AFRICAN INSTITUTE FOR MATHEMATICAL SCIENCES
(AIMS RWANDA, KIGALI)

Name: Aubrey Undi Phiri
Course: STOCHASTIC PROCESSES

Assignment Number: 1
Date: February 1, 2025

QUESTION 1

A gambler starts with an initial stake of \$3 and plays until reaching \$8 or going bankrupt (\$0). At each play its either he goes up by \$ or comes down by a \$1.

Let X_n as the amount of money the gambler has after n plays. The possible states are:

$$S = \{0, 1, 2, 3, 4, 5, 6, 7, 8\}$$

where 0 and 8 are absorbing states, because we are sure that at \$0 or at \$8 he will stop playing, thus the probability is one.

a) One-Step Transition Matrix

Let $P_{i,j} = P(X_{n+1} = j \mid X_n = i)$ be the probability of transitioning from state i to state j, also considering that the gambler cannot play with \$0 or jump by more than \$1 in a single play, the transition probabilities are:

- $P_{i,i+1} = 0.6$ for $i = 1, 2, 3, 4, 5, 6, 7$
- $P_{i,i-1} = 0.4$ for $i = 1, 2, 3, 4, 5, 6, 7$
- $P_{i,i+2} = 0$ for $i = 1, 2, 3, 4, 5, 6, 7$
- $P_{i,i-2} = 0$ for $i = 1, 2, 3, 4, 5, 6, 7$
- $P_{0,0} = 1$ (absorbing state)
- $P_{8,8} = 1$ (absorbing state)

State Transition Diagram

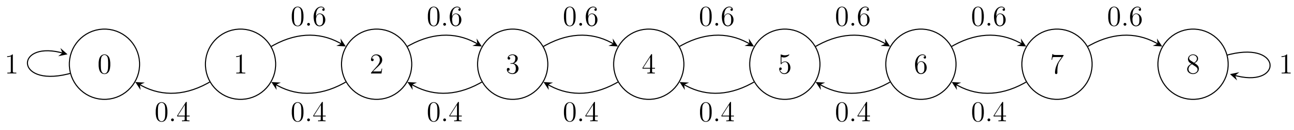

Figure 1: State Transition Diagram for the gambler

The one-step transition matrix P is:

$$P = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.4 & 0 & 0.6 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.4 & 0 & 0.6 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.4 & 0 & 0.6 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.4 & 0 & 0.6 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.4 & 0 & 0.6 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.4 & 0 & 0.6 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.4 & 0 & 0.6 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

b) Four-Step Transition Matrix

To find the four-step transition matrix, we compute:

$$P^{(4)} = P^4$$

which can be done using R Studio's matrix exponentiation or matrix multiplication.

```
# Load necessary package for matrix exponentiation
library(expm)
library(matrixcal)

# Define the transition matrix P
P <- matrix(c(
    1,   0,   0,   0,   0,   0,   0,   0,   0,
    0.4, 0,   0.6, 0,   0,   0,   0,   0,   0,
    0,   0.4, 0,   0.6, 0,   0,   0,   0,   0,
    0,   0,   0.4, 0,   0.6, 0,   0,   0,   0,
    0,   0,   0,   0.4, 0,   0.6, 0,   0,   0,
    0,   0,   0,   0,   0.4, 0,   0.6, 0,   0,
    0,   0,   0,   0,   0,   0.4, 0,   0.6, 0,
    0,   0,   0,   0,   0,   0,   0.4, 0,   0.6,
    0,   0,   0,   0,   0,   0,   0,   0,   1
), nrow = 9, byrow = TRUE)

# Compute P^4 using matrix exponentiation
```

```
P_4 <- P %^% 4

# Label the rows and columns from 0 to 8
colnames(P_4) <- 0:8
rownames(P_4) <- 0:8

# Print the result
print(P_4)
```

Output:

```
    > print(P_4)
       0      1      2      3      4      5      6      7      8
0 1.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
1 0.4960 0.1152 0.0000 0.2592 0.0000 0.1296 0.0000 0.0000 0.0000
2 0.2368 0.0000 0.2880 0.0000 0.3456 0.0000 0.1296 0.0000 0.0000
3 0.0640 0.1152 0.0000 0.3456 0.0000 0.3456 0.0000 0.1296 0.0000
4 0.0256 0.0000 0.1536 0.0000 0.3456 0.0000 0.3456 0.0000 0.1296
5 0.0000 0.0256 0.0000 0.1536 0.0000 0.3456 0.0000 0.2592 0.2160
6 0.0000 0.0000 0.0256 0.0000 0.1536 0.0000 0.2880 0.0000 0.5328
7 0.0000 0.0000 0.0000 0.0256 0.0000 0.1152 0.0000 0.1152 0.7440
8 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 1.0000
```

c) Expected Fortune After Four Plays

Let $E_i(4)$ denote the expected fortune of the gambler starting at state i after four plays. specifically, the expected fortune is $E(x_4|x_0 = 3)$ We compute:

$$E_i(4) = \sum_{j=0}^{8} jP_{i,j}^{(4)}$$

For $i = 3$ (initial stake), we sum over the states weighted by their four-step transition probabilities:

$$E_3(4) = \sum_{j=0}^{8} jP_{3,j}^{(4)}$$

where $P_{3,j}^{(4)}$ are the entries in the fourth row of P^4.

To find the expected fortune after four plays, we define:

The gambler starts with \$3:

$E_4(3)$= $(0 \times 0.0640) + (1 \times 0.1152) + (2 \times 0.0000) + (3 \times 0.3456) + (4 \times 0.0000) +$

$(5\times 0.3456) + (6 \times 0.0000) + (7 \times 0.1296) + (8 \times 0.0000)$

$$= 0 + 0.1152 + 0 + 1.0368 + 0 + 1.728 + 0 + 0.9072 + 0$$

$$= 3.7872$$

Thus, if the gambler starts with \$3, the expected amount after four plays is \$3.79.

Finding the Transition Matrix

We define the states:

- **Vowel (V)**
- **Consonant (C)**

From the given data:

$$\begin{aligned}
\text{Total vowel-initial pairs} &= 8,638 \\
\text{Vowel} \to \text{Vowel transitions} &= 1,104 \\
\text{Vowel} \to \text{Consonant transitions} &= 8,638 - 1,104 = 7,534 \\
\text{Total consonant-initial pairs} &= 11,362 \\
\text{Consonant} \to \text{Consonant transitions} &= 3,827 \\
\text{Consonant} \to \text{Vowel transitions} &= 11,362 - 3,827 = 7,535
\end{aligned}$$

We calculate the transition probabilities:

$$\begin{aligned}
P(V \to V) &= \frac{1104}{8638} \approx 0.1279, \quad P(V \to C) = \frac{7534}{8638} \approx 0.8721 \\
P(C \to C) &= \frac{3827}{11362} \approx 0.3369, \quad P(C \to V) = \frac{7535}{11362} \approx 0.6631
\end{aligned}$$

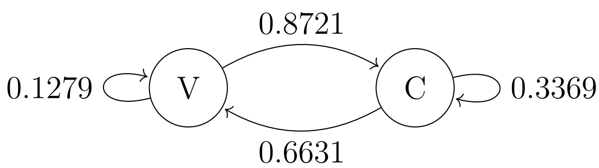

Figure 2: Transition Diagram for States Vowel and Consonant

Thus, the transition matrix is:

$$P = \begin{bmatrix} 0.1279 & 0.8721 \\ 0.6631 & 0.3369 \end{bmatrix}$$

Simulating the First 20 Letters

````
```{r}
# Load necessary libraries
library(ggplot2)
library(dplyr)

# Simulation settings
set.seed(123)
nTrials <- 1  # Number of trials
nSteps <- 20    # Number of steps
states <- c("V", "C")  # States
transition_matrix <- matrix(c(0.1279, 0.8721,
                              0.6631, 0.3369),
                            nrow = 2, byrow = TRUE)
# Function to simulate a single Markov chain trial
simulate_markov_chain <- function(nSteps, transition_matrix, trial) {
  current_state <- "V"
  sequence <- tibble(Step = 1, State = current_state, Trial = trial)
  for (i in 2:nSteps) {
    current_state <- if (current_state == "V") {
      sample(states, size = 1, prob = transition_matrix[1, ])
    } else {
      sample(states, size = 1, prob = transition_matrix[2, ])
    }
    sequence <- bind_rows(sequence, tibble(Step = i,
    State = current_state, Trial = trial))
  }
  sequence <- sequence %>% mutate(StateNum = ifelse(State == "V", 1, 2))
  return(sequence)
}
# Run multiple simulations
simulations <- bind_rows(lapply(1:nTrials, function(trial) {
  simulate_markov_chain(nSteps, transition_matrix, trial)
}))
# Plot the state transitions across trials as line graphs
ggplot(simulations, aes(x = Step, y = StateNum, group = Trial, color = State)) +
  geom_line(alpha = 0.5, size = 1) +  # Line graph with transparent lines
  scale_color_manual(values = c("V" = "blue", "C" = "red")) +
  scale_y_continuous(breaks = c(1, 2), labels = c("V", "C")) +
  theme_minimal() +
  labs(title = "Markov Chain State Transitions Across Multiple Trials",
       subtitle = "Each line represents a trial; Blue = 'V', Red = 'C'",
       x = "Time Step",
       y = "State") +
  theme(legend.position = "top")  # Move the legend to the top
````
````

The output below has a sequence that represents a possible realization of the Markov chain model of the poem starting with a vowel.
Output.

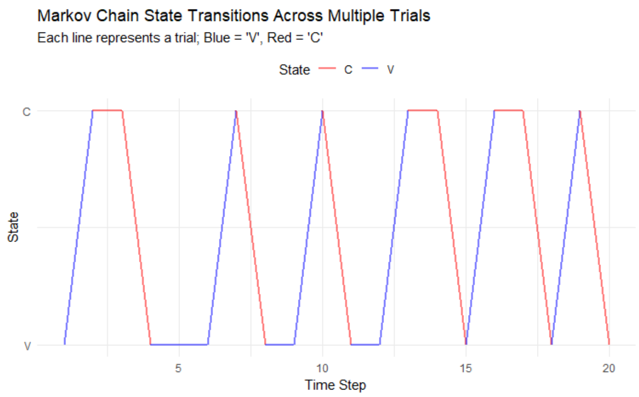

Figure 3: Simulation

The animation of the simulation,

```
# Load libraries
library(ggplot2)
library(gganimate)
library(dplyr)

# Set seed and define transition matrix
set.seed(123)
transition_matrix <- matrix(c(0.1279, 0.8721, 0.6631, 0.3369),
nrow = 2, byrow = TRUE)
states <- c("V", "C")

# Simulate Markov chain
n_steps <- 20
sequence <- tibble(Step = 1:n_steps, State = character(n_steps))
sequence$State[1] <- "V"
for (i in 2:n_steps) {
  sequence$State[i] <- sample(states, size = 1,
  prob = transition_matrix[sequence$State[i - 1] == states, ])
}
sequence <- sequence %>% mutate(StateNum = ifelse(State == "V", 1, 2))

# Plot and animate
p <- ggplot(sequence, aes(x = Step, y = StateNum, color = State, group = 1)) +
  geom_point(size = 5) +
  geom_line(color = "black", linewidth = 1) +
  # Explicitly set the group aesthetic
  scale_color_manual(values = c("V" = "blue", "C" = "red")) +
  scale_y_continuous(breaks = 1:2, labels = states) +
  theme_minimal() +
  labs(title = "Markov Chain Simulation of State Transitions",
       subtitle = "Step: {frame_time}",
       x = "Time Step",
       y = "State") +
  transition_reveal(Step)

# Save and show animation
animate(p, nframes = n_steps,
fps = 2, renderer = gifski_renderer("markov_simulation.gif"))
```

‘‘‘

Output of the animation, click the link below:

Link: https://youtu.be/GmxbrP3IoLs

Question 3

We are given a two-state Markov chain with state space $S = \{0, 1\}$ and transition matrix:

$$P = \begin{bmatrix} 1-p & p \\ q & 1-q \end{bmatrix}.$$

We define a new Markov chain on the bivariate process $Z_n = (X_{n-1}, X_n)$ with state space $S \times S = \{(0,0), (0,1), (1,0), (1,1)\}$.
The transition probabilities of the new chain are determined as follows:

- From $(0,0)$ to:
  - $(0,0)$: Probability $(1-p)$
  - $(0,1)$: Probability p
- From $(0,1)$ to:
  - $(1,0)$: Probability q
  - $(1,1)$: Probability $(1-q)$
- From $(1,0)$ to:
  - $(0,0)$: Probability $(1-p)$
  - $(0,1)$: Probability p
- From $(1,1)$ to:
  - $(1,0)$: Probability q
  - $(1,1)$: Probability $(1-q)$

Transition Graph

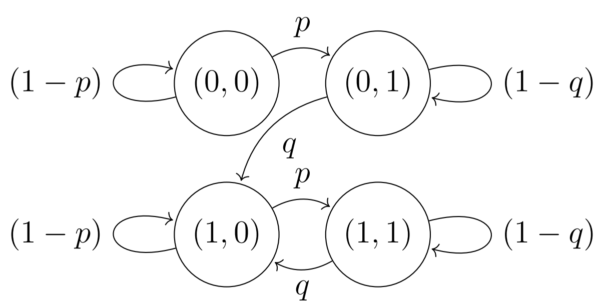

Figure 4: Transition Diagram for States $(0,0)$, $(0,1)$, $(1,0)$, and $(1,1)$

Thus, the transition matrix of the new chain is:

$$P' = \begin{bmatrix} (1-p) & p & 0 & 0 \\ 0 & 0 & q & (1-q) \\ (1-p) & p & 0 & 0 \\ 0 & 0 & q & (1-q) \end{bmatrix}.$$

Question 4

Markov chain model for dolphin activity with the following transition matrix:

$$A = \begin{bmatrix} 0.84 & 0.11 & 0.01 & 0.04 & 0.00 \\ 0.03 & 0.80 & 0.04 & 0.10 & 0.03 \\ 0.01 & 0.15 & 0.70 & 0.07 & 0.07 \\ 0.03 & 0.19 & 0.02 & 0.75 & 0.01 \\ 0.03 & 0.09 & 0.05 & 0.00 & 0.83 \end{bmatrix}.$$

To estimate the long-term distribution of dolphin activity, we need to find the stationary distribution $\pi = (\pi_1, \pi_2, \pi_3, \pi_4, \pi_5)$ such that:

$$\pi A = \pi,$$

with the constraint:

$$\sum_{i=1}^{5} \pi_i = 1.$$

Since there are five missing variables, we need five equations to form a system of equations, with a requirement that the equation $\pi_1 + \pi_2 + \pi_3 + \pi_4 + \pi_5 = 1$ is in the system.

$$\begin{aligned} 0.84\pi_1 + 0.03\pi_2 + 0.01\pi_3 + 0.03\pi_4 + 0.03\pi_5 &= \pi_1, \\ 0.11\pi_1 + 0.80\pi_2 + 0.15\pi_3 + 0.19\pi_4 + 0.09\pi_5 &= \pi_2, \\ 0.01\pi_1 + 0.04\pi_2 + 0.70\pi_3 + 0.02\pi_4 + 0.05\pi_5 &= \pi_3, \\ 0.04\pi_1 + 0.10\pi_2 + 0.07\pi_3 + 0.75\pi_4 + 0.00\pi_5 &= \pi_4, \\ \pi_1 + \pi_2 + \pi_3 + \pi_4 + \pi_5 &= 1. \end{aligned}$$

Solving this system yields the long-term distribution of dolphin activity:

```
# Define the coefficient matrix A
A <- matrix(c(
  0.84 - 1, 0.03, 0.01, 0.03, 0.03,
  0.11, 0.80 - 1, 0.15, 0.19, 0.09,
  0.01, 0.04, 0.70 - 1, 0.02, 0.05,
  0.04, 0.10, 0.07, 0.75 - 1, 0.00,
  1, 1, 1, 1, 1
), nrow = 5, byrow = TRUE)

# Define the right-hand side vector B
B <- c(0, 0, 0, 0, 1)

# Solve for pi using the solve function
pi_values <- solve(A, B)

# Print the solution
print(pi_values)
```

```
OUTPUT

[1] 0.1478358 0.4149254 0.0955597 0.2163806 0.1252985
```

(**key:** socializing, traveling, milling, feeding, and resting).

$$\pi \approx (0.1478358, 0.4149254, 0.0955597, 0.2163806, 0.1252985).$$

Dolphins in Patagonia primarily spend their time traveling (0.4149), likely due to environmental factors or human disturbances. Feeding (0.2164) is the second most common behavior, while socializing (0.1478) and resting (0.1253) occur less frequently. Milling (0.0956) is the least common, indicating minimal time spent in low-energy states. The findings suggest that human activities, such as tourism, may be influencing dolphin behavior by reducing their time for rest and low-energy activities.

Question 5

$$\begin{aligned}
\text{Rich} &\to \text{Average} : 0.75 \\
\text{Rich} &\to \text{Poor} : 0.20 \\
\text{Rich} &\to \text{InDebt} : 0.05 \\
\\
\text{Average} &\to \text{Rich} : 0.05 \\
\text{Average} &\to \text{Average} : 0.20 \\
\text{Average} &\to \text{InDebt} : 0.45 \\
\\
\text{Poor} &\to \text{Average} : 0.40 \\
\text{Poor} &\to \text{Poor} : 0.30 \\
\text{Poor} &\to \text{InDebt} : 0.20 \\
\\
\text{InDebt} &\to \text{Average} : 0.15 \\
\text{InDebt} &\to \text{Poor} : 0.30 \\
\text{InDebt} &\to \text{InDebt} : 0.55
\end{aligned}$$

The Discrete Markov Chain
Denoting the states corresponding to the rows(R_n) and columns(C_n):

- S_1 =C_1= R_1 = Rich
- S_2 =C_2= R_2 = Average
- S_3 =C_3= R_3 = Poor
- S_4 = C_4= R_4 = In Debt

Given the transition probabilities from the question, we have the transition matrix,

$$T = \begin{pmatrix} 0 & 0.75 & 0.2 & 0.05 \\ 0.05 & 0.2 & 0 & 0.45 \\ 0 & 0.4 & 0.3 & 0.2 \\ 0 & 0.15 & 0.3 & 0.55 \end{pmatrix}$$

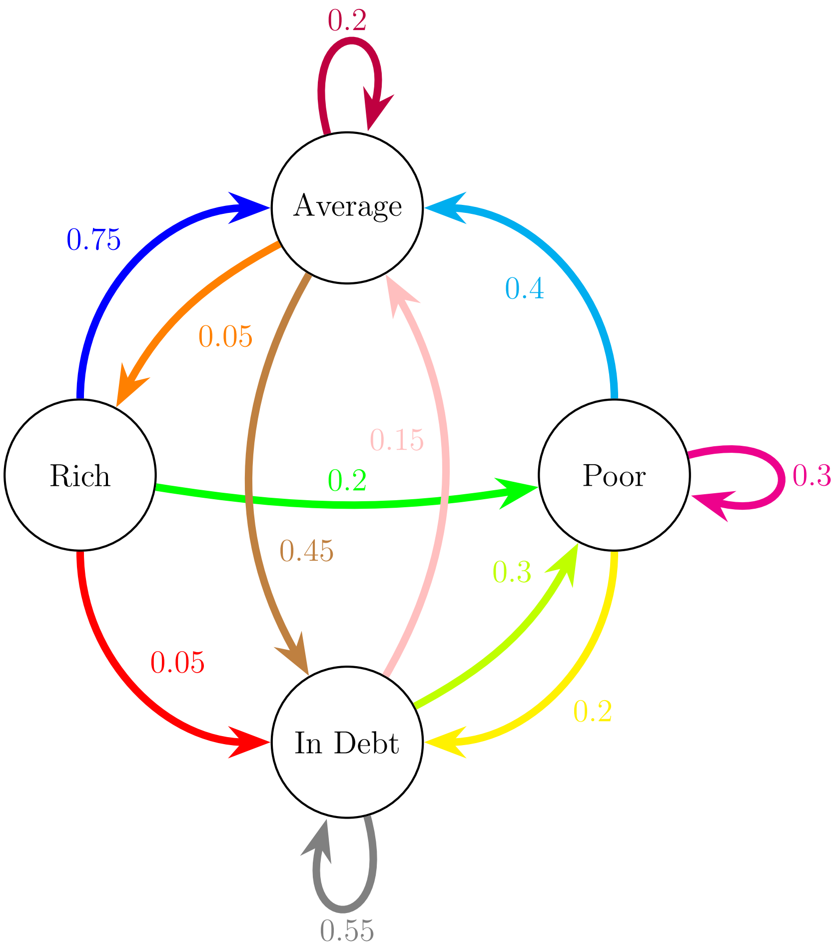

Figure 5: State Transition Diagram for Student States

The Stochastic Markov chain and Stochastic Transition Matrix

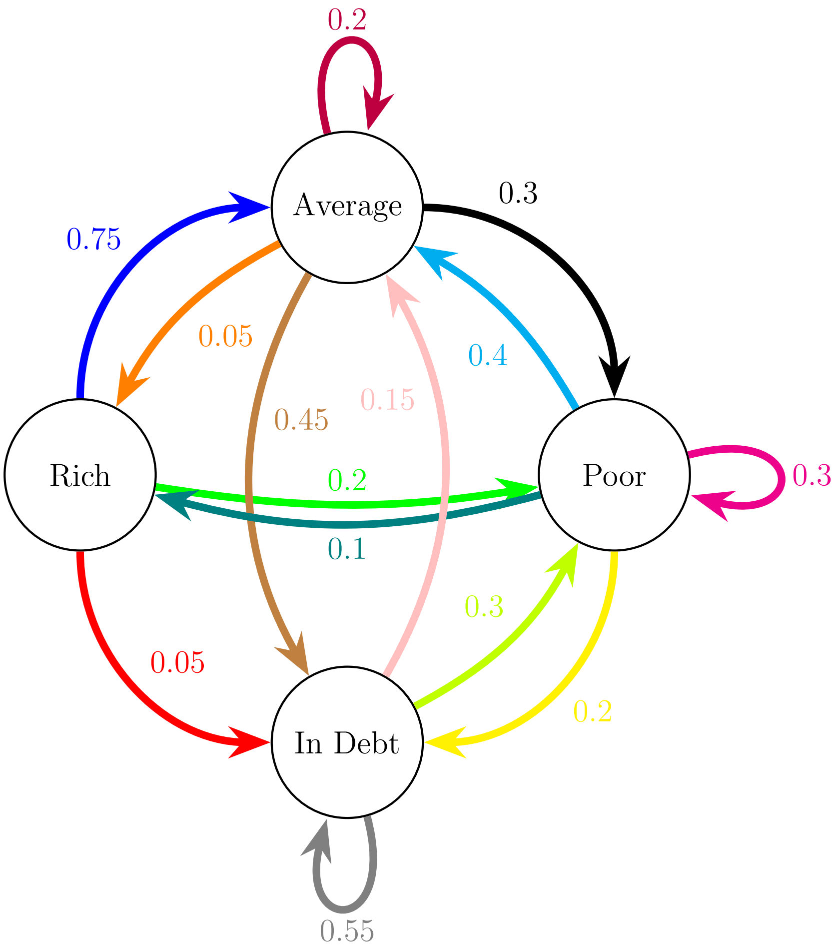

Figure 6: State Transition Diagram for Student States

Creating the stochastic matrix P will be:

$$P = \begin{pmatrix} 0 & 0.75 & 0.2 & 0.05 \\ 0.05 & 0.2 & 0.3 & 0.45 \\ 0.1 & 0.4 & 0.3 & 0.2 \\ 0 & 0.15 & 0.3 & 0.55 \end{pmatrix}$$

- Each row corresponds to the current state.
- Each column corresponds to the next state.
- The entry P_{ij} represents the probability of transitioning from state S_i to state S_j.

Moving from average to Rich in N steps

```
```{r}
# Define the transition matrix P
P <- matrix(c(0, 0.75, 0.2, 0.05,
              0.05, 0.2, 0.3, 0.45,
              0.1, 0.4, 0.3, 0.2,
              0, 0.15, 0.3, 0.55),
            nrow = 4, byrow = TRUE)

# Function to calculate the n-step transition matrix
calculate_n_step_matrix <- function(P, n) {
  return(P %^% n)
}

# Calculate the 1-step, 2-step transition matrices
P_1_step <- calculate_n_step_matrix(P, 1)

# Calculate the 2-step transition matrices
P_2_step <- calculate_n_step_matrix(P, 2)

# Calculate the 3-step transition matrices
P_3_step <- calculate_n_step_matrix(P, 3)

# Print the results
cat("1-step transition matrix:\n")
print(P_1_step)

cat("\n2-step transition matrix:\n")
print(P_2_step)

cat("\n3-step transition matrix:\n")
print(P_3_step)

```
```

```
OUTPUT

1-step transition matrix:
     [,1] [,2] [,3] [,4]
[1,] 0.00 0.75  0.2 0.05
[2,] 0.05 0.20  0.3 0.45
[3,] 0.10 0.40  0.3 0.20
[4,] 0.00 0.15  0.3 0.55
```

```
2-step transition matrix:
       [,1]   [,2]  [,3]  [,4]
[1,] 0.0575 0.2375 0.300 0.405
[2,] 0.0400 0.2650 0.295 0.400
[3,] 0.0500 0.3050 0.290 0.355
[4,] 0.0375 0.2325 0.300 0.430

3-step transition matrix:
         [,1]     [,2]    [,3]    [,4]
[1,] 0.041875 0.271375 0.29425 0.39250
[2,] 0.042750 0.261000 0.29600 0.40025
[3,] 0.044250 0.267750 0.29500 0.39300
[4,] 0.041625 0.259125 0.29625 0.40300
```

The stochastic matrix P is:

$$P = \begin{pmatrix} 0.00 & 0.75 & 0.2 & 0.05 \\ 0.05 & 0.20 & 0.3 & 0.45 \\ 0.10 & 0.40 & 0.3 & 0.20 \\ 0.00 & 0.15 & 0.3 & 0.55 \end{pmatrix}$$

i) 1 Time Step
The probability after 1 time step is simply the entry P_{21}.

$$P_{21} = 0.05 = 5\%$$

ii) 2 Time Steps

$$P^2 = \begin{pmatrix} 0.0575 & 0.2375 & 0.300 & 0.405 \\ 0.0400 & 0.2650 & 0.295 & 0.400 \\ 0.0500 & 0.3050 & 0.290 & 0.355 \\ 0.0375 & 0.2325 & 0.300 & 0.430 \end{pmatrix}$$

So, the probability of transitioning from state 2 to state 1 after 2 time steps is:

$$P_{21}^2 = 0.0400 = 4\%$$

iii) 3 Time Steps

$$P^3 = \begin{pmatrix} 0.041875 & 0.271375 & 0.29425 & 0.39250 \\ 0.042750 & 0.261000 & 0.29600 & 0.40025 \\ 0.044250 & 0.267750 & 0.29500 & 0.39300 \\ 0.041625 & 0.259125 & 0.29625 & 0.40300 \end{pmatrix}$$

So, the probability of transitioning from state 2 to state 1 after 3 time steps is:

$$P_{21}^3 = 0.042750 \approx 4.3\%$$

Question 6

Given a Markov chain $X_0, X_1, \ldots$ with an initial distribution α and transition matrix P, we want to justify the joint probability:

$$P(X_6 = i, X_{10} = j, X_{13} = k, X_{25} = l)$$

By the definition of conditional probability, we can decompose the joint probability as:

$$\begin{aligned}
P(X_6 = i, X_{10} = j, X_{13} = k, X_{25} = l) = {} & P(X_{25} = l \mid X_{13} = k, X_{10} = j, X_6 = i) \\
& \times P(X_{13} = k \mid X_{10} = j, X_6 = i) \\
& \times P(X_{10} = j \mid X_6 = i) P(X_6 = i)
\end{aligned}$$

Since the Markov property states that the future state only depends on the present state:

$$P(X_n \mid X_{n-1}, X_{n-2}, \ldots, X_0) = P(X_n \mid X_{n-1})$$

$$\begin{aligned}
P(X_{25} = l \mid X_{13} = k, X_{10} = j, X_6 = i) &= P(X_{25} = l \mid X_{13} = k), \\
P(X_{13} = k \mid X_{10} = j, X_6 = i) &= P(X_{13} = k \mid X_{10} = j), \\
P(X_{10} = j \mid X_6 = i) &= P(X_{10} = j \mid X_6 = i).
\end{aligned}$$

Thus, the joint probability simplifies to:

$$P(X_6 = i, X_{10} = j, X_{13} = k, X_{25} = l) =$$

$$P(X_{25} = l \mid X_{13} = k) P(X_{13} = k \mid X_{10} = j) P(X_{10} = j \mid X_6 = i) P(X_6 = i).$$

For a Markov chain with transition matrix P, the transition probabilities are given by:

$$P(X_n = s \mid X_m = r) = P_{r,s}^{(n-m)},$$

where $P_{r,s}^{(t)}$ is the (r, s)-entry of the t-step transition matrix P^t. Applying this:

$$\begin{aligned}
P(X_{10} = j \mid X_6 = i) &= P_{i,j}^{(4)}, \\
P(X_{13} = k \mid X_{10} = j) &= P_{j,k}^{(3)}, \\
P(X_{25} = l \mid X_{13} = k) &= P_{k,l}^{(12)}.
\end{aligned}$$

The probability of being in state i at time 6 where "m" runs over all possible states in the Markov chain is given by:

$$P(X_6 = i) = \sum_m \alpha_m P_{m,i}^{(6)} = (\alpha P^6)_i.$$

where α_m is the initial distribution. Thus, the joint probability is:

$$P(X_6 = i, X_{10} = j, X_{13} = k, X_{25} = l) = P_{k,l}^{(12)} P_{j,k}^{(3)} P_{i,j}^{(4)} \sum_m P_{m,i}^{(6)} \alpha_m. = (\alpha P^6)_i P_{k,l}^{(12)} P_{j,k}^{(3)} P_{i,j}^{(4)}$$