frac{((1-\tau)*(z.*w^{-})).^{(1-\eta)}}{(1-\eta)}$$

In [52]:
```
function U_emp_tax(z,w̄,η,α,τ)
    UE = ( ( ((1-τ)*(z .* w̄)) .^ (1 - η) ) ./ (1-η) ) .- α
    return UE
end
```

```
U_emp_tax (generic function with 1 method)
```

if we have tax then consumption change to $(zw) \times (1 - \tau)$

In [53]:
```
function HouseHold_Choice_with_tax(θ, Ω, β, ζ, μ, σ, b, w̄, τ, HouseHold_number
)
    Random.seed!(1400)
    L = zeros(HouseHold_number)
    Consumption = zeros(HouseHold_number)
    Production = zeros(HouseHold_number)
    z = rand(LogNormal(θ,Ω), HouseHold_number)
    η = rand(LogNormal(β,ζ), HouseHold_number)
    α = rand(LogNormal(μ,σ), HouseHold_number)

    for n = 1:HouseHold_number
        if U_emp_tax(z[n],w̄,η[n],α[n],τ) > U_une(b,η[n])
            L[n] = 1
            Consumption[n] = (z[n] .* w̄) * (1-τ)
            Production[n] = z[n]
        else
            L[n] = 0
            Consumption[n] = b
            Production[n] = 0

        end
    end
    return L, Consumption, Production
end
```

```
HouseHold_Choice_with_tax (generic function with 1 method)
```

so to calculate the $\bar{w}$ and b we should use a function like "simul" that works like SMM but with some different! we define for described moment in the question and solve the problem using OPTIM.

In [68]:
```julia
function simul(A)
    τ = 0.02
    L, Consumption, Production = HouseHold_Choice_with_tax(2,3,0.3,1,0.5,1, A[
1], A[2],τ, 1000)

    Sm1 = sum(Consumption)
    Sm2 = sum(Production)
    wage = Consumption ./ (1 - τ)
    Sm3 = sum(τ .* (wage[L .== 1]))
    Sm4 = sum(Consumption[L .== 0])
    return ((Sm1 - Sm2)^2 + (Sm3 - Sm4)^2)
end
```

```
simul (generic function with 1 method)
```

In [69]:
```julia
optim = optimize(simul, [0.5 , 0.5] )
println("Estimated b is: ",optim.minimizer[1])
println("Estimated w̄ is: ",optim.minimizer[2])
```

```
Estimated b is: 15.233327705959741
Estimated w̄ is: 1.0000000001752247
```

if we define τ equal 2% then equilibrium value of $\bar{w}$ and b are 1 and 15.23 respectively.

In [71]:
```julia
sum(HouseHold_Choice_with_tax(2,3,0.3,1,0.5,1, 15 , 1 ,0.1, 1000)[1])
```

```
120.0
```