1 Solving for steady state

1.1 Write an equation that expresses price at period t as the discounted future flow of rent

$P_t = R_t + ((1-\delta)/(1+r))R_t + 1 + ((1-\delta)^2/(1+r)^2)R_t + 2 + \dots$

1.2 Write down steady state equations

$K_s = (1-\delta)K_s + I_s$
$R_s = P_s - (1-\delta)/(1+r))P_s$
$K_s = 100 - R_s$
$I_s = P_s$

1.3 Analytically solve steady state equations

$P_s = 100(1+r)(\delta)/(\delta^2 + (1+\delta)r + 1)$
$I_s = 100(1+r)(\delta)/(\delta^2 + (1+\delta)r + 1)$
$K_s = 100(1+r)/(\delta^2 + (1+\delta)r + 1)$
$R_s = 100 - (100(1+r)/(\delta^2 + (1+\delta)r + 1))$

1.4 How do the steady state values change if r increases? What's the intuition?

- Steady state price level should increase, assuming $\delta > 1/99$.
- Steady state investment level should increase.
- Steady state capital stock should increase.
- Steady state capital stock should decrease.

Economic Intuition: If investors can now make more money keeping money in the bank or investing elsewhere in the economy, they will need more of an incentive to continue to invest in housing, and that incentive is higher prices. These higher prices will drive an increase in capital stock which will eventualy cause the natural steady-stat price to fall as there is now more housing stock.

1.5 How do the steady state values change if delta increases? What's the intuition?

- Steady state price level should increase

- Steady state investment level should increase.

- Steady state capital stock should decrease.

- Steady state capital stock should increase.

Economic Intuition: An increase in delta means that houses depreciate more quickly, which means that more investment is required to maintain housing stock for demand, but this increase in depreciation also means that overall housing stock will go down as housing becomes a less appealing asset. and rents will increase as the PDV of a house shifts more towards the present.

1.6 Suppose demand shifts to 120 - R

If demand changes to 120 - R. this essentially means that the maximum rent that someone is willing to pay in time t has increased from 100 to 120. This upward/rightward shift in the demand curve means that steady-state price, investment, and housing stock values will increase as there becomes essentially more price discrimination in the market and the market will expand to create stock fo rthe new high-end demand, and rents should decrease as consumers now have more housing options.

1.7 Suppose gov't begins taxing housing construction. How will this change steady state?

This effectively means that there is a tax on investment, which increase the steady state price of a house, and steady state investment rates, but has no effect on rental rates at steady state or housing stock.

If the gov't decided to begin taxing rent, it obviously means that steady-state rental rates would go up with the tax, and that demand and housing stock would fall, with investment and the price of a house

2 Counterfactual analysis

Graphs to follow. /newline

3 Numerically solving for the steady state

3.1 Write a function that analytically solves for the steady state of the model in section 1 given and r as inputs into the function

```
library(ggplot2)
library(reshape2)

library(foreach)
library(doParallel)

## Loading required package:  iterators
## Loading required package:  parallel

library(parallel)

require(foreach)
require(doParallel)
require(parallel)
require(ggplot2)

numCores <- detectCores()
cl <- makeCluster(numCores)
registerDoParallel(cl)

testr1 = .2
testdelt1 = .1
testr2 = .3
testdelt2 = .9

oldsteadystatefunction <- function(r = testr, = testdelt){

Pss <- (100*(1+r)*) / ((^2) + ((1+)*r) + 1)
Iss <- Pss
Kss <- (100*(1+r)) / ((^2) + ((1+)*r) + 1)
Rss <- 100 - Kss

out <- c(Pss, Iss, Kss, Rss)
return(out)

}

oldsteadystatefunction(testr1, testdelt1)

## [1]  9.756098  9.756098 97.560976  2.439024
```

```
oldsteadystatefunction(testr2, testdelt1)

## [1]  9.701493  9.701493 97.014925  2.985075

oldsteadystatefunction(testr1, testdelt2)

## [1] 49.31507 49.31507 54.79452 45.20548

newsteadystatefunction <- function(r = testr, = testdelt){

Pss <- (120*(1+r)*) / ((^2) + ((1+)*r) + 1)
Iss <- Pss
Kss <- (120*(1+r)) / ((^2) + ((1+)*r) + 1)
Rss <- 120 - Kss

out <- c(Pss, Iss, Kss, Rss)
return(out)

}
```

3.2 Write a function that numerically solves for the steady state of the model

```
set.seed(20866)
library(nleqslv)

oldsteadystateSolve <- function(x,r,){

  P = x[1]
  I = x[2]
  K = x[3]
  R = x[4]

  F1 = ( * K) - P
  F2 = P - (((1 -)/(1 + r)) * P) - R
  F3 = 100 - R - K
  F4 = P - I

  return(c(Pss=F1, Iss=F4, Kss=F3, Rss=F2))
}

newsteadystateSolve <- function(x,r,){

  P = x[1]
  I = x[2]
  K = x[3]
  R = x[4]

  F1 = ( * K) - P
  F2 = P - (((1 -)/(1 + r)) * P) - R
  F3 = 120 - R - K
  F4 = P - I

  return(c(Pss=F1, Iss=F4, Kss=F3, Rss=F2))
}

sol1 <- nleqslv(x=c(10,10,10,10),
fn = oldsteadystateSolve, r = .2, = .1)

sol2 <- nleqslv(x=c(10,10,10,10),
fn = oldsteadystateSolve, r = .3, = .1)

sol3 <- nleqslv(x=c(10,10,10,10),
fn = oldsteadystateSolve, r = .2, = .9)

newsteadystateSolve(c(10,10,97.56098,2.439024), .2, .1)

##       Pss       Iss       Kss       Rss
## -0.243902  0.000000 19.999996  0.060976

print(sol1$x)

## [1]  9.756098  9.756098 97.560976  2.439024

print(sol2$x)

## [1]  9.701493  9.701493 97.014925  2.985075

print(sol3$x)

## [1] 49.31507 49.31507 54.79452 45.20548
```

4 Simulating economic transitions

4.1 Change in the demand for housing

4.1.1 Write function that takes previous steady state plus guess at next period price and returns the value of all parameters for the next 100 periods

```r
guess = 10

oldSS = oldsteadystatefunction(testr1, testdelt1)
newSS = newsteadystatefunction(testr1, testdelt1)

print(newSS)

## [1]  11.707317  11.707317 117.073171   2.926829

adjustPath <- function(p = guess, maxperiods = 100, newSS, oldSS,
shockperiod = 3, delta = 0.1, r = 0.2) {

  table <- matrix(ncol = 4, nrow = maxperiods)
  Period <- c(1:nrow(table))
  PlusOneColumn <- c(rep(NA, nrow(table)))
  table <- cbind(Period,table,PlusOneColumn)
  colnames(table) <- c("Period","P", "I", "K", "R", "Pt + 1")

  for (i in 1:nrow(table)){

    if (i < shockperiod){

      table[i,2:5] = oldSS

    }

    if (i == shockperiod){

      Pt = guess
      Ktminusone = table[i-1,4]

      It = Pt
      Kt = ((1 - delta) * Ktminusone) + It
      Rt = 120 - Kt
      Ptplusone = (((1 + r) / (1 - delta)) * (Pt - Rt))

      table[i,2] = guess
      table[i,3] = guess
      table[i,4] = Kt
      table[i, 5] = Rt
      table[i, 6] = Ptplusone

    }

    if (i > shockperiod){
      Pt = table[i-1, 6]
      Ktminusone = table[i-1,4]

      It = Pt
      Kt = ((1 - delta) * Ktminusone) + It
      Rt = 120 - Kt
      Ptplusone = (((1 + r) / (1 - delta)) * (Pt - Rt))

      table[i,2] = Pt
      table[i,3] = Pt
      table[i,4] = Kt
      table[i, 5] = Rt
      table[i, 6] = Ptplusone

    }

  }

  table <- table[,-6]
  ## Remove P t + 1 column

  table <- table[-14:-100,]
  ## Remove all entries exceeding 10
  ## time periods past shock for display

  return(table)

}

adjustPath(guess, 100, newSS, oldSS, 3, 0.1, 0.2)

##       Period             P             I             K            R
##  [1,]      1  9.756098e+00  9.756098e+00  9.756098e+01 2.439024e+00
##  [2,]      2  9.756098e+00  9.756098e+00  9.756098e+01 2.439024e+00
##  [3,]      3  1.000000e+01  1.000000e+01  9.780488e+01 2.219512e+01
##  [4,]      4 -1.626016e+01 -1.626016e+01  7.176423e+01 4.823577e+01
##  [5,]      5 -8.599458e+01 -8.599458e+01 -2.140678e+01 1.414068e+02
##  [6,]      6 -3.032018e+02 -3.032018e+02 -3.224679e+02 4.424679e+02
##  [7,]      7 -9.942263e+02 -9.942263e+02 -1.284447e+03 1.404447e+03
##  [8,]      8 -3.198232e+03 -3.198232e+03 -4.354234e+03 4.474234e+03
##  [9,]      9 -1.022995e+04 -1.022995e+04 -1.414877e+04 1.426877e+04
## [10,]     10 -3.266496e+04 -3.266496e+04 -4.539885e+04 4.551885e+04
## [11,]     11 -1.042451e+05 -1.042451e+05 -1.451040e+05 1.452240e+05
## [12,]     12 -3.326255e+05 -3.326255e+05 -4.632191e+05 4.633391e+05
## [13,]     13 -1.061286e+06 -1.061286e+06 -1.478183e+06 1.478303e+06
```

4.1.2 Modify the function so that rather than filling every period of maxperiods, it only fills in rows of the matrix until sum of squared diff is less than .01 OR you reach max periods

```r
guess = 8

oldSS = oldsteadystatefunction(testr1, testdelt1)
newSS = newsteadystatefunction(testr1, testdelt1)



adjustPathlimited <- function(p = guess, maxperiods = 40, newSS, oldSS,
shockperiod = 3, delta = 0.1, r = 0.2) {

  table <- matrix(ncol = 4, nrow = maxperiods)
  Period <- c(1:nrow(table))
  PlusOneColumn <- c(rep(NA, nrow(table)))
  SqDiffColumn <- c(rep(NA, nrow(table)))
  table <- cbind(Period,table,PlusOneColumn,SqDiffColumn)
  colnames(table) <- c("Period","P", "I", "K", "R", "Pt + 1", "Sq Diff")

  cycles <- 0
  totalsqdiff <- 100000000000

  while (cycles < maxperiods){

    for (i in 1:nrow(table)){
      cycles <- cycles + 1

      if (i < shockperiod){

        table[i,2:5] = oldSS

      }

      if (i == shockperiod){

        Pt = guess
        Ktminusone = table[i-1,4]

        It = Pt
        Kt = ((1 - delta) * Ktminusone) + It
        Rt = 120 - Kt
        Ptplusone = (((1 + r) / (1 - delta)) * (Pt - Rt))
        table[i,2] = guess
        table[i,3] = guess
        table[i,4] = Kt
        table[i, 5] = Rt
        table[i, 6] = Ptplusone

      }

      if (i > shockperiod){

        Pt = table[i-1, 6]
        Ktminusone = table[i-1,4]

        It = Pt
        Kt = ((1 - delta) * Ktminusone) + It
        Rt = 120 - Kt
        Ptplusone = (((1 + r) / (1 - delta)) * (Pt - Rt))

        table[i,2] = Pt
        table[i,3] = Pt
        table[i,4] = Kt
        table[i, 5] = Rt
        table[i, 6] = Ptplusone

        if(totalsqdiff < .01){
          break
        }

      }

      Psqdiff = (newSS[1] - table[i,2])^2
      Isqdiff = (newSS[2] - table[i,3])^2
      Ksqdiff = (newSS[3] - table[i,4])^2
      Rsqdiff = (newSS[4] - table[i,5])^2
      totalsqdiff = Psqdiff + Isqdiff + Ksqdiff + Rsqdiff
      table[i,7]  = totalsqdiff

    }
  }

  tablecopy <- table
  tablecopy <- tablecopy[,-6]

  return(tablecopy)
```

```
}

adjustPathlimited(guess, 40, newSS, oldSS, 3, 0.1, 0.2)

##       Period             P             I             K            R
##  [1,]      1  9.756098e+00  9.756098e+00  9.756098e+01 2.439024e+00
##  [2,]      2  9.756098e+00  9.756098e+00  9.756098e+01 2.439024e+00
##  [3,]      3  8.000000e+00  8.000000e+00  9.580488e+01 2.419512e+01
##  [4,]      4 -2.159350e+01 -2.159350e+01  6.463089e+01 5.536911e+01
##  [5,]      5 -1.026168e+02 -1.026168e+02 -4.444900e+01 1.644490e+02
##  [6,]      6 -3.560877e+02 -3.560877e+02 -3.960918e+02 5.160918e+02
##  [7,]      7 -1.162906e+03 -1.162906e+03 -1.519389e+03 1.639389e+03
##  [8,]      8 -3.736393e+03 -3.736393e+03 -5.103843e+03 5.223843e+03
##  [9,]      9 -1.194698e+04 -1.194698e+04 -1.654044e+04 1.666044e+04
## [10,]     10 -3.814323e+04 -3.814323e+04 -5.302962e+04 5.314962e+04
## [11,]     11 -1.217238e+05 -1.217238e+05 -1.694505e+05 1.695705e+05
## [12,]     12 -3.883924e+05 -3.883924e+05 -5.408978e+05 5.410178e+05
## [13,]     13 -1.239214e+06 -1.239214e+06 -1.726022e+06 1.726142e+06
## [14,]     14 -3.953807e+06 -3.953807e+06 -5.507226e+06 5.507346e+06
## [15,]     15 -1.261487e+07 -1.261487e+07 -1.757137e+07 1.757149e+07
## [16,]     16 -4.024849e+07 -4.024849e+07 -5.606272e+07 5.606284e+07
## [17,]     17 -1.284151e+08 -1.284151e+08 -1.788716e+08 1.788717e+08
## [18,]     18 -4.097157e+08 -4.097157e+08 -5.707001e+08 5.707002e+08
## [19,]     19 -1.307221e+09 -1.307221e+09 -1.820851e+09 1.820851e+09
## [20,]     20 -4.170764e+09 -4.170764e+09 -5.809530e+09 5.809530e+09
## [21,]     21 -1.330706e+10 -1.330706e+10 -1.853563e+10 1.853563e+10
## [22,]     22 -4.245692e+10 -4.245692e+10 -5.913900e+10 5.913900e+10
## [23,]     23 -1.354612e+11 -1.354612e+11 -1.886863e+11 1.886863e+11
## [24,]     24 -4.321967e+11 -4.321967e+11 -6.020144e+11 6.020144e+11
## [25,]     25 -1.378948e+12 -1.378948e+12 -1.920761e+12 1.920761e+12
## [26,]     26 -4.399612e+12 -4.399612e+12 -6.128298e+12 6.128298e+12
## [27,]     27 -1.403721e+13 -1.403721e+13 -1.955268e+13 1.955268e+13
## [28,]     28 -4.478653e+13 -4.478653e+13 -6.238394e+13 6.238394e+13
## [29,]     29 -1.428940e+14 -1.428940e+14 -1.990395e+14 1.990395e+14
## [30,]     30 -4.559113e+14 -4.559113e+14 -6.350468e+14 6.350468e+14
## [31,]     31 -1.454611e+15 -1.454611e+15 -2.026153e+15 2.026153e+15
## [32,]     32 -4.641018e+15 -4.641018e+15 -6.464556e+15 6.464556e+15
## [33,]     33 -1.480743e+16 -1.480743e+16 -2.062553e+16 2.062553e+16
## [34,]     34 -4.724395e+16 -4.724395e+16 -6.580693e+16 6.580693e+16
## [35,]     35 -1.507345e+17 -1.507345e+17 -2.099608e+17 2.099608e+17
## [36,]     36 -4.809270e+17 -4.809270e+17 -6.698917e+17 6.698917e+17
## [37,]     37 -1.534425e+18 -1.534425e+18 -2.137327e+18 2.137327e+18
## [38,]     38 -4.895670e+18 -4.895670e+18 -6.819265e+18 6.819265e+18
## [39,]     39 -1.561991e+19 -1.561991e+19 -2.175725e+19 2.175725e+19
## [40,]     40 -4.983622e+19 -4.983622e+19 -6.941774e+19 6.941774e+19
##             Sq Diff
##  [1,] 3.885782e+02
##  [2,] 3.885782e+02
##  [3,] 9.321689e+02
##  [4,] 7.718273e+03
##  [5,] 7.831883e+04
##  [6,] 7.972230e+05
##  [7,] 8.115448e+06
##  [8,] 8.261244e+07
##  [9,] 8.409660e+08
## [10,] 8.560741e+09
## [11,] 8.714537e+10
## [12,] 8.871096e+11
## [13,] 9.030467e+12
## [14,] 9.192702e+13
## [15,] 9.357851e+14
## [16,] 9.525967e+15
## [17,] 9.697103e+16
## [18,] 9.871314e+17
## [19,] 1.004865e+19
## [20,] 1.022918e+20
## [21,] 1.041295e+21
## [22,] 1.060002e+22
## [23,] 1.079045e+23
## [24,] 1.098431e+24
## [25,] 1.118164e+25
## [26,] 1.138252e+26
## [27,] 1.158701e+27
## [28,] 1.179518e+28
## [29,] 1.200708e+29
## [30,] 1.222279e+30
## [31,] 1.244238e+31
## [32,] 1.266591e+32
## [33,] 1.289345e+33
## [34,] 1.312509e+34
## [35,] 1.336088e+35
## [36,] 1.360091e+36
## [37,] 1.384526e+37
## [38,] 1.409399e+38
## [39,] 1.434719e+39
## [40,] 1.460494e+40
```

4.1.3 Write a function that calls the function above and returns the sum of squared differences between the last period produed and the steady state values

```
guess = 8

oldSS = oldsteadystatefunction(testr1, testdelt1)
newSS = newsteadystatefunction(testr1, testdelt1)

squaresfunction <- function(){

  z <- adjustPathlimited(guess, 40, newSS, oldSS, 3, 0.1, 0.2)
  z2 <- z[,6]
  out <- tail(na.omit(z2),1)
  return(out)
}

squaresfunction()

## [1] 1.460494e+40

guess = 10

squaresfunction()

## [1] 1.071208e+40
```

4.1.4 Use the "optimize" command and the function written above to solve for the optimum p3

```
optimum <- optimize(squaresfunction, maxperiods = 40, newSS = newSS, oldSS = oldSS,
shockperiod = 3, delta = 0.1, r = 0.2, lower = .1, upper = 100)[1]

## Error in f(arg, ...):  unused arguments (arg, maxperiods = 40, newSS
= c(11.7073170731707, 11.7073170731707, 117.073170731707, 2.92682926829268),
oldSS = c(9.75609756097561, 9.75609756097561, 97.5609756097561, 2.4390243902439),
shockperiod = 3, delta = 0.1, r = 0.2)

print(optimum)

## Error in print(optimum):  object 'optimum' not found
```