Week\_07

230701195

2025-04-07

# MARKOV CHAINS

# Problem 1:  
  
# A smart bulb operates in three modes: On (actively glowing), Off (powered down), and Idle (standby mode). The probability of transitioning between these modes every hour is as follows:  
  
# From\To On Off Idle  
# On 0.60 0.30 0.10  
# Off 0.10 0.80 0.10  
# Idle 0.30 0.30 0.40  
  
# Using R, i) Model the bulb's behavior as a Markov chain. ii) Determine the steady-state probabilities of each mode. iii) Simulate the bulb's state transitions for 20 hours starting from the "On" state. iv) Check if the Markov chain is ergodic.  
  
  
library(markovchain)

## Warning: package 'markovchain' was built under R version 4.4.3

## Loading required package: Matrix

## Package: markovchain  
## Version: 0.10.0  
## Date: 2024-11-14 00:00:02 UTC  
## BugReport: https://github.com/spedygiorgio/markovchain/issues

states <- c("On", "Off", "Idle")  
transition\_matrix <- matrix(  
 c(0.6, 0.3, 0.1,  
 0.1, 0.8, 0.1,  
 0.3, 0.3, 0.4),   
 nrow = 3, byrow = TRUE  
)  
  
mc <- new("markovchain", states = states, transitionMatrix = transition\_matrix)  
print(mc)

## On Off Idle  
## On 0.6 0.3 0.1  
## Off 0.1 0.8 0.1  
## Idle 0.3 0.3 0.4

steady\_state <- steadyStates(mc)  
print(steady\_state)

## On Off Idle  
## [1,] 0.2571429 0.6 0.1428571

set.seed(456)  
sim <- rmarkovchain(n = 20, object = mc, t0 = "On")  
cat("\nBulb Simulation for 30 days:\n\n")

##   
## Bulb Simulation for 30 days:

print(sim)

## [1] "On" "On" "Off" "On" "Off" "Off" "Off" "Off" "Off" "Off" "Off" "Off"  
## [13] "Off" "On" "On" "Off" "On" "On" "Off" "Off"

is.ergodic <- function(mc) {  
 return(is.irreducible(mc) && all(period(mc) == 1))  
}  
if (is.ergodic(mc)) {  
 cat("\nThe Markov chain is ergodic.\n")  
} else {  
 cat("\nThe Markov chain is not ergodic.\n")  
}

##   
## The Markov chain is ergodic.

# Problem 2:  
  
# A region's weather transitions daily between three states: Sunny, Cloudy, and Rainy. The daily weather pattern follows these probabilities:  
  
# From Sunny: 70% chance of remaining Sunny, 10% turning Rainy  
# From Cloudy: 30% chance of becoming Sunny, 50% remaining Cloudy  
# From Rainy: 30% becoming Cloudy, 50% remaining Rainy  
  
# Using R, i) Model the weather system as a Markov chain. ii) Determine the steady-state probabilities of each weather condition. iii) Simulate the weather for 30 days starting from a Sunny day. iv) Check if the Markov chain is ergodic.  
  
library(markovchain)  
  
states <- c("Sunny", "Cloudy", "Rainy")  
transition\_matrix <- matrix(  
 c(0.7, 0.2, 0.1,  
 0.3, 0.5, 0.2,  
 0.2, 0.3, 0.5),   
 nrow = 3, byrow = TRUE  
)  
  
mc\_weather <- new("markovchain", states = states, transitionMatrix = transition\_matrix)  
  
print(mc\_weather)

## Sunny Cloudy Rainy  
## Sunny 0.7 0.2 0.1  
## Cloudy 0.3 0.5 0.2  
## Rainy 0.2 0.3 0.5

steady\_state\_weather <- steadyStates(mc\_weather)  
print(steady\_state\_weather)

## Sunny Cloudy Rainy  
## [1,] 0.4634146 0.3170732 0.2195122

set.seed(789)  
sim\_weather <- rmarkovchain(n = 30, object = mc\_weather, t0 = "Sunny")  
cat("\nWeather Simulation for 30 days:\n\n")

##   
## Weather Simulation for 30 days:

print(sim\_weather)

## [1] "Sunny" "Sunny" "Sunny" "Sunny" "Sunny" "Sunny" "Sunny" "Sunny"   
## [9] "Sunny" "Sunny" "Sunny" "Sunny" "Sunny" "Sunny" "Sunny" "Sunny"   
## [17] "Sunny" "Sunny" "Cloudy" "Cloudy" "Cloudy" "Cloudy" "Cloudy" "Cloudy"  
## [25] "Cloudy" "Sunny" "Sunny" "Sunny" "Cloudy" "Sunny"

is.ergodic <- function(mc) {  
 return(is.irreducible(mc) && all(period(mc) == 1))  
}  
if (is.ergodic(mc\_weather)) {  
 cat("\nThe weather Markov chain is ergodic.\n")  
} else {  
 cat("\nThe weather Markov chain is not ergodic.\n")  
}

##   
## The weather Markov chain is ergodic.

# Problem 3:  
  
# A small website has three pages: A, B, and C. The transition matrix for users moving between pages is:  
  
# From \ To A B C  
# A 0.1 0.6 0.3  
# B 0.3 0.4 0.3  
# C 0.4 0.2 0.4  
  
# i) Define the Markov Chain.  
# ii) Compute steady-state probabilities (PageRank).  
# iii) Simulate user navigation for 20 steps starting at page A.  
  
library(markovchain)  
  
states <- c("A", "B", "C")  
trans\_matrix <- matrix(  
 c(0.1, 0.6, 0.3,   
 0.3, 0.4, 0.3,   
 0.4, 0.2, 0.4),  
 byrow = TRUE, nrow = 3,  
 dimnames = list(states, states)  
)  
  
mc\_pagerank <- new("markovchain", transitionMatrix = trans\_matrix)  
  
cat("PageRank Transition Matrix:\n")

## PageRank Transition Matrix:

print(mc\_pagerank)

## A B C  
## A 0.1 0.6 0.3  
## B 0.3 0.4 0.3  
## C 0.4 0.2 0.4

steady\_state <- steadyStates(mc\_pagerank)  
cat("\nPageRank Steady-State Probabilities:\n")

##   
## PageRank Steady-State Probabilities:

print(steady\_state)

## A B C  
## [1,] 0.2777778 0.3888889 0.3333333

set.seed(123)  
sim\_navigation <- rmarkovchain(n = 20, object = mc\_pagerank, t0 = "A")  
cat("\nSimulated User Navigation (20 Steps):\n")

##   
## Simulated User Navigation (20 Steps):

print(sim\_navigation)

## [1] "B" "C" "C" "B" "C" "A" "B" "C" "C" "C" "B" "A" "C" "C" "A" "C" "A" "B" "B"  
## [20] "C"

is.ergodic <- function(mc) {  
 is.irreducible(mc) && all(period(mc) == 1)  
}  
  
if (is.ergodic(mc\_pagerank)) {  
 cat("\nThe PageRank Markov chain is ergodic.\n")  
} else {  
 cat("\nThe PageRank Markov chain is not ergodic.\n")  
}

##   
## The PageRank Markov chain is ergodic.

# Problem 4:  
  
# A stock market can be in three states: Bull, Bear, or Stagnant. The transition probabilities are:  
  
# From Bull: 20% Bear, 10% Stagnant.  
# From Bear: 40% Bull, 20% Stagnant.  
# From Stagnant: 30% Bull, 30% Bear.  
  
# i) Define the Markov Chain.  
# ii) Simulate the stock market over 12 months, starting in a Bull market.  
# iii) Find the probability of being in a Bear market after 6 steps.  
  
  
library(markovchain)  
  
states <- c("Bull", "Bear", "Stagnant")  
trans\_matrix <- matrix(c(0.7, 0.2, 0.1,  
 0.4, 0.4, 0.2,  
 0.3, 0.3, 0.4),  
 nrow = 3, byrow = TRUE)  
  
mc <- new("markovchain", states = states, transitionMatrix = trans\_matrix)  
  
cat("Transition Matrix:\n")

## Transition Matrix:

print(mc)

## Bull Bear Stagnant  
## Bull 0.7 0.2 0.1  
## Bear 0.4 0.4 0.2  
## Stagnant 0.3 0.3 0.4

cat("\nSteady-State Probabilities:\n")

##   
## Steady-State Probabilities:

print(steadyStates(mc))

## Bull Bear Stagnant  
## [1,] 0.5454545 0.2727273 0.1818182

set.seed(123)  
cat("\nSimulation (12 Months from Bull):\n")

##   
## Simulation (12 Months from Bull):

print(rmarkovchain(n = 12, object = mc, t0 = "Bull"))

## [1] "Bull" "Bear" "Bear" "Stagnant" "Bear" "Bull"   
## [7] "Bull" "Bear" "Bear" "Bear" "Stagnant" "Bull"

sim\_6 <- rmarkovchain(n = 6, object = mc, t0 = "Bull")  
cat("\nState after 6 Steps:\n")

##   
## State after 6 Steps:

print(sim\_6)

## [1] "Bull" "Bull" "Bull" "Bear" "Bull" "Bull"

prob\_bear <- mean(sim\_6 == "Bear")  
cat("\nProbability of Bear after 6 Steps:", prob\_bear, "\n")

##   
## Probability of Bear after 6 Steps: 0.1666667

if (is.irreducible(mc) && all(period(mc) == 1)) {  
 cat("The Markov chain is ergodic.\n")  
} else {  
 cat("The Markov chain is not ergodic.\n")  
}

## The Markov chain is ergodic.