Stochastic homework 2

kallakuri radhakrishna

29/05/2020

example 1

a

```
#states
states <- c("own","com. rent.","soc. rent.")

#transition matrix

tmat <- matrix(data=c(.6,.3,.1,
                      .4,.4,.2,
                      .2,.4,.4), byrow=TRUE, nrow=3,
               dimnames=list(states,states))

tmat
```

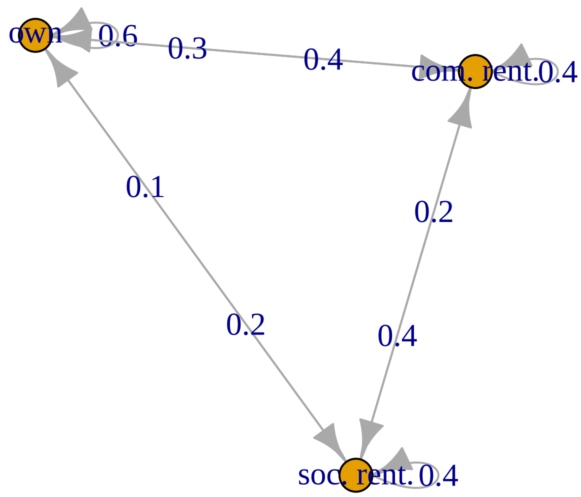
```
##             own com. rent. soc. rent.
## own         0.6        0.3        0.1
## com. rent. 0.4        0.4        0.2
## soc. rent. 0.2        0.4        0.4
```

```
library("markovchain")
```

```
## Warning: package 'markovchain' was built under R version 3.6.3
```

```
## Package:  markovchain
## Version:  0.8.4.1
## Date:     2020-05-04
## BugReport: http://github.com/spedygiorgio/markovchain/issues
```

```
# Markov chain
mc <- new("markovchain", states = states, byrow = TRUE, transitionMatrix = tmat, name = "residence clas

# transition diagram
plot(mc)
```



b

the probability that a granddaughter lives in her own apartment if her grandmother was a social renter is 0.36

```
(mc^2)
```

```
## residence classes^2
##  A  3 - dimensional discrete Markov Chain defined by the following states:
##  own, com. rent., soc. rent.
##  The transition matrix  (by rows)  is defined as follows:
##              own com. rent. soc. rent.
## own        0.50       0.34       0.16
## com. rent. 0.44       0.36       0.20
## soc. rent. 0.36       0.38       0.26
```

```
# transitionafter two steps from  parents soc. rent. -> own apartment
(mc^2)["soc. rent.","own"]
```

```
## [1] 0.36
```

c

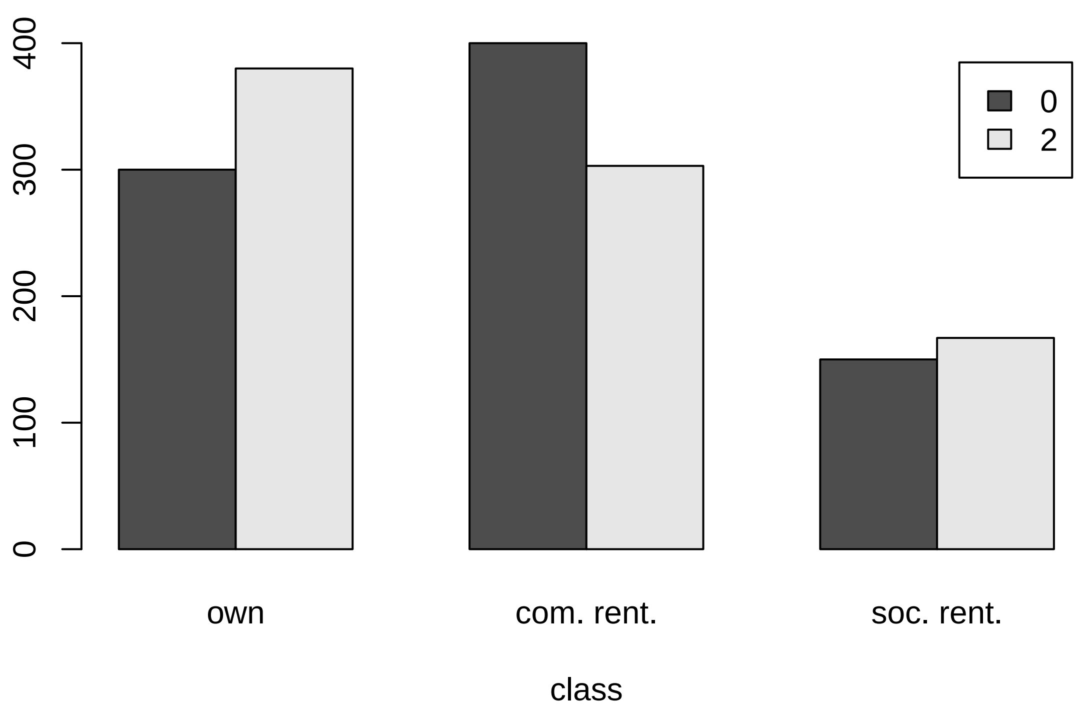
Reseidence structure after two generations

```
q0 <- c(300,400,150)
```

```
dist <-  q0*(mc^2) # after two generations
print(dist)
```

```
##      own com. rent. soc. rent.
## [1,] 380        303        167
```

```
steps <- c(0,2)
dist1 <- rbind(q0, q0*(mc^2))
dimnames(dist1) <- list(steps, states)

barplot(dist1, beside = T, xlab = "class", legend = steps)
```



d

yes the chain is regular as we can see higher powers of markov chain are all having same rows. The distribution after a long time therefore does not depend on the initial distribution. That is also equal to the steady state.

```
steadyStates(mc)
```

```
##            own com. rent. soc. rent.
## [1,] 0.4516129  0.3548387  0.1935484
```

```
mc^100
```

```
## residence classes^100
##  A  3 - dimensional discrete Markov Chain defined by the following states:
##  own, com. rent., soc. rent.
```

```
##  The transition matrix  (by rows)  is defined as follows:
##                   own com. rent. soc. rent.
## own        0.4516129  0.3548387  0.1935484
## com. rent. 0.4516129  0.3548387  0.1935484
## soc. rent. 0.4516129  0.3548387  0.1935484
```

```
mc^500
```

```
## residence classes^500
##  A  3 - dimensional discrete Markov Chain defined by the following states:
##  own, com. rent., soc. rent.
##  The transition matrix  (by rows)  is defined as follows:
##                   own com. rent. soc. rent.
## own        0.4516129  0.3548387  0.1935484
## com. rent. 0.4516129  0.3548387  0.1935484
## soc. rent. 0.4516129  0.3548387  0.1935484
```

```
mc^1000
```

```
## residence classes^1000
##  A  3 - dimensional discrete Markov Chain defined by the following states:
##  own, com. rent., soc. rent.
##  The transition matrix  (by rows)  is defined as follows:
##                   own com. rent. soc. rent.
## own        0.4516129  0.3548387  0.1935484
## com. rent. 0.4516129  0.3548387  0.1935484
## soc. rent. 0.4516129  0.3548387  0.1935484
```

e

long term residence structure own=383.8710, com.rent=301.6129, soc.rent=164.5161

```
steps <- c(0, 1, 5, 10, 50,60,100)
initialState <- c(300,400,150)
pred <- c(); # create an empty matrix
for (k in steps) {
  pr <- initialState*(mc^k) # calculate the distribution after k steps
pred <- rbind(pred, pr) # add to the list of of distributions
}
# names of rows and columns
dimnames(pred) <- list(steps, states)
pred
```

```
##          own com. rent. soc. rent.
## 0   300.0000   400.0000   150.0000
## 1   370.0000   310.0000   170.0000
## 5   383.7240   301.6560   164.6200
## 10  383.8703   301.6131   164.5166
## 50  383.8710   301.6129   164.5161
## 60  383.8710   301.6129   164.5161
## 100 383.8710   301.6129   164.5161
```

f

prais mobility index is 0.8

```
pi <- (3-sum(diag(tmat)))/(3-1)
pi
```

```
## [1] 0.8
```

example 2

adjacency matrix

```
states2 <- c()
for(i in 1:3)
{
  for(j in 1:3)
  {

states2 <- c(states2, c(i,j))
  }
}
states2
```

```
##  [1] 1 1 1 2 1 3 2 1 2 2 2 3 3 1 3 2 3 3
```

```
states3 <- c("(1,1)","(1,2)","(1,3)","(2,1)","(2,2)","(2,3)","(3,1)","(3,2)","(3,3)")
states3
```

```
## [1] "(1,1)" "(1,2)" "(1,3)" "(2,1)" "(2,2)" "(2,3)" "(3,1)" "(3,2)" "(3,3)"
```

```
adjmat <- matrix(data=c(0,1,0,1,0,0,0,0,0,
                        1,0,1,0,1,0,0,0,0,
                        0,1,0,0,0,1,0,0,0,
                        1,0,0,0,1,0,1,0,0,
                        0,1,0,1,0,1,0,1,0,
                        0,0,1,0,1,0,0,0,1,
                        0,0,0,1,0,0,0,1,0,
                        0,0,0,0,1,0,1,0,1,
                        0,0,0,0,0,1,0,1,0), byrow=TRUE, nrow=9,
               dimnames=list(states3,states3))
```

a

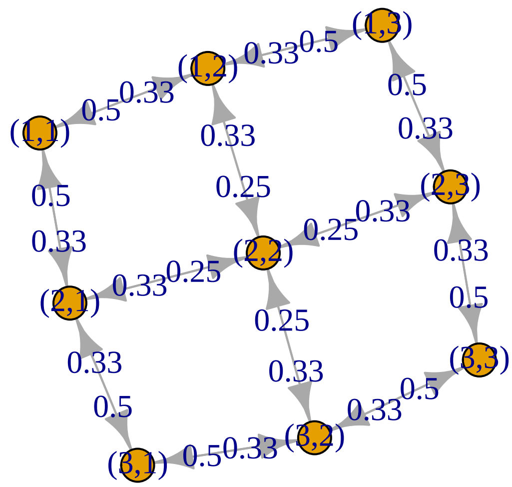
```
adjmat
```

```
##       (1,1) (1,2) (1,3) (2,1) (2,2) (2,3) (3,1) (3,2) (3,3)
## (1,1)     0     1     0     1     0     0     0     0     0
## (1,2)     1     0     1     0     1     0     0     0     0
## (1,3)     0     1     0     0     0     1     0     0     0
## (2,1)     1     0     0     0     1     0     1     0     0
## (2,2)     0     1     0     1     0     1     0     1     0
## (2,3)     0     0     1     0     1     0     0     0     1
## (3,1)     0     0     0     1     0     0     0     1     0
## (3,2)     0     0     0     0     1     0     1     0     1
## (3,3)     0     0     0     0     0     1     0     1     0
```

```r
rs <- rowSums(adjmat)
trmat <- diag(1/rs) %*% adjmat
trmat
```

```
##          (1,1) (1,2)     (1,3) (2,1)     (2,2) (2,3)     (3,1) (3,2)
##  [1,] 0.0000000  0.50 0.0000000  0.50 0.0000000  0.00 0.0000000  0.00
##  [2,] 0.3333333  0.00 0.3333333  0.00 0.3333333  0.00 0.0000000  0.00
##  [3,] 0.0000000  0.50 0.0000000  0.00 0.0000000  0.50 0.0000000  0.00
##  [4,] 0.3333333  0.00 0.0000000  0.00 0.3333333  0.00 0.3333333  0.00
##  [5,] 0.0000000  0.25 0.0000000  0.25 0.0000000  0.25 0.0000000  0.25
##  [6,] 0.0000000  0.00 0.3333333  0.00 0.3333333  0.00 0.0000000  0.00
##  [7,] 0.0000000  0.00 0.0000000  0.50 0.0000000  0.00 0.0000000  0.50
##  [8,] 0.0000000  0.00 0.0000000  0.00 0.3333333  0.00 0.3333333  0.00
##  [9,] 0.0000000  0.00 0.0000000  0.00 0.0000000  0.50 0.0000000  0.50
##          (3,3)
##  [1,] 0.0000000
##  [2,] 0.0000000
##  [3,] 0.0000000
##  [4,] 0.0000000
##  [5,] 0.0000000
##  [6,] 0.3333333
##  [7,] 0.0000000
##  [8,] 0.3333333
##  [9,] 0.0000000
```

```r
rwgraph <- new("markovchain", states = states3, byrow = TRUE,
               transitionMatrix = trmat, name = "Random walk")
plot(rwgraph)
```



b

```
# simulation
simulation1 <- rmarkovchain(10, rwgraph, t0 = "(1,1)")
simulation1
```

```
##  [1] "(2,1)" "(3,1)" "(2,1)" "(3,1)" "(2,1)" "(2,2)" "(2,3)" "(3,3)"
##  [9] "(3,2)" "(3,3)"
```

c

As even 10th adn 11th power of the transition matrix does not contain only strictly positive elements, we suspect that the matrix is not regular

```
dist3 <- rwgraph^10
dist3
```

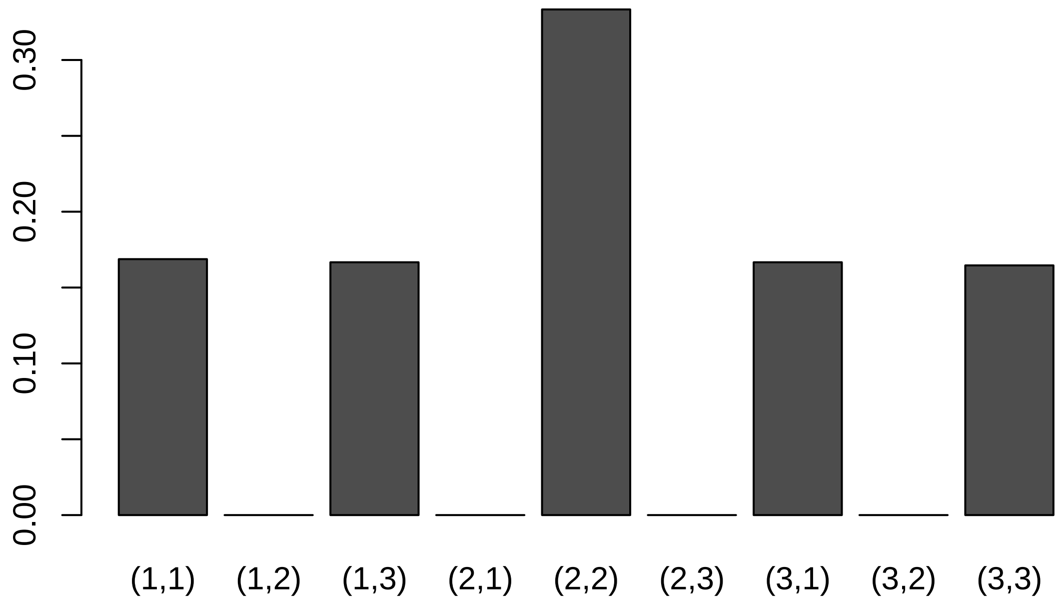
```
## Random walk^10
##  A  9 - dimensional discrete Markov Chain defined by the following states:
##  (1,1), (1,2), (1,3), (2,1), (2,2), (2,3), (3,1), (3,2), (3,3)
##  The transition matrix  (by rows)  is defined as follows:
##           (1,1)     (1,2)     (1,3)     (2,1)     (2,2)     (2,3)
## (1,1) 0.1687243 0.0000000 0.1666667 0.0000000 0.3333333 0.0000000
## (1,2) 0.0000000 0.2520576 0.0000000 0.2500000 0.0000000 0.2500000
## (1,3) 0.1666667 0.0000000 0.1687243 0.0000000 0.3333333 0.0000000
## (2,1) 0.0000000 0.2500000 0.0000000 0.2520576 0.0000000 0.2479424
```

```
## (2,2) 0.1666667 0.0000000 0.1666667 0.0000000 0.3333333 0.0000000
## (2,3) 0.0000000 0.2500000 0.0000000 0.2479424 0.0000000 0.2520576
## (3,1) 0.1666667 0.0000000 0.1646091 0.0000000 0.3333333 0.0000000
## (3,2) 0.0000000 0.2479424 0.0000000 0.2500000 0.0000000 0.2500000
## (3,3) 0.1646091 0.0000000 0.1666667 0.0000000 0.3333333 0.0000000
##           (3,1)     (3,2)     (3,3)
## (1,1) 0.1666667 0.0000000 0.1646091
## (1,2) 0.0000000 0.2479424 0.0000000
## (1,3) 0.1646091 0.0000000 0.1666667
## (2,1) 0.0000000 0.2500000 0.0000000
## (2,2) 0.1666667 0.0000000 0.1666667
## (2,3) 0.0000000 0.2500000 0.0000000
## (3,1) 0.1687243 0.0000000 0.1666667
## (3,2) 0.0000000 0.2520576 0.0000000
## (3,3) 0.1666667 0.0000000 0.1687243
```

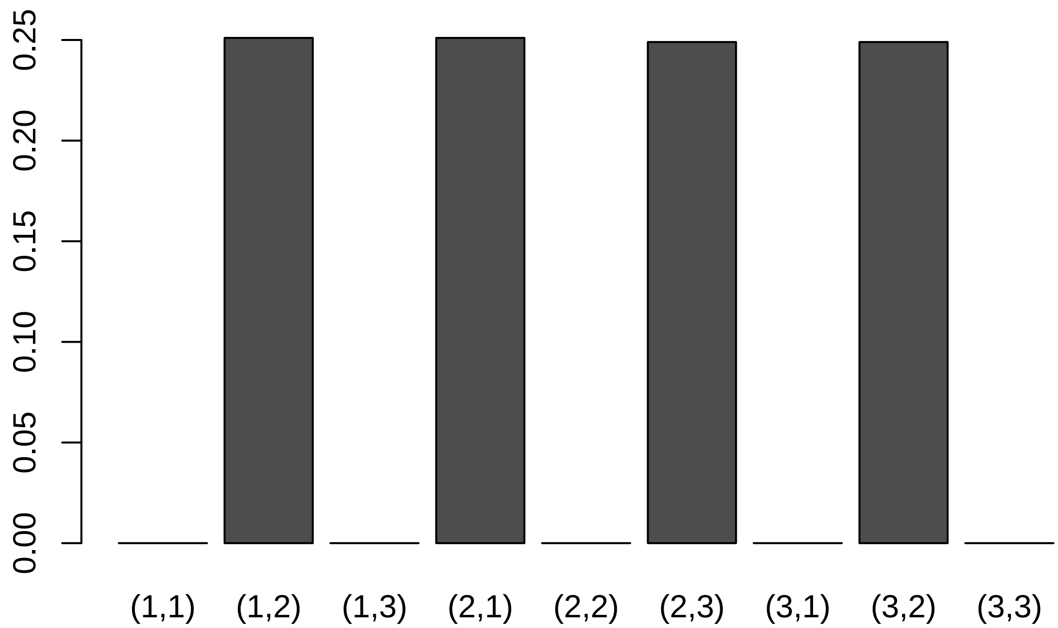
```r
dist4 <- rwgraph^11
dist4
```

```
## Random walk^11
##  A  9 - dimensional discrete Markov Chain defined by the following states:
##  (1,1), (1,2), (1,3), (2,1), (2,2), (2,3), (3,1), (3,2), (3,3)
##  The transition matrix  (by rows)  is defined as follows:
##           (1,1)     (1,2)     (1,3)     (2,1)     (2,2)     (2,3)
## (1,1) 0.0000000 0.2510288 0.0000000 0.2510288 0.0000000 0.2489712
## (1,2) 0.1673525 0.0000000 0.1673525 0.0000000 0.3333333 0.0000000
## (1,3) 0.0000000 0.2510288 0.0000000 0.2489712 0.0000000 0.2510288
## (2,1) 0.1673525 0.0000000 0.1659808 0.0000000 0.3333333 0.0000000
## (2,2) 0.0000000 0.2500000 0.0000000 0.2500000 0.0000000 0.2500000
## (2,3) 0.1659808 0.0000000 0.1673525 0.0000000 0.3333333 0.0000000
## (3,1) 0.0000000 0.2489712 0.0000000 0.2510288 0.0000000 0.2489712
## (3,2) 0.1659808 0.0000000 0.1659808 0.0000000 0.3333333 0.0000000
## (3,3) 0.0000000 0.2489712 0.0000000 0.2489712 0.0000000 0.2510288
##           (3,1)     (3,2)     (3,3)
## (1,1) 0.0000000 0.2489712 0.0000000
## (1,2) 0.1659808 0.0000000 0.1659808
## (1,3) 0.0000000 0.2489712 0.0000000
## (2,1) 0.1673525 0.0000000 0.1659808
## (2,2) 0.0000000 0.2500000 0.0000000
## (2,3) 0.1659808 0.0000000 0.1673525
## (3,1) 0.0000000 0.2510288 0.0000000
## (3,2) 0.1673525 0.0000000 0.1673525
## (3,3) 0.0000000 0.2510288 0.0000000
```

```r
Q0 <- c(1, 0, 0, 0, 0, 0, 0, 0, 0)
Q10 <- Q0*(rwgraph^10)
barplot(Q10)
```




```
Q11 <- Q0*(rwgraph^11)
barplot(Q11)
```



d even at higher powers it does not have strictly positive elements. Hence, the chain is not regular. also it can be seen that there are two rows which are repeating itself.

```
rwgraph^50
```

```
## Random walk^50
##  A  9 - dimensional discrete Markov Chain defined by the following states:
##  (1,1), (1,2), (1,3), (2,1), (2,2), (2,3), (3,1), (3,2), (3,3)
##  The transition matrix  (by rows)  is defined as follows:
##           (1,1) (1,2)     (1,3) (2,1)     (2,2) (2,3)     (3,1) (3,2)
## (1,1) 0.1666667  0.00 0.1666667  0.00 0.3333333  0.00 0.1666667  0.00
## (1,2) 0.0000000  0.25 0.0000000  0.25 0.0000000  0.25 0.0000000  0.25
## (1,3) 0.1666667  0.00 0.1666667  0.00 0.3333333  0.00 0.1666667  0.00
## (2,1) 0.0000000  0.25 0.0000000  0.25 0.0000000  0.25 0.0000000  0.25
## (2,2) 0.1666667  0.00 0.1666667  0.00 0.3333333  0.00 0.1666667  0.00
## (2,3) 0.0000000  0.25 0.0000000  0.25 0.0000000  0.25 0.0000000  0.25
## (3,1) 0.1666667  0.00 0.1666667  0.00 0.3333333  0.00 0.1666667  0.00
## (3,2) 0.0000000  0.25 0.0000000  0.25 0.0000000  0.25 0.0000000  0.25
## (3,3) 0.1666667  0.00 0.1666667  0.00 0.3333333  0.00 0.1666667  0.00
##           (3,3)
## (1,1) 0.1666667
## (1,2) 0.0000000
## (1,3) 0.1666667
## (2,1) 0.0000000
## (2,2) 0.1666667
## (2,3) 0.0000000
## (3,1) 0.1666667
```

```
## (3,2) 0.0000000
## (3,3) 0.1666667
```

```r
rwgraph^75
```

```
## Random walk^75
##  A  9 - dimensional discrete Markov Chain defined by the following states:
##  (1,1), (1,2), (1,3), (2,1), (2,2), (2,3), (3,1), (3,2), (3,3)
##  The transition matrix  (by rows)  is defined as follows:
##           (1,1) (1,2)     (1,3) (2,1)     (2,2) (2,3)     (3,1) (3,2)
## (1,1) 0.0000000  0.25 0.0000000  0.25 0.0000000  0.25 0.0000000  0.25
## (1,2) 0.1666667  0.00 0.1666667  0.00 0.3333333  0.00 0.1666667  0.00
## (1,3) 0.0000000  0.25 0.0000000  0.25 0.0000000  0.25 0.0000000  0.25
## (2,1) 0.1666667  0.00 0.1666667  0.00 0.3333333  0.00 0.1666667  0.00
## (2,2) 0.0000000  0.25 0.0000000  0.25 0.0000000  0.25 0.0000000  0.25
## (2,3) 0.1666667  0.00 0.1666667  0.00 0.3333333  0.00 0.1666667  0.00
## (3,1) 0.0000000  0.25 0.0000000  0.25 0.0000000  0.25 0.0000000  0.25
## (3,2) 0.1666667  0.00 0.1666667  0.00 0.3333333  0.00 0.1666667  0.00
## (3,3) 0.0000000  0.25 0.0000000  0.25 0.0000000  0.25 0.0000000  0.25
##           (3,3)
## (1,1) 0.0000000
## (1,2) 0.1666667
## (1,3) 0.0000000
## (2,1) 0.1666667
## (2,2) 0.0000000
## (2,3) 0.1666667
## (3,1) 0.0000000
## (3,2) 0.1666667
## (3,3) 0.0000000
```

```r
rwgraph^100
```

```
## Random walk^100
##  A  9 - dimensional discrete Markov Chain defined by the following states:
##  (1,1), (1,2), (1,3), (2,1), (2,2), (2,3), (3,1), (3,2), (3,3)
##  The transition matrix  (by rows)  is defined as follows:
##           (1,1) (1,2)     (1,3) (2,1)     (2,2) (2,3)     (3,1) (3,2)
## (1,1) 0.1666667  0.00 0.1666667  0.00 0.3333333  0.00 0.1666667  0.00
## (1,2) 0.0000000  0.25 0.0000000  0.25 0.0000000  0.25 0.0000000  0.25
## (1,3) 0.1666667  0.00 0.1666667  0.00 0.3333333  0.00 0.1666667  0.00
## (2,1) 0.0000000  0.25 0.0000000  0.25 0.0000000  0.25 0.0000000  0.25
## (2,2) 0.1666667  0.00 0.1666667  0.00 0.3333333  0.00 0.1666667  0.00
## (2,3) 0.0000000  0.25 0.0000000  0.25 0.0000000  0.25 0.0000000  0.25
## (3,1) 0.1666667  0.00 0.1666667  0.00 0.3333333  0.00 0.1666667  0.00
## (3,2) 0.0000000  0.25 0.0000000  0.25 0.0000000  0.25 0.0000000  0.25
## (3,3) 0.1666667  0.00 0.1666667  0.00 0.3333333  0.00 0.1666667  0.00
##           (3,3)
## (1,1) 0.1666667
## (1,2) 0.0000000
## (1,3) 0.1666667
## (2,1) 0.0000000
## (2,2) 0.1666667
## (2,3) 0.0000000
## (3,1) 0.1666667
## (3,2) 0.0000000
```

```
## (3,3) 0.1666667
```

e

from previous example we can say that there are two invariant distributions 0.1666667 0.00 0.1666667 0.00 0.3333333 0.00 0.1666667 0.00 0.1666667 0.0000000 0.25 0.0000000 0.25 0.0000000 0.25 0.0000000 0.25 0.0000000 we can see that markov chain also uniquely converges despite not being regular. Regularity is therefore a suffcient but not necessary condition for unique convergence.

```r
invariant <- steadyStates(rwgraph)
invariant
```

```
##            (1,1) (1,2)      (1,3) (2,1)     (2,2) (2,3)      (3,1) (3,2)
## [1,] 0.08333333 0.125 0.08333333 0.125 0.1666667 0.125 0.08333333 0.125
##            (3,3)
## [1,] 0.08333333
```

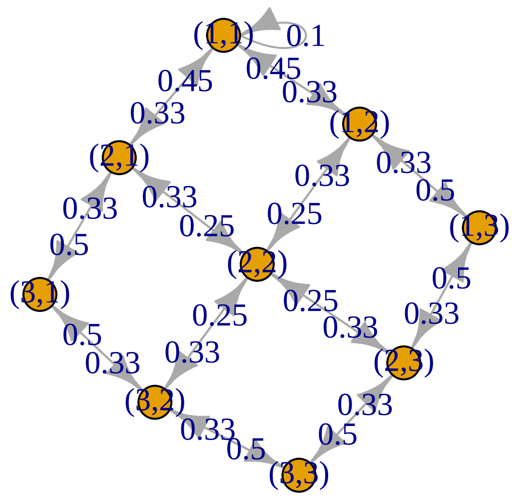
f

modified markov chain

```r
trmat[1,1] <- 0.1

trmat[1,2] <- 0.45
trmat[1,4] <- 0.45

trmat1 <- trmat
trmat1
```

```
##            (1,1) (1,2)     (1,3) (2,1)     (2,2) (2,3)     (3,1) (3,2)
##  [1,] 0.1000000  0.45 0.0000000  0.45 0.0000000  0.00 0.0000000  0.00
##  [2,] 0.3333333  0.00 0.3333333  0.00 0.3333333  0.00 0.0000000  0.00
##  [3,] 0.0000000  0.50 0.0000000  0.00 0.0000000  0.50 0.0000000  0.00
##  [4,] 0.3333333  0.00 0.0000000  0.00 0.3333333  0.00 0.3333333  0.00
##  [5,] 0.0000000  0.25 0.0000000  0.25 0.0000000  0.25 0.0000000  0.25
##  [6,] 0.0000000  0.00 0.3333333  0.00 0.3333333  0.00 0.0000000  0.00
##  [7,] 0.0000000  0.00 0.0000000  0.50 0.0000000  0.00 0.0000000  0.50
##  [8,] 0.0000000  0.00 0.0000000  0.00 0.3333333  0.00 0.3333333  0.00
##  [9,] 0.0000000  0.00 0.0000000  0.00 0.0000000  0.50 0.0000000  0.50
##            (3,3)
##  [1,] 0.0000000
##  [2,] 0.0000000
##  [3,] 0.0000000
##  [4,] 0.0000000
##  [5,] 0.0000000
##  [6,] 0.3333333
##  [7,] 0.0000000
##  [8,] 0.3333333
##  [9,] 0.0000000
```

```r
rwgraph1 <- new("markovchain", states = states3, byrow = TRUE,
                transitionMatrix = trmat1, name = "Random walk1")
plot(rwgraph1)
```



g

yes the chain is regular, as it now has strictly positive elements which are all equal as can be seen in its 1000th power. Each row has the limit distribution.

```
rwgraph1^500
```

```
## Random walk1^500
##  A  9 - dimensional discrete Markov Chain defined by the following states:
##  (1,1), (1,2), (1,3), (2,1), (2,2), (2,3), (3,1), (3,2), (3,3)
##  The transition matrix  (by rows)  is defined as follows:
##            (1,1)     (1,2)      (1,3)     (2,1)     (2,2)     (2,3)
## (1,1) 0.09177935 0.1237942 0.08261063 0.1237942 0.1652213 0.1237886
## (1,2) 0.09169943 0.1239243 0.08251838 0.1239243 0.1650368 0.1239311
## (1,3) 0.09178958 0.1237776 0.08262243 0.1237776 0.1652449 0.1237703
## (2,1) 0.09169943 0.1239243 0.08251838 0.1239243 0.1650368 0.1239311
## (2,2) 0.09178958 0.1237776 0.08262243 0.1237776 0.1652449 0.1237703
## (2,3) 0.09169525 0.1239311 0.08251355 0.1239311 0.1650271 0.1239386
## (3,1) 0.09178958 0.1237776 0.08262243 0.1237776 0.1652449 0.1237703
## (3,2) 0.09169525 0.1239311 0.08251355 0.1239311 0.1650271 0.1239386
## (3,3) 0.09179170 0.1237741 0.08262488 0.1237741 0.1652498 0.1237665
##            (3,1)     (3,2)      (3,3)
## (1,1) 0.08261063 0.1237886 0.08261253
## (1,2) 0.08251838 0.1239311 0.08251608
## (1,3) 0.08262243 0.1237703 0.08262488
## (2,1) 0.08251838 0.1239311 0.08251608
```

```
## (2,2) 0.08262243 0.1237703 0.08262488
## (2,3) 0.08251355 0.1239386 0.08251103
## (3,1) 0.08262243 0.1237703 0.08262488
## (3,2) 0.08251355 0.1239386 0.08251103
## (3,3) 0.08262488 0.1237665 0.08262744
```

```r
rwgraph1^1000
```

```
## Random walk1^1000
##  A  9 - dimensional discrete Markov Chain defined by the following states:
##  (1,1), (1,2), (1,3), (2,1), (2,2), (2,3), (3,1), (3,2), (3,3)
##  The transition matrix  (by rows)  is defined as follows:
##            (1,1)     (1,2)      (1,3)     (2,1)     (2,2)     (2,3)
## (1,1) 0.09174314 0.1238532 0.08256883 0.1238532 0.1651377 0.1238532
## (1,2) 0.09174309 0.1238533 0.08256878 0.1238533 0.1651376 0.1238533
## (1,3) 0.09174315 0.1238532 0.08256884 0.1238532 0.1651377 0.1238532
## (2,1) 0.09174309 0.1238533 0.08256878 0.1238533 0.1651376 0.1238533
## (2,2) 0.09174315 0.1238532 0.08256884 0.1238532 0.1651377 0.1238532
## (2,3) 0.09174309 0.1238533 0.08256877 0.1238533 0.1651375 0.1238533
## (3,1) 0.09174315 0.1238532 0.08256884 0.1238532 0.1651377 0.1238532
## (3,2) 0.09174309 0.1238533 0.08256877 0.1238533 0.1651375 0.1238533
## (3,3) 0.09174315 0.1238532 0.08256884 0.1238532 0.1651377 0.1238532
##            (3,1)     (3,2)      (3,3)
## (1,1) 0.08256883 0.1238532 0.08256883
## (1,2) 0.08256878 0.1238533 0.08256877
## (1,3) 0.08256884 0.1238532 0.08256884
## (2,1) 0.08256878 0.1238533 0.08256877
## (2,2) 0.08256884 0.1238532 0.08256884
## (2,3) 0.08256877 0.1238533 0.08256877
## (3,1) 0.08256884 0.1238532 0.08256884
## (3,2) 0.08256877 0.1238533 0.08256877
## (3,3) 0.08256884 0.1238532 0.08256884
```

h

as we can see that steadystates function is giving an output which is numerically similar to all the rows of rwgraph1^1000 hence it is again proved that it is a regular markov chain which has a unique convergence.

```r
invariant1 <- steadyStates(rwgraph1)
invariant1
```

```
##           (1,1)     (1,2)      (1,3)     (2,1)     (2,2)     (2,3)
## [1,] 0.09174312 0.1238532 0.08256881 0.1238532 0.1651376 0.1238532
##           (3,1)     (3,2)      (3,3)
## [1,] 0.08256881 0.1238532 0.08256881
```

i

```r
print(invariant)
```

```
##           (1,1) (1,2)      (1,3) (2,1)     (2,2) (2,3)      (3,1) (3,2)
## [1,] 0.08333333 0.125 0.08333333 0.125 0.1666667 0.125 0.08333333 0.125
##           (3,3)
## [1,] 0.08333333
```

```r
print(invariant1)
```

```
##           (1,1)     (1,2)      (1,3)     (2,1)     (2,2)     (2,3)
## [1,] 0.09174312 0.1238532 0.08256881 0.1238532 0.1651376 0.1238532
##           (3,1)     (3,2)      (3,3)
## [1,] 0.08256881 0.1238532 0.08256881
```

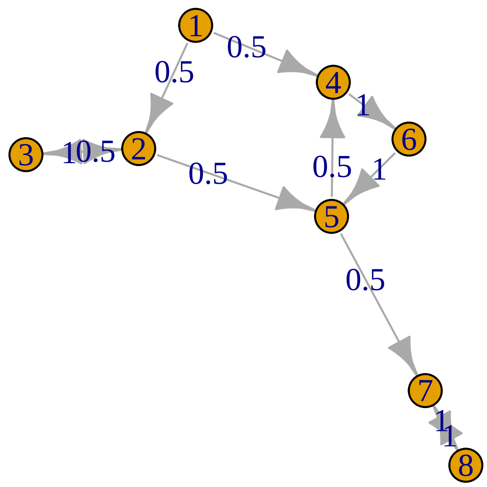
example 3

markov chain

```r
#states
states4 <- c("1","2","3","4","5","6","7","8")

#transition matrix

tmat4 <- matrix(data=c(0,.5,0,.5,0,0,0,0,
                       0,0,.5,0,.5,0,0,0,
                       0,1,0,0,0,0,0,0,
                       0,0,0,0,0,1,0,0,
                       0,0,0,.5,0,0,.5,0,
                       0,0,0,0,1,0,0,0,
                       0,0,0,0,0,0,0,1,
                       0,0,0,0,0,0,1,0), byrow=TRUE, nrow=8,
               dimnames=list(states4,states4))

tmat4
```

```
##   1   2   3   4   5 6   7 8
## 1 0 0.5 0.0 0.5 0.0 0 0.0 0
## 2 0 0.0 0.5 0.0 0.5 0 0.0 0
## 3 0 1.0 0.0 0.0 0.0 0 0.0 0
## 4 0 0.0 0.0 0.0 0.0 1 0.0 0
## 5 0 0.0 0.0 0.5 0.0 0 0.5 0
## 6 0 0.0 0.0 0.0 1.0 0 0.0 0
## 7 0 0.0 0.0 0.0 0.0 0 0.0 1
## 8 0 0.0 0.0 0.0 0.0 0 1.0 0
```

```r
library("markovchain")
# Markov chain
mc4 <- new("markovchain", states = states4, byrow = TRUE, transitionMatrix = tmat4, name = "communicati

# transition diagram
plot(mc4)
```



a

when talking about states accesible from 5 we look for outgoing arrows. 4 is accesible from 5 6 is accessible from 5 through 4 7 is accesible from 5 8 is accesible from 5 through 7

b

when talking about which states lead to 5 we look for incoming arrows. state 2 leads to 5 state 1 and 3 through 2 lead to 5. state 1 also through states 4 and 6 leads to 5. state 6 leads to 5. state 4 through 6 leads to 5.

c

states 4 and 6 communicate with 5

d

we have 4 communication classes 1 2,3 4,5,6 7,8

```r
communicatingClasses(mc4)
```

```
## [[1]]
## [1] "1"
##
```

```
## [[2]]
## [1] "2" "3"
##
## [[3]]
## [1] "4" "5" "6"
##
## [[4]]
## [1] "7" "8"
```

e

transient classes "1" "2" "3"

"4" "5" "6"

Transient states "1" "2" "3" "4" "5" "6"

```r
transientClasses(mc4)
```

```
## [[1]]
## [1] "1"
##
## [[2]]
## [1] "2" "3"
##
## [[3]]
## [1] "4" "5" "6"
```

```r
transientStates(mc4)
```

```
## [1] "1" "2" "3" "4" "5" "6"
```

f

recurrent class "7" "8" recurrent states "7" "8"

```r
recurrentClasses(mc4)
```

```
## [[1]]
## [1] "7" "8"
```

```r
recurrentStates(mc4)
```

```
## [1] "7" "8"
```

g

If a recurrent class contains a single state, then this state is called absorbing state. A state i is called absorbing if it is impossible to leave this state. Therefore, the state i is absorbing if and only if P(i to i)=1 and P(i to j)=0. A Markov chain is an absorbing chain

there is at least one absorbing state and it is possible to go from any state to at least one absorbing state in a finite number of steps. In an absorbing Markov chain, a state that is not absorbing is called transient. In our markov chain there are no absorbing states.

h

A unichain is a Markov chain with a single recurrent class, and possibly some transient classes. yes our markov chain is a unichain as it has one recurring class "7" and "8".

i

```r
canonicForm(mc4)
```

```
## communicating classes
##  A  8 - dimensional discrete Markov Chain defined by the following states:
##  7, 8, 1, 2, 3, 4, 5, 6
##  The transition matrix  (by rows)  is defined as follows:
##     7 8 1   2   3   4   5 6
## 7 0.0 1 0 0.0 0.0 0.0 0.0 0
## 8 1.0 0 0 0.0 0.0 0.0 0.0 0
## 1 0.0 0 0 0.5 0.0 0.5 0.0 0
## 2 0.0 0 0 0.0 0.5 0.0 0.5 0
## 3 0.0 0 0 1.0 0.0 0.0 0.0 0
## 4 0.0 0 0 0.0 0.0 0.0 0.0 1
## 5 0.5 0 0 0.0 0.0 0.5 0.0 0
## 6 0.0 0 0 0.0 0.0 0.0 1.0 0
```