Practice R Code – Assignment

**Question 1:**

eps <- 0.2

P <- TransMat(eps)

N <- 50

X0 <- 1

X <- MarkovChain(N, X0, P)

X is our Markov Chain but we want it to start from X0 = 1, so:

c(1, X)

This gives a vector that begins with 1 as required.

We now want to plot this Markov Chain to check that the code is sufficient and has worked:

plot(0:N, c(1, X),

type = "b",

col = "blue",

xlab = "Time Step, n",

ylim = c(-N, N),

ylab = bquote("Markov Chain, " ~ X[n]),

main = bquote("Markov Chain” ~ .(N) ~ "steps, transition matrix with epsilon =" ~ .(eps)))

We can keep ylim in the code but taking it out allows us to observe the jumps in the markov chain more closely:

A picture containing screenshot

Description automatically generatedA screenshot of a cell phone

Description automatically generated

**Question 2**

num\_visits <- table(factor(X, 1:6))

proportion\_time <- num\_visits/N

num\_visits produces a table that tells us how many times the markov chain enters a state

proportion\_time produces a table that tells us how long the markov chain stayed in a state

We can now illustrate this example in a graph. We compare this with the stationary distribution produced by ‘StatDist’. We compare using a bar chart:

stationary <- StatDist(P)

rbind(proportion\_time, stationary)

barplot(rbind(proportion\_time, stationary), beside = TRUE,

col = c("red","blue"),

xlab = "States",

ylab = "Overall Time Spent at Particular State",

main = bquote("Proportion of Time Spent in Each State for" ~ .(N) ~ "Steps”),

legend.text = c("Proportion of Time", "Stationary Distribution"))

For our N value of 50, we get the following graph:

A screenshot of a cell phone

Description automatically generated

Ergodic theorem tells us that when the number of steps N of an irreducible finite-state Markov chain is very large, then the proportion of time spent in each state tends to the stationary distribution. We want to observe this by inputting varied values of N.

eps <- 0.2

P <- TransMat(eps)

N <- 10000

X0 <- 1

X <- MarkovChain(N, X0, P)

num\_visits <- table(factor(X, 1:6))

C <- data.matrix(num\_visits)

proportion\_time <- C[,1]/N

stationary <- StatDist(P)

rbind(proportion\_time, stationary)

barplot(rbind(proportion\_time,stationary), beside = TRUE,

col = c("red","blue"),

xlab = "States",

ylim = c(0,0.5),

ylab = "Overall Time Spent at Particular State",

main = bquote("Proportion of Time Spent in Each State for" ~.(N)~ "Steps"),

legend.text = c("Proportion of Time", "Stationary Distribution")

)

We get:

A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generated A screenshot of a cell phone

Description automatically generated

So, we can conclude that increasing the value of n (steps of the markov chain) means that the proportion of time spent in each state reaches a stationary distribution.

**Question 3**

The return time is the n values for a specific number in the markov chain. i.e



So which n values land on a specific i. For this we use the ‘which’ command. i.e

‘which(X==3)’ gives a list of numbers as low as 1 and up to 50.

Then, the *expected* return time is defined as



Then, to get the expected return time, we take the difference between the numbers produced by the which command (so we can observe how frequently they occur) and take the mean of this vector. Altogether, we get the following code:

expected\_return <- function(vector, value){

visit <- which(vector == value)

return\_time <- diff(visit)

mean(return\_time)

}

This creates a function that produces the expected return time for a specific value and specific vector. We can extend this further to return the expected return time for all states for a vector:

expected\_return\_times <- sapply(1:6, expected\_return, vector = X)

expected\_return\_times

For an irreducible finite-state Markov chain, the stationary distribution ***π*** and the expected return times μi are related by πi=1/μi. To observe this, we can input the following code:

eps <- 0.2

P <- TransMat(eps)

N <- 50

X0 <- 1

X <- MarkovChain(N, X0, P)

expected\_return\_times <- sapply(1:6, expected\_return, vector = X)

barplot(rbind(1/expected\_return\_times,stationary), beside = TRUE,

col = c("red","blue"),

xlab = "States",

ylab = "Expected Return Times",

main = bquote("Expected Return Time vs Stationary Distribution for" ~.(N)~ "Steps"),

legend.text = c("1/µ", "Stationary Distribution")

)

Varying the value of n we observe the following:

A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generated

Hence, as n tends to a large number, 1/µ tends towards the stationary distribution.

**Question 4**

If epsilon is very small (10^-6) or 0.01 then you can’t jump from 5 to 6

See diagram drawn in practical