STAT220 Oblig 4-5

Sigbjørn Fjelland

10/7/2020

Library in use:

```r
library(tinytex)
library(matrixcalc)
library(shape)
library(diagram)
library(igraph)
```

```
##
## Attaching package: 'igraph'

## The following object is masked from 'package:matrixcalc':
##
##     %s%

## The following objects are masked from 'package:stats':
##
##     decompose, spectrum

## The following object is masked from 'package:base':
##
##     union
```

```r
library(devtools)
```

```
## Loading required package: usethis
```

```r
library(expm)
```

```
## Loading required package: Matrix

##
## Attaching package: 'expm'

## The following object is masked from 'package:Matrix':
##
##     expm
```

Problem 6.1

Vi har gitt transition matrix P:

```
P=matrix(c(0.7, 0.2, 0.1, 0.0, 0.6, 0.4, 0.5, 0.0, 0.5), nrow=3, ncol=3, byrow = TRUE)
dimnames(P)=list(c('0','1','2'), c('0','1','2'))
print(P)
```

```
##     0   1   2
## 0 0.7 0.2 0.1
## 1 0.0 0.6 0.4
## 2 0.5 0.0 0.5
```

a) Since $P_{i,j}^{m} = P(x_{m+n} = j|x_n)$ and $P^{(m)} = P^m$:

   $\Rightarrow P^{(3)} = P^3$ og vi får følgende matrise

```
P_3 = matrix.power(P,3)
print(P_3)
```

```
##       0     1     2
## 0 0.478 0.264 0.258
## 1 0.360 0.256 0.384
## 2 0.570 0.180 0.250
```

```
print(P_3[1,2])
```

```
## [1] 0.264
```

slik at: $P(x_3 = 1|x_0 = 0) = \underline{\underline{0.264}}$

samme egenskap gjelder også for $P^{(3)} = P^3$

```
P_4 = matrix.power(P,4)
print(P_4)
```

```
##        0      1      2
## 0 0.4636 0.2540 0.2824
## 1 0.4440 0.2256 0.3304
## 2 0.5240 0.2220 0.2540
```

```
print(P_4[1,2])
```

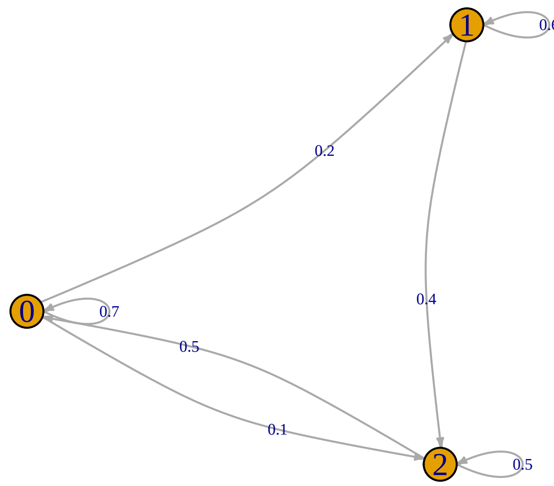
```
## [1] 0.254
```

og vi får da:

$P(x_3 = 1|x_0 = 0) = \underline{\underline{0.254}}$

b) As we se from the graphic presentation below, it is possible to go from any state to another trough one or more steps:

```
h <- graph_from_adjacency_matrix(P, weighted = "prob")
E(h)$prob <- ifelse(is.nan(E(h)$prob), NA, E(h)$prob)
plot(h, edge.label = round(E(h)$prob, 2), edge.arrow.size = .25, edge.curved=-0.2, edge.label.cex = .5)
```



from state 0: $0 \to 2 \to 1$

from state 1: $1 \to 0 \to 2$

from state 2:() $2 \to 1 \to 0$ and $1 \to 0$

It is therfore irreducible.

Periodicity: Since the period of a state is the largest d that satisfy following properties: - $p_{ii}^{(n)} = 0$ whenever n is not deviceble by d. - The period i is shown by $d(i)$. - If $P_{ii}^{(n)} = 0$, for all $n > 0 \to i = \infty$

```
and...
```

i is periodic if $d(i) > 1$ and aperiodic if $di = 1$

It is aperiodic since there are several sequences of steps to go from a state and back again ($i \to i$), icluding the fact that all the states are self periodic the markov chain is aperiodic.

Transient or recurrent? $\to$ It is recurrent! It might stop and loop at a position or between state 2 and 0 wich are comunicating, but eventually it will occure in any of the states by a certanty of 100% as in the formal definition: $f_{ii} = P(X_n = i, \text{for some } n \geq |w_0 = i)$ for any state i is Recurrent if $f_{ii} = 1$ and transient if Recurrent if $f_{ii} < 1$.

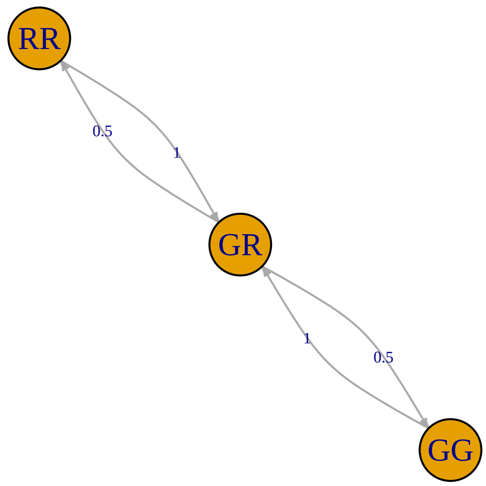
Problem 6.?

a) $G : Green\ R : Red$

We start initiatly at GR and pick a random ball with $p = 1/2$. We ar then sent to a state of RR or GG where the probability of being returned to State GR is 100%

```r
A=matrix(c(0, 1, 0, 0.5, 0, 0.5, 0, 1, 0), nrow=3, ncol=3, byrow = TRUE)
dimnames(A)=list(c('GG','GR','RR'), c('GG','GR','RR'))
```

```r
f <- graph_from_adjacency_matrix(A, weighted = "prob")
E(f)$prob <- ifelse(is.nan(E(f)$prob), NA, E(f)$prob)
plot(f, edge.label = round(E(f)$prob, 2),vertex.size=30, edge.arrow.size = .25, edge.curved=0.3, edge.la
```



Wich gives following matrix

```r
print(A)
```

```
##     GG GR  RR
## GG 0.0  1 0.0
## GR 0.5  0 0.5
## RR 0.0  1 0.0
```

and following for picking red ball in the future entering the state of $GR \rightarrow P(X_n = j|X_0 = 1)$: #her må jeg få sendt ett spørsmål

```r
print(A[2,])
```

```
##  GG  GR  RR
## 0.5 0.0 0.5
```

b) $d(GR) = 2$ og $d(GG) = d(RR) = 4$ som gir største felles nevner 2 og dermed er den periodisk med periode 2.

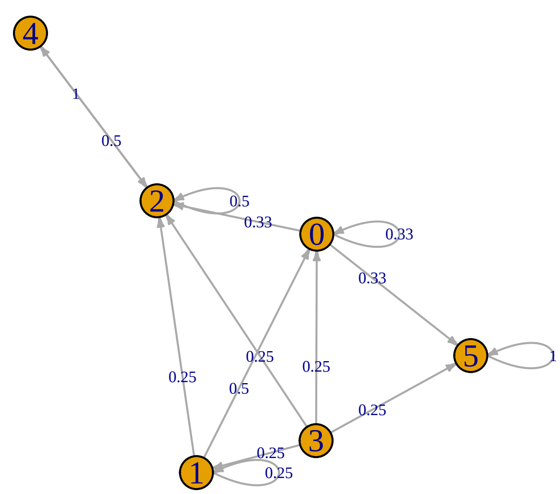
Problem 6.2

Due to previous use of P as variable, it is here substituted for B a)

```
B=round(matrix(c((1/3), 0.0, (1/3), 0.0, 0.0, (1/3),
           (1/2), (1/4), (1/4), 0.0, 0.0, 0.0,
           0.0, 0.0, (1/2), 0.0, (1/2), 0.0,
           (1/4), (1/4), (1/4), 0.0, 0.0, (1/4),
           0.0, 0.0, 1.0, 0.0, 0.0, 0.0,
           0.0, 0.0, 0.0, 0.0, 0.0, 1.0), nrow=6, ncol=6, byrow = TRUE),2)
dimnames(B)=list(c('0','1','2','3','4','5'), c('0','1','2','3','4','5'))
print(B)
```

```
##      0    1    2 3   4    5
## 0 0.33 0.00 0.33 0 0.0 0.33
## 1 0.50 0.25 0.25 0 0.0 0.00
## 2 0.00 0.00 0.50 0 0.5 0.00
## 3 0.25 0.25 0.25 0 0.0 0.25
## 4 0.00 0.00 1.00 0 0.0 0.00
## 5 0.00 0.00 0.00 0 0.0 1.00
```

```
g <- graph_from_adjacency_matrix(B, weighted = "prob")
E(g)$prob <- ifelse(is.nan(E(g)$prob), NA, E(g)$prob)
plot(g, edge.label = round(E(g)$prob, 2), edge.arrow.size = .25, edge.label.cex = .5)
```



b) As we se from the graph above:

We have three recurrent states, 2, 4 and 5, devided in two classes.
class 1 - State 5 is self-recurrent class 2 - State 2 and 4

Transient States:

State 0, 1, and 3 will at some pint possibly loop, but ultimatly it will end up in recurrent state class 1 or 2.

c) Communicating States:

$(2 \leftrightarrow 4)$, $(1 \leftrightarrow 1)$, $(2 \leftrightarrow 2)$ and $(5 \leftrightarrow 5)$

d) Matrix realocated to statespace = {1,3,0,2,4,5}: (Matrix denoted - C to separate matrixes in the code)

```
C=round(matrix(c((1/4), 0.0, (1/2), (1/4), 0.0, 0.0,
           (1/4), 0.0, (1/4), (1/4), 0.0, (1/4),
           0.0, 0.0, (1/3), (1/3), 0.0, (1/3),
           0.0, 0.0, 0.0, (1/2), (1/2), 0.0,
           0.0, 0.0, 0.0, 1.0, 0.0, 0.0,
           0.0, 0.0, 0.0, 0.0, 0.0, 1.0), nrow=6, ncol=6, byrow = TRUE),2)
dimnames(C)=list(c('1','3','0','2','4','5'), c('1','3','0','2','4','5'))

print(C)
```

```
##      1 3    0    2   4    5
## 1 0.25 0 0.50 0.25 0.0 0.00
## 3 0.25 0 0.25 0.25 0.0 0.25
## 0 0.00 0 0.33 0.33 0.0 0.33
## 2 0.00 0 0.00 0.50 0.5 0.00
## 4 0.00 0 0.00 1.00 0.0 0.00
## 5 0.00 0 0.00 0.00 0.0 1.00
```

e)

```
R<-round(matrix(c(C[1, 4:6],
C[2, 4:6],
C[3, 4:6]), nrow=3, ncol=3, byrow = TRUE), 2)
dimnames(R)=list(c('1','3','0'), c('2','4','5'))
print(R)
```

```
##      2 4    5
## 1 0.25 0 0.00
## 3 0.25 0 0.25
## 0 0.33 0 0.33
```

This gives $P_0(X_T = 5) = P(X_T = 5|X_{T-1} = 0) = 0.3333 = \frac{1}{3}$

and the expectation:
$E(X_T = 5|X_{T-1}) = 1 + \sum_{j \in S} E(T_P = j|X_{T-1} = 0) \cdot P(X_T = j|X_{T-1} = 0)$
$= 1 + \sum_{j \in S} m_{0j} \cdot P(X_T = j|X_{T-1} = 0)$

We are proceeding for state 5 such that $m_{j,0} = 1$, $m_{2,0} = 0$.

$E(X_T = 5|X_{T-1}) = 1 + m_{0,0} \cdot P(X_T = 0|X_{T-1} = 0)$
$+ m_{2,0} \cdot P(X_T = 2|X_{T-1} = 0) + m_{5,0} \cdot P(X_T = 5|X_{T-1} = 0)$
$= 1 + 1 \cdot \frac{1}{3} + 1 \cdot \frac{1}{3} + 0 \cdot \frac{1}{3}$
$= \underline{\underline{\frac{5}{3} \approx 1,667}}$

f) for p^{∞}:

```
P_infinity = round(matrix.power(B,1000000),2)
print(P_infinity)
```

```
##   0 1    2 3    4    5
## 0 0 0 0.33 0 0.16 0.49
## 1 0 0 0.44 0 0.22 0.33
## 2 0 0 0.67 0 0.33 0.00
## 3 0 0 0.36 0 0.18 0.46
## 4 0 0 0.67 0 0.33 0.00
## 5 0 0 0.00 0 0.00 1.00
```

```
C=matrix(c((1/4), 0.0, (1/2), (1/4), 0.0, 0.0,
           0.0, 0.0, 0.0, (1/2), (1/2), 0.0,
           0.0, 0.0, (1/3), (1/3), 0.0, (1/3),
           0.0, 0.0, 0.0, (1/2), (1/2), 0.0,
           0.0, 0.0, 0.0, 1.0, 0.0, 0.0,
           0.0, 0.0, 0.0, 0.0, 0.0, 1.0), nrow=6, ncol=6, byrow = TRUE)
#dimnames(B)=list(c('1','3','0','2','4','5'), c('1','3','0','2','4','5'))
```

We see the matrix converges in the recurrent states in class 1 and 2.

problem 6.3

```
mP=round(matrix( c(0.62, 0.00, 0.38, 0.0,
                   0.55, 0.00, 0.45, 0.00,
                   0.00, 0.24, 0.00, 0.76,
                   0.00, 0.19, 0.00, 0.81),nrow=4, ncol=4, byrow = TRUE),2)
dimnames(mP)=list(c('0','1','2','3'), c('0','1','2','3'))
nu<-t(mP[1,])
# The first row vector of mP
mI<-diag(rep(1,4))
# unitmatrix of order 4
s<-mI[,1]
# first unit vector, e_1, in R**4
msnu<-s%*%nu
mR<- mP-msnu
```

Vi har matrisen P:

```
print(mP)
```

```
##      0    1    2    3
## 0 0.62 0.00 0.38 0.00
## 1 0.55 0.00 0.45 0.00
## 2 0.00 0.24 0.00 0.76
## 3 0.00 0.19 0.00 0.81
```

Vektor ν:

```
print(nu)
```

```
##         0 1    2 3
## [1,] 0.62 0 0.38 0
```

Vektor s:

```
print(s)
```

```
## [1] 1 0 0 0
```

$s \otimes \nu$:

```
print(msnu)
```

```
##         0 1    2 3
## [1,] 0.62 0 0.38 0
## [2,] 0.00 0 0.00 0
## [3,] 0.00 0 0.00 0
## [4,] 0.00 0 0.00 0
```

matrix $R = P - s \otimes \nu$:

```
mR<-mP-msnu
print(mR)
```

```
##      0    1    2    3
## 0 0.00 0.00 0.00 0.00
## 1 0.55 0.00 0.45 0.00
## 2 0.00 0.24 0.00 0.76
## 3 0.00 0.19 0.00 0.81
```

b) We chose an $M > 1$ and set M = 666:

```r
M<-666
Gsv <- matrix(0,4,4)

for(n in 0:M){
  Gsv<-Gsv + matrix.power(mR,n)
  }

print(Gsv)
```

```
##   0        1         2        3
## 0 1 0.000000 0.0000000 0.000000
## 1 1 1.818182 0.8181818 3.272727
## 2 1 1.818182 1.8181818 7.272727
## 3 1 1.818182 0.8181818 8.535885
```

```r
pis<-nu%*%Gsv
wtpi<- pis/sum(pis)
```

$\tilde{\pi}^s$:

```r
print(pis)
```

```
##      0         1         2        3
## [1,] 1 0.6909091 0.6909091 2.763636
```

and the the normalised $\tilde{\pi}$:

```r
print(round(wtpi,4))
```

```
##           0      1      2      3
## [1,] 0.1943 0.1343 0.1343 0.5371
```

True distribution: $\overrightarrow{\boldsymbol{\pi}} \cdot P = \overrightarrow{\boldsymbol{\pi}} = [\pi_0, \pi_1, \pi_2, \pi_3]$

$$\Rightarrow \begin{cases} 0.62 \cdot \pi_0 + 0.55 \cdot \pi_1 + 0.00 \cdot \pi_2 + 0.00 \cdot \pi_3 = \pi_0 \\ 0.00 \cdot \pi_0 + 0.00 \cdot \pi_1 + 0.24 \cdot \pi_2 + 0.19 \cdot \pi_3 = \pi_1 \\ 0.38 \cdot \pi_0 + 0.45 \cdot \pi_1 + 0.00 \cdot \pi_2 + 0.00 \cdot \pi_3 = \pi_2 \\ 0.00 \cdot \pi_0 + 0.00 \cdot \pi_1 + 0.76 \cdot \pi_2 + 0.81 \cdot \pi_3 = \pi_3 \end{cases}$$

Simplyfied:

$\overrightarrow{\boldsymbol{\pi}} \cdot P = \overrightarrow{\boldsymbol{\pi}} = [\pi_0, \pi_1, \pi_2, \pi_3]$

$$\Rightarrow \begin{cases} 0.62 \cdot \pi_0 + 0.55 \cdot \pi_1 = \pi_0 \to \pi_0 = \frac{0.55 \cdot 0.19}{0.19 \cdot (0.55+0.76)+0.38 \cdot 0.76} = \underline{0.1943} \\ 0.24 \cdot \pi_2 + 0.19 \cdot \pi_3 = \pi_1 \to \pi_1 = \frac{0.38}{0.55} \cdot \pi_0 = \frac{0.38}{0.55} \cdot 0.1943 = \underline{0.1343} \\ 0.38 \cdot \pi_0 + 0.45 \cdot \pi_1 = \pi_2 \to \pi_2 = \frac{0.38}{0.55} \cdot \pi_0 = \frac{0.38}{0.55} \cdot 0.1943 = \underline{0.1343} \\ 0.76 \cdot \pi_2 + 0.81 \cdot \pi_3 = \pi_3 \to \pi_3 = \frac{1.52}{0.55} \cdot \pi_0 = \frac{1.52}{0.55} \cdot \ 0.1943 = \underline{0.5371} \end{cases}$$

$\overrightarrow{\boldsymbol{\pi}} = [0.1943, 0.1343, 0.1343, 0.5371]$

```r
vPi<-c(0.1943, 0.1343, 0.1343, 0.5371)

print(wtpi-vPi)
```

```
##                 0             1             2            3
## [1,] 4.628975e-05 -2.438163e-05 -2.438163e-05 2.473498e-06
```

Diffrence between true and estimated π_0 is quite marginal (within 5 decimals).

c)