

# 1 Numerically solving for the steady state

## 1.1 Write a function that analytically solves for the steady state of the model in section 1 given $\beta$ 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)^2 + ((1+r)*r) + 1)
Iss <- Pss
Kss <- (100*(1+r)) / ((1+r)^2 + ((1+r)*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)*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)

}

```

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

```

set.seed(20866)
library(nleqslv)

steadystateSolve <- 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))
}

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

```

```

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

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

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

```

## 2 Simulating economic transitions

### 2.1 Change in the demand for housing

**2.1.1** 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

```

guess = 10

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

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))
  table <- cbind(Period,table)
  colnames(table) <- c("Period","P", "I", "K", "R")

  return(table)

}

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

```

##	Period	P	I	K	R	
##	[1,]	1	NA	NA	NA	NA
##	[2,]	2	NA	NA	NA	NA
##	[3,]	3	NA	NA	NA	NA
##	[4,]	4	NA	NA	NA	NA
##	[5,]	5	NA	NA	NA	NA
##	[6,]	6	NA	NA	NA	NA
##	[7,]	7	NA	NA	NA	NA
##	[8,]	8	NA	NA	NA	NA
##	[9,]	9	NA	NA	NA	NA
##	[10,]	10	NA	NA	NA	NA
##	[11,]	11	NA	NA	NA	NA
##	[12,]	12	NA	NA	NA	NA
##	[13,]	13	NA	NA	NA	NA
##	[14,]	14	NA	NA	NA	NA
##	[15,]	15	NA	NA	NA	NA
##	[16,]	16	NA	NA	NA	NA
##	[17,]	17	NA	NA	NA	NA
##	[18,]	18	NA	NA	NA	NA
##	[19,]	19	NA	NA	NA	NA
##	[20,]	20	NA	NA	NA	NA
##	[21,]	21	NA	NA	NA	NA
##	[22,]	22	NA	NA	NA	NA
##	[23,]	23	NA	NA	NA	NA
##	[24,]	24	NA	NA	NA	NA
##	[25,]	25	NA	NA	NA	NA
##	[26,]	26	NA	NA	NA	NA
##	[27,]	27	NA	NA	NA	NA
##	[28,]	28	NA	NA	NA	NA
##	[29,]	29	NA	NA	NA	NA
##	[30,]	30	NA	NA	NA	NA
##	[31,]	31	NA	NA	NA	NA
##	[32,]	32	NA	NA	NA	NA
##	[33,]	33	NA	NA	NA	NA
##	[34,]	34	NA	NA	NA	NA
##	[35,]	35	NA	NA	NA	NA
##	[36,]	36	NA	NA	NA	NA
##	[37,]	37	NA	NA	NA	NA
##	[38,]	38	NA	NA	NA	NA
##	[39,]	39	NA	NA	NA	NA
##	[40,]	40	NA	NA	NA	NA
##	[41,]	41	NA	NA	NA	NA
##	[42,]	42	NA	NA	NA	NA
##	[43,]	43	NA	NA	NA	NA
##	[44,]	44	NA	NA	NA	NA

##	[45,]	45	NA	NA	NA	NA
##	[46,]	46	NA	NA	NA	NA
##	[47,]	47	NA	NA	NA	NA
##	[48,]	48	NA	NA	NA	NA
##	[49,]	49	NA	NA	NA	NA
##	[50,]	50	NA	NA	NA	NA
##	[51,]	51	NA	NA	NA	NA
##	[52,]	52	NA	NA	NA	NA
##	[53,]	53	NA	NA	NA	NA
##	[54,]	54	NA	NA	NA	NA
##	[55,]	55	NA	NA	NA	NA
##	[56,]	56	NA	NA	NA	NA
##	[57,]	57	NA	NA	NA	NA
##	[58,]	58	NA	NA	NA	NA
##	[59,]	59	NA	NA	NA	NA
##	[60,]	60	NA	NA	NA	NA
##	[61,]	61	NA	NA	NA	NA
##	[62,]	62	NA	NA	NA	NA
##	[63,]	63	NA	NA	NA	NA
##	[64,]	64	NA	NA	NA	NA
##	[65,]	65	NA	NA	NA	NA
##	[66,]	66	NA	NA	NA	NA
##	[67,]	67	NA	NA	NA	NA
##	[68,]	68	NA	NA	NA	NA
##	[69,]	69	NA	NA	NA	NA
##	[70,]	70	NA	NA	NA	NA
##	[71,]	71	NA	NA	NA	NA
##	[72,]	72	NA	NA	NA	NA
##	[73,]	73	NA	NA	NA	NA
##	[74,]	74	NA	NA	NA	NA
##	[75,]	75	NA	NA	NA	NA
##	[76,]	76	NA	NA	NA	NA
##	[77,]	77	NA	NA	NA	NA
##	[78,]	78	NA	NA	NA	NA
##	[79,]	79	NA	NA	NA	NA
##	[80,]	80	NA	NA	NA	NA
##	[81,]	81	NA	NA	NA	NA
##	[82,]	82	NA	NA	NA	NA
##	[83,]	83	NA	NA	NA	NA
##	[84,]	84	NA	NA	NA	NA
##	[85,]	85	NA	NA	NA	NA
##	[86,]	86	NA	NA	NA	NA
##	[87,]	87	NA	NA	NA	NA
##	[88,]	88	NA	NA	NA	NA
##	[89,]	89	NA	NA	NA	NA

##	[90,]	90	NA	NA	NA	NA
##	[91,]	91	NA	NA	NA	NA
##	[92,]	92	NA	NA	NA	NA
##	[93,]	93	NA	NA	NA	NA
##	[94,]	94	NA	NA	NA	NA
##	[95,]	95	NA	NA	NA	NA
##	[96,]	96	NA	NA	NA	NA
##	[97,]	97	NA	NA	NA	NA
##	[98,]	98	NA	NA	NA	NA
##	[99,]	99	NA	NA	NA	NA
##	[100,]	100	NA	NA	NA	NA