stochastic homework 3

kallakuri radhakrishna

11/06/2020

example 1

a

```
#states
states <- c("1","2","3","2c","3c","G","D")

#transition matrix

tmat <- matrix(data=c(.05,.7,0,.2,0,0,.05,
                      0,.05,.7,0,.2,0,.05,
                      0,0,.15,0,0,.8,.05,
                      0,.1,.6,0,.2,0,.1,
                      0,0,.2,0,0,.7,.1,
                      0,0,0,0,0,1,0,
                      0,0,0,0,0,0,1),byrow=TRUE, nrow=7,
               dimnames=list(states,states))

tmat
```

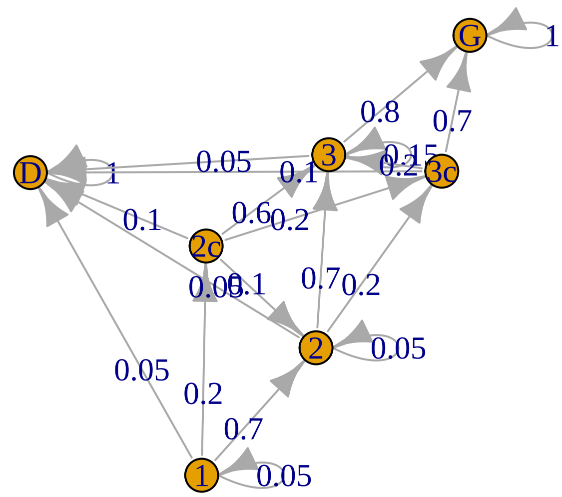
```
##       1    2    3  2c  3c   G    D
## 1  0.05 0.70 0.00 0.2 0.0 0.0 0.05
## 2  0.00 0.05 0.70 0.0 0.2 0.0 0.05
## 3  0.00 0.00 0.15 0.0 0.0 0.8 0.05
## 2c 0.00 0.10 0.60 0.0 0.2 0.0 0.10
## 3c 0.00 0.00 0.20 0.0 0.0 0.7 0.10
## G  0.00 0.00 0.00 0.0 0.0 1.0 0.00
## D  0.00 0.00 0.00 0.0 0.0 0.0 1.00
```

```
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 = "study program")

# transition diagram
plot(mc)
```



b

For a chain to be absorbing the following conditions have to be met. it contains at least one absorbing state,(A state i is absorbing if pii = 1) every non-absorbing state leads to an absorbing state.

There are two absorbing states G and D as thier pii=1. We can see that states 1,2,3,2c,3c all lead to D that is all non-absorbing states are leading to an absorbing state. That is why it is an absorbing chain.

c

the probability of student in first year to be regularly accepted into third year ahter two years of study is 0.61.

```
q0 <- c(1,0,0,0,0,0,0)
dist <- q0*(mc^2)
print(dist)
```

```
##           1    2    3   2c   3c G      D
## [1,] 0.0025 0.09 0.61 0.01 0.18 0 0.1075
```

```
#altenate method
# transitionafter two steps from  "1" -> "3"
(mc^2)["1","3"]
```

```
## [1] 0.61
```

d

canonical form

```r
cf <-canonicForm(mc)
cf
```

```
## study program
##  A  7 - dimensional discrete Markov Chain defined by the following states:
##  G, D, 1, 2, 3, 2c, 3c
##  The transition matrix  (by rows)  is defined as follows:
##      G    D    1    2    3  2c  3c
## G  1.0 0.00 0.00 0.00 0.00 0.0 0.0
## D  0.0 1.00 0.00 0.00 0.00 0.0 0.0
## 1  0.0 0.05 0.05 0.70 0.00 0.2 0.0
## 2  0.0 0.05 0.00 0.05 0.70 0.0 0.2
## 3  0.8 0.05 0.00 0.00 0.15 0.0 0.0
## 2c 0.0 0.10 0.00 0.10 0.60 0.0 0.2
## 3c 0.7 0.10 0.00 0.00 0.20 0.0 0.0
```

e

fundamental matrix

```r
# list all transient states
ts <- transientStates(cf)
# list the absorbing states
as <- absorbingStates(cf)
ts
```

```
## [1] "1"  "2"  "3"  "2c" "3c"
```

```r
as
```

```
## [1] "G" "D"
```

```r
# matrices Q and S
Q <- cf[ts, ts]
S <- cf[ts, as]
Q
```

```
##       1    2    3  2c  3c
## 1  0.05 0.70 0.00 0.2 0.0
## 2  0.00 0.05 0.70 0.0 0.2
## 3  0.00 0.00 0.15 0.0 0.0
## 2c 0.00 0.10 0.60 0.0 0.2
## 3c 0.00 0.00 0.20 0.0 0.0
```

```r
S
```

```
##      G    D
## 1  0.0 0.05
## 2  0.0 0.05
## 3  0.8 0.05
## 2c 0.0 0.10
## 3c 0.7 0.10
```

```r
#calculation of fundamental matrix
fund <- solve(diag(length(ts))-Q)
```

```r
fund
```

```
##           1         2         3        2c        3c
## 1  1.052632 0.7977839 0.8530552 0.2105263 0.2016620
## 2  0.000000 1.0526316 0.9164087 0.0000000 0.2105263
## 3  0.000000 0.0000000 1.1764706 0.0000000 0.0000000
## 2c 0.000000 0.1052632 0.8445820 1.0000000 0.2210526
## 3c 0.000000 0.0000000 0.2352941 0.0000000 1.0000000
```

f

f1

a time student spends in the process 1 3.115659 2 2.179567 3 1.176471 2c 2.170898 3c 1.235294

```r
# time to absorption
rowSums(fund)
```

```
##        1        2        3       2c       3c
## 3.115659 2.179567 1.176471 2.170898 1.235294
```

```r
# time to absorption
fund %*% rep(1, length(ts))
```

```
##        [,1]
## 1  3.115659
## 2  2.179567
## 3  1.176471
## 2c 2.170898
## 3c 1.235294
```

f2

The probability that student enrolled in frist year eventually graduates is 0.82

f3

a student accepted into second year conditionally will drop out with probability 0.169.

```r
# probabilities of absorption
fund %*% S
```

```
##            G          D
## 1  0.8236076 0.17639237
## 2  0.8804954 0.11950464
## 3  0.9411765 0.05882353
## 2c 0.8304025 0.16959752
## 3c 0.8882353 0.11176471
```

f4

expeceted number of years a student enrolled in 2nd year will spend in 3rd year regularly or conditionally is $0.916 + 0.21 = 1.126$

```
fund
```

```
##           1         2         3        2c        3c
## 1  1.052632 0.7977839 0.8530552 0.2105263 0.2016620
## 2  0.000000 1.0526316 0.9164087 0.0000000 0.2105263
## 3  0.000000 0.0000000 1.1764706 0.0000000 0.0000000
## 2c 0.000000 0.1052632 0.8445820 1.0000000 0.2210526
## 3c 0.000000 0.0000000 0.2352941 0.0000000 1.0000000
```

example 2

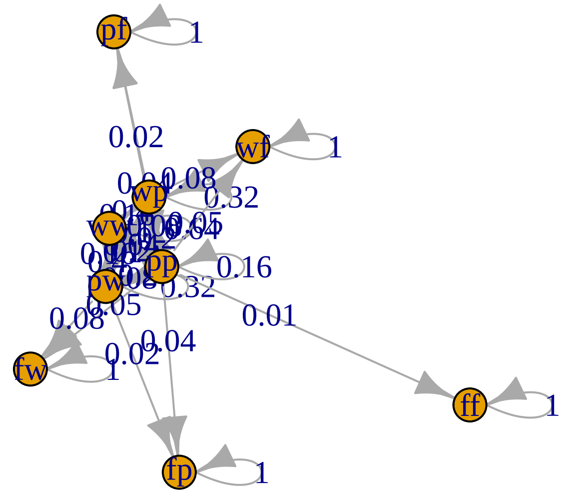
a

markov chain for machine

```
#states

states1 <- c("ww","wp","wf","pw","pp","pf","fw","fp","ff")

#transition matrix

tmat1 <- matrix(data=c(.64,.16,0,.16,.04,0,0,0,0,
                        .4,.32,.08,.10,.08,.02,0,0,0,
                       0,0,1,0,0,0,0,0,0,
                       .4,.1,0,.32,.08,0,.08,.02,0,
                       .25,.2,.05,.2,.16,.04,.05,.04,.01,
                       0,0,0,0,0,1,0,0,0,
                       0,0,0,0,0,0,1,0,0,
                       0,0,0,0,0,0,0,1,0,
                       0,0,0,0,0,0,0,0,1),byrow=TRUE, nrow=9,
               dimnames=list(states1,states1))

tmat1
```

```
##      ww   wp   wf   pw   pp   pf   fw   fp   ff
## ww 0.64 0.16 0.00 0.16 0.04 0.00 0.00 0.00 0.00
## wp 0.40 0.32 0.08 0.10 0.08 0.02 0.00 0.00 0.00
## wf 0.00 0.00 1.00 0.00 0.00 0.00 0.00 0.00 0.00
## pw 0.40 0.10 0.00 0.32 0.08 0.00 0.08 0.02 0.00
## pp 0.25 0.20 0.05 0.20 0.16 0.04 0.05 0.04 0.01
## pf 0.00 0.00 0.00 0.00 0.00 1.00 0.00 0.00 0.00
## fw 0.00 0.00 0.00 0.00 0.00 0.00 1.00 0.00 0.00
## fp 0.00 0.00 0.00 0.00 0.00 0.00 0.00 1.00 0.00
## ff 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 1.00
```

```
library("markovchain")
# Markov chain
mc1 <- new("markovchain", states = states1, byrow = TRUE, transitionMatrix = tmat1, name = "machine")

# transition diagram
plot(mc1)
```



b

absorbing states "wf" "pf" "fw" "fp" "ff"

```
As1 <- absorbingStates(mc1)
As1
```

```
## [1] "wf" "pf" "fw" "fp" "ff"
```

c

probability that machine stops operating in 5 weeks or less if currently both are working for ww taking all prob of 2,3,4,5 steps for pf,fw,fp,ff and adding them 0.0048+0.0148+0.0048+0.0004+ 0.010656+0.031888+0.010656+0.000976+ 0.01651200+ 0.04876304 +0.01651200 +0.00157136+ 0.02213568+ 0.06492176 +0.02213568+ 0.00214736 = 0.27

```
mc1^2
```

```
## machine^2
##  A  9 - dimensional discrete Markov Chain defined by the following states:
##  ww, wp, wf, pw, pp, pf, fw, fp, ff
##  The transition matrix  (by rows)  is defined as follows:
##        ww     wp     wf     pw     pp     pf     fw     fp     ff
## ww 0.5476 0.1776 0.0148 0.1776 0.0576 0.0048 0.0148 0.0048 0.0004
## wp 0.4440 0.1924 0.1096 0.1440 0.0624 0.0296 0.0120 0.0052 0.0008
## wf 0.0000 0.0000 1.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
```

```
## pw 0.4440 0.1440 0.0120 0.1924 0.0624 0.0052 0.1096 0.0296 0.0008
## pp 0.3600 0.1560 0.0740 0.1560 0.0676 0.0504 0.0740 0.0504 0.0116
## pf 0.0000 0.0000 0.0000 0.0000 0.0000 1.0000 0.0000 0.0000 0.0000
## fw 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 1.0000 0.0000 0.0000
## fp 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 1.0000 0.0000
## ff 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 1.0000
```

```
mc1^3
```

```
## machine^3
##  A  9 - dimensional discrete Markov Chain defined by the following states:
##  ww, wp, wf, pw, pp, pf, fw, fp, ff
##  The transition matrix  (by rows)  is defined as follows:
##          ww       wp       wf       pw       pp       pf       fw       fp
## ww 0.506944 0.173728 0.031888 0.173728 0.059536 0.010656 0.031888 0.010656
## wp 0.434320 0.159488 0.128112 0.148840 0.054656 0.035944 0.026640 0.010576
## wf 0.000000 0.000000 1.000000 0.000000 0.000000 0.000000 0.000000 0.000000
## pw 0.434320 0.148840 0.026640 0.159488 0.054656 0.010576 0.128112 0.035944
## pp 0.372100 0.136640 0.089860 0.136640 0.050176 0.056224 0.089860 0.056224
## pf 0.000000 0.000000 0.000000 0.000000 0.000000 1.000000 0.000000 0.000000
## fw 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 1.000000 0.000000
## fp 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 1.000000
## ff 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
##          ff
## ww 0.000976
## wp 0.001424
## wf 0.000000
## pw 0.001424
## pp 0.012276
## pf 0.000000
## fw 0.000000
## fp 0.000000
## ff 1.000000
```

```
mc1^4
```

```
## machine^4
##  A  9 - dimensional discrete Markov Chain defined by the following states:
##  ww, wp, wf, pw, pp, pf, fw, fp, ff
##  The transition matrix  (by rows)  is defined as follows:
##           ww        wp         wf        pw         pp         pf
## ww 0.4783106 0.1659840 0.04876304 0.1659840 0.05760000 0.01651200
## wp 0.4149600 0.1463426 0.14360384 0.1440000 0.05078400 0.04132000
## wf 0.0000000 0.0000000 1.00000000 0.0000000 0.00000000 0.00000000
## pw 0.4149600 0.1440000 0.04128000 0.1463426 0.05078400 0.01573904
## pp 0.3600000 0.1269600 0.10330000 0.1269600 0.04477456 0.06096384
## pf 0.0000000 0.0000000 0.00000000 0.0000000 0.00000000 1.00000000
## fw 0.0000000 0.0000000 0.00000000 0.0000000 0.00000000 0.00000000
## fp 0.0000000 0.0000000 0.00000000 0.0000000 0.00000000 0.00000000
## ff 0.0000000 0.0000000 0.00000000 0.0000000 0.00000000 0.00000000
##            fw         fp         ff
## ww 0.04876304 0.01651200 0.00157136
## wp 0.04128000 0.01573904 0.00197056
## wf 0.00000000 0.00000000 0.00000000
## pw 0.14360384 0.04132000 0.00197056
## pp 0.10330000 0.06096384 0.01277776
```

```
## pf 0.00000000 0.00000000 0.00000000
## fw 1.00000000 0.00000000 0.00000000
## fp 0.00000000 1.00000000 0.00000000
## ff 0.00000000 0.00000000 1.00000000
```

```r
mc1^5
```

```
## machine^5
##  A  9 - dimensional discrete Markov Chain defined by the following states:
##  ww, wp, wf, pw, pp, pf, fw, fp, ff
##  The transition matrix  (by rows)  is defined as follows:
##           ww        wp         wf        pw         pp         pf
## ww 0.4533060 0.1577630 0.06492176 0.1577630 0.05490586 0.02213568
## wp 0.3944074 0.1377800 0.15785044 0.1372647 0.04795124 0.04627821
## wf 0.0000000 0.0000000 1.00000000 0.0000000 0.00000000 0.00000000
## pw 0.3944074 0.1372647 0.05533920 0.1377800 0.04795124 0.02065040
## pp 0.3431616 0.1198781 0.11569553 0.1198781 0.04187753 0.06529402
## pf 0.0000000 0.0000000 0.00000000 0.0000000 0.00000000 1.00000000
## fw 0.0000000 0.0000000 0.00000000 0.0000000 0.00000000 0.00000000
## fp 0.0000000 0.0000000 0.00000000 0.0000000 0.00000000 0.00000000
## ff 0.0000000 0.0000000 0.00000000 0.0000000 0.00000000 0.00000000
##            fw         fp         ff
## ww 0.06492176 0.02213568 0.00214736
## wp 0.05533920 0.02065040 0.00247840
## wf 0.00000000 0.00000000 0.00000000
## pw 0.15785044 0.04627821 0.00247840
## pp 0.11569553 0.06529402 0.01322551
## pf 0.00000000 0.00000000 0.00000000
## fw 1.00000000 0.00000000 0.00000000
## fp 0.00000000 1.00000000 0.00000000
## ff 0.00000000 0.00000000 1.00000000
```

d

```r
cf1 <-canonicForm(mc1)
cf1
```

```
## machine
##  A  9 - dimensional discrete Markov Chain defined by the following states:
##  wf, pf, fw, fp, ff, ww, wp, pw, pp
##  The transition matrix  (by rows)  is defined as follows:
##      wf   pf   fw   fp   ff   ww   wp   pw   pp
## wf 1.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
## pf 0.00 1.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
## fw 0.00 0.00 1.00 0.00 0.00 0.00 0.00 0.00 0.00
## fp 0.00 0.00 0.00 1.00 0.00 0.00 0.00 0.00 0.00
## ff 0.00 0.00 0.00 0.00 1.00 0.00 0.00 0.00 0.00
## ww 0.00 0.00 0.00 0.00 0.00 0.64 0.16 0.16 0.04
## wp 0.08 0.02 0.00 0.00 0.00 0.40 0.32 0.10 0.08
## pw 0.00 0.00 0.08 0.02 0.00 0.40 0.10 0.32 0.08
## pp 0.05 0.04 0.05 0.04 0.01 0.25 0.20 0.20 0.16
```

```r
# list all transient states
ts1 <- transientStates(cf1)
# list the absorbing states
```

```r
as1 <- absorbingStates(cf1)
ts1
```

```
## [1] "ww" "wp" "pw" "pp"
```

```r
as1
```

```
## [1] "wf" "pf" "fw" "fp" "ff"
```

```r
# matrices Q1 and S1
Q1 <- cf1[ts1, ts1]
S1 <- cf1[ts1, as1]
Q1
```

```
##      ww   wp   pw   pp
## ww 0.64 0.16 0.16 0.04
## wp 0.40 0.32 0.10 0.08
## pw 0.40 0.10 0.32 0.08
## pp 0.25 0.20 0.20 0.16
```

```r
S1
```

```
##      wf   pf   fw   fp   ff
## ww 0.00 0.00 0.00 0.00 0.00
## wp 0.08 0.02 0.00 0.00 0.00
## pw 0.00 0.00 0.08 0.02 0.00
## pp 0.05 0.04 0.05 0.04 0.01
```

```r
#calculation of fundamental matrix
fund1 <- solve(diag(length(ts1))-Q1)
fund1
```

```
##           ww       wp       pw       pp
## ww 12.057639 3.771986 3.771986 1.292647
## wp  9.429964 4.513668 3.231617 1.186692
## pw  9.429964 3.231617 4.513668 1.186692
## pp  8.079042 2.966730 2.966730 2.140284
```

d1

if currently both parts are working expected operating time is 20.89 weeks.

```r
# d1
rowSums(fund1)
```

```
##       ww       wp       pw       pp
## 20.89426 18.36194 18.36194 16.15279
```

d2

to find the prob of machine stops while one of the machine is working that means other is in partial working is to find rowsums of S1 matrix to finds row sums of wp and pw p = 0.1+0.1 =0.2 the other method is to use probability take two cases by fixing one part is working then other part is aprtially working it will fail with probabiltiy as 0.1 as anyone part can be working so we multiply 0.1 by 2=0.2

```r
rowSums(S1)
```

```
##   ww   wp   pw   pp
```

```
## 0.00 0.10 0.10 0.19
```

d3

by observing fundmental matrix, we select row of both partially working "pp" we see what is the value for both working in that row that is 8.079.

```r
fund1
```

```
##           ww       wp       pw       pp
## ww 12.057639 3.771986 3.771986 1.292647
## wp  9.429964 4.513668 3.231617 1.186692
## pw  9.429964 3.231617 4.513668 1.186692
## pp  8.079042 2.966730 2.966730 2.140284
```

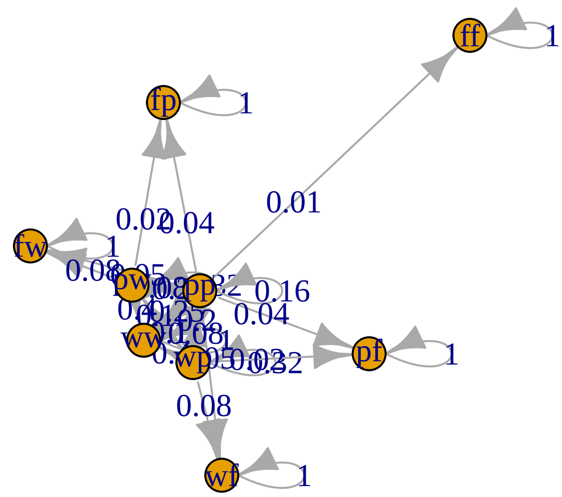
d4

the probability that both parts become working again before the machine stops operating if currently one part is working and the other is partially working is 0.782.

```r
states1 <- c("ww","wp","wf","pw","pp","pf","fw","fp","ff")

#transition matrix

tmat2 <- matrix(data=c(1,0,0,0,0,0,0,0,0,
                       .4,.32,.08,.10,.08,.02,0,0,0,
                       0,0,1,0,0,0,0,0,0,
                       .4,.1,0,.32,.08,0,.08,.02,0,
                       .25,.2,.05,.2,.16,.04,.05,.04,.01,
                       0,0,0,0,0,1,0,0,0,
                       0,0,0,0,0,0,1,0,0,
                       0,0,0,0,0,0,0,1,0,
                       0,0,0,0,0,0,0,0,1),byrow=TRUE, nrow=9,
                dimnames=list(states1,states1))
tmat2
```

```
##      ww   wp   wf   pw   pp   pf   fw   fp   ff
## ww 1.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
## wp 0.40 0.32 0.08 0.10 0.08 0.02 0.00 0.00 0.00
## wf 0.00 0.00 1.00 0.00 0.00 0.00 0.00 0.00 0.00
## pw 0.40 0.10 0.00 0.32 0.08 0.00 0.08 0.02 0.00
## pp 0.25 0.20 0.05 0.20 0.16 0.04 0.05 0.04 0.01
## pf 0.00 0.00 0.00 0.00 0.00 1.00 0.00 0.00 0.00
## fw 0.00 0.00 0.00 0.00 0.00 0.00 1.00 0.00 0.00
## fp 0.00 0.00 0.00 0.00 0.00 0.00 0.00 1.00 0.00
## ff 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 1.00
```

```r
library("markovchain")
# Markov chain
mc2 <- new("markovchain", states = states1, byrow = TRUE, transitionMatrix = tmat2, name = "machine")

# transition diagram
plot(mc2)
```




```
absorbingStates(mc2)
```

```
## [1] "ww" "wf" "pf" "fw" "fp" "ff"
```

```
absorptionProbabilities(mc2)
```

```
##           ww         wf         pf         fw         fp          ff
## wp 0.7820738 0.13388311 0.03830382 0.03131900 0.01266279 0.001757469
## pw 0.7820738 0.03131900 0.01266279 0.13388311 0.03830382 0.001757469
## pp 0.6700351 0.09885764 0.05975395 0.09885764 0.05975395 0.012741652
```

example 3

a

Mean recurrence time of all states (1,1) (1,2) (1,3) (2,1) (2,2) (2,3) (3,1) (3,2) (3,3) [1,] 12 8 12 8 6 8 12 8 12

```
# adjacency matrix



states2 <- c("(1,1)","(1,2)","(1,3)","(2,1)","(2,2)","(2,3)","(3,1)","(3,2)","(3,3)")
states2
```

```
## [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,
```

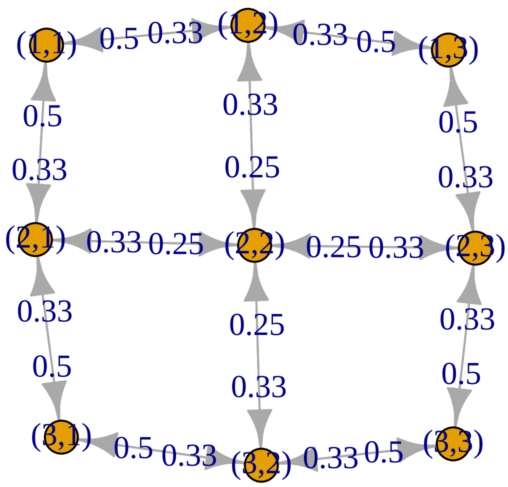
```
                            0,1,0,0,0,1,0,0,0,
                            1,0,0,0,1,0,1,0,0,
                            0,1,0,1,0,1,0,1,0,
                            0,0,1,0,1,0,0,0,1,
                            0,0,0,1,0,0,0,1,0,
                            0,0,0,0,1,0,1,0,1,
                            0,0,0,0,0,1,0,1,0), byrow=TRUE, nrow=9,
                    dimnames=list(states2,states2))

# example 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
```

```
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
```

```
rwgraph <- new("markovchain", states = states2, byrow = TRUE,
               transitionMatrix = trmat, name = "Random walk")
plot(rwgraph)
```




```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
```

```r
# mean time to recurrence
rt <- 1/invariant
show(rt)
```

```
##      (1,1) (1,2) (1,3) (2,1) (2,2) (2,3) (3,1) (3,2) (3,3)
## [1,]    12     8    12     8     6     8    12     8    12
```

```r
# alternate method
# mean recurrence times
meanRecurrenceTime(rwgraph)
```

```
## (1,1) (1,2) (1,3) (2,1) (2,2) (2,3) (3,1) (3,2) (3,3)
##    12     8    12     8     6     8    12     8    12
```

b

Calculate the matrix of mean passage times between states. The expected time to reach state (3, 3) from (1, 1) is 18.

```r
# the W matrix - all rows contain pi
Wmat <- matrix(rep(invariant, 9), nrow = 9, byrow = T)
show(Wmat)
```

```
##             [,1]  [,2]       [,3]  [,4]      [,5]  [,6]       [,7]  [,8]
##  [1,] 0.08333333 0.125 0.08333333 0.125 0.1666667 0.125 0.08333333 0.125
##  [2,] 0.08333333 0.125 0.08333333 0.125 0.1666667 0.125 0.08333333 0.125
##  [3,] 0.08333333 0.125 0.08333333 0.125 0.1666667 0.125 0.08333333 0.125
##  [4,] 0.08333333 0.125 0.08333333 0.125 0.1666667 0.125 0.08333333 0.125
##  [5,] 0.08333333 0.125 0.08333333 0.125 0.1666667 0.125 0.08333333 0.125
##  [6,] 0.08333333 0.125 0.08333333 0.125 0.1666667 0.125 0.08333333 0.125
##  [7,] 0.08333333 0.125 0.08333333 0.125 0.1666667 0.125 0.08333333 0.125
##  [8,] 0.08333333 0.125 0.08333333 0.125 0.1666667 0.125 0.08333333 0.125
##  [9,] 0.08333333 0.125 0.08333333 0.125 0.1666667 0.125 0.08333333 0.125
##             [,9]
##  [1,] 0.08333333
##  [2,] 0.08333333
##  [3,] 0.08333333
##  [4,] 0.08333333
##  [5,] 0.08333333
##  [6,] 0.08333333
##  [7,] 0.08333333
##  [8,] 0.08333333
##  [9,] 0.08333333
```

```r
#fundamental matrix
Zmat <- solve(diag(9) - trmat + Wmat)
show(Zmat)
```

```
##              [,1]    [,2]        [,3]    [,4]        [,5]    [,6]
## (1,1)  1.20833333  0.4375 -0.04166667  0.4375 -0.08333333 -0.3125
## (1,2)  0.29166667  1.1875  0.29166667 -0.0625  0.08333333 -0.0625
## (1,3) -0.04166667  0.4375  1.20833333 -0.3125 -0.08333333  0.4375
## (2,1)  0.29166667 -0.0625 -0.20833333  1.1875  0.08333333 -0.3125
## (2,2) -0.04166667  0.0625 -0.04166667  0.0625  0.91666667  0.0625
## (2,3) -0.20833333 -0.0625  0.29166667 -0.3125  0.08333333  1.1875
## (3,1) -0.04166667 -0.3125 -0.29166667  0.4375 -0.08333333 -0.3125
## (3,2) -0.20833333 -0.3125 -0.20833333 -0.0625  0.08333333 -0.0625
## (3,3) -0.29166667 -0.3125 -0.04166667 -0.3125 -0.08333333  0.4375
##              [,7]    [,8]        [,9]
## (1,1) -0.04166667 -0.3125 -0.29166667
## (1,2) -0.20833333 -0.3125 -0.20833333
## (1,3) -0.29166667 -0.3125 -0.04166667
## (2,1)  0.29166667 -0.0625 -0.20833333
## (2,2) -0.04166667  0.0625 -0.04166667
## (2,3) -0.20833333 -0.0625  0.29166667
## (3,1)  1.20833333  0.4375 -0.04166667
## (3,2)  0.29166667  1.1875  0.29166667
## (3,3) -0.04166667  0.4375  1.20833333
```

```r
#mean first passage matrix
# the diagonal elements
dia <- rowSums(diag(9)*Zmat)
show(dia)
```

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

```
## 1.2083333 1.1875000 1.2083333 1.1875000 0.9166667 1.1875000 1.2083333
##     (3,2)     (3,3)
## 1.1875000 1.2083333
```

```r
# the matrix of zjj
Zjj <- matrix(rep(dia, 9), nrow = 9, byrow = T)
show(Zjj)
```

```
##           [,1]   [,2]     [,3]   [,4]      [,5]   [,6]     [,7]   [,8]
##  [1,] 1.208333 1.1875 1.208333 1.1875 0.9166667 1.1875 1.208333 1.1875
##  [2,] 1.208333 1.1875 1.208333 1.1875 0.9166667 1.1875 1.208333 1.1875
##  [3,] 1.208333 1.1875 1.208333 1.1875 0.9166667 1.1875 1.208333 1.1875
##  [4,] 1.208333 1.1875 1.208333 1.1875 0.9166667 1.1875 1.208333 1.1875
##  [5,] 1.208333 1.1875 1.208333 1.1875 0.9166667 1.1875 1.208333 1.1875
##  [6,] 1.208333 1.1875 1.208333 1.1875 0.9166667 1.1875 1.208333 1.1875
##  [7,] 1.208333 1.1875 1.208333 1.1875 0.9166667 1.1875 1.208333 1.1875
##  [8,] 1.208333 1.1875 1.208333 1.1875 0.9166667 1.1875 1.208333 1.1875
##  [9,] 1.208333 1.1875 1.208333 1.1875 0.9166667 1.1875 1.208333 1.1875
##           [,9]
##  [1,] 1.208333
##  [2,] 1.208333
##  [3,] 1.208333
##  [4,] 1.208333
##  [5,] 1.208333
##  [6,] 1.208333
##  [7,] 1.208333
##  [8,] 1.208333
##  [9,] 1.208333
```

```r
# the mean passage matrix
Mmat <- (Zjj-Zmat)/Wmat
show(Mmat)
```

```
##       [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9]
## (1,1)    0    6   15    6    6   12   15   12   18
## (1,2)   11    0   11   10    5   10   17   12   17
## (1,3)   15    6    0   12    6    6   18   12   15
## (2,1)   11   10   17    0    5   12   11   10   17
## (2,2)   15    9   15    9    0    9   15    9   15
## (2,3)   17   10   11   12    5    0   17   10   11
## (3,1)   15   12   18    6    6   12    0    6   15
## (3,2)   17   12   17   10    5   10   11    0   11
## (3,3)   18   12   15   12    6    6   15    6    0
```

```r
# the expecetd time to reach state (3,3) from (1,1) is 18.

# alternate method
#mean passage times matrix
meanFirstPassageTime(rwgraph)
```

```
##       (1,1) (1,2) (1,3) (2,1) (2,2) (2,3) (3,1) (3,2) (3,3)
## (1,1)     0     6    15     6     6    12    15    12    18
## (1,2)    11     0    11    10     5    10    17    12    17
## (1,3)    15     6     0    12     6     6    18    12    15
## (2,1)    11    10    17     0     5    12    11    10    17
## (2,2)    15     9    15     9     0     9    15     9    15
```

```
## (2,3)    17    10    11    12     5     0    17    10    11
## (3,1)    15    12    18     6     6    12     0     6    15
## (3,2)    17    12    17    10     5    10    11     0    11
## (3,3)    18    12    15    12     6     6    15     6     0
```

c

probability that starting from (1, 1), the chain will hit (3, 3) before (2, 2) is 0.142.

```r
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
# make states (2,2) and (3,3) absorbing
tmabs <- trmat
tmabs[5, ] <- c(0,0,0,0,1,0,0,0,0)
tmabs[9, ] <- c(0,0,0,0,0,0,0,0,1)
tmabs
```

```
##          (1,1) (1,2)     (1,3) (2,1)     (2,2) (2,3)     (3,1) (3,2)
##  [1,] 0.0000000   0.5 0.0000000   0.5 0.0000000   0.0 0.0000000   0.0
##  [2,] 0.3333333   0.0 0.3333333   0.0 0.3333333   0.0 0.0000000   0.0
##  [3,] 0.0000000   0.5 0.0000000   0.0 0.0000000   0.5 0.0000000   0.0
##  [4,] 0.3333333   0.0 0.0000000   0.0 0.3333333   0.0 0.3333333   0.0
##  [5,] 0.0000000   0.0 0.0000000   0.0 1.0000000   0.0 0.0000000   0.0
##  [6,] 0.0000000   0.0 0.3333333   0.0 0.3333333   0.0 0.0000000   0.0
##  [7,] 0.0000000   0.0 0.0000000   0.5 0.0000000   0.0 0.0000000   0.5
##  [8,] 0.0000000   0.0 0.0000000   0.0 0.3333333   0.0 0.3333333   0.0
##  [9,] 0.0000000   0.0 0.0000000   0.0 0.0000000   0.0 0.0000000   0.0
##          (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,] 1.0000000
```

```
mcabs <- new("markovchain", states = states2, byrow = TRUE, transitionMatrix = tmabs, name = "absorbing

# absorption probabilities
absorptionProbabilities(mcabs)
```

```
##           (2,2)     (3,3)
## (1,1) 0.8571429 0.1428571
## (1,2) 0.8571429 0.1428571
## (1,3) 0.7142857 0.2857143
## (2,1) 0.8571429 0.1428571
## (2,3) 0.5714286 0.4285714
## (3,1) 0.7142857 0.2857143
## (3,2) 0.5714286 0.4285714
```