VISHWANATH M
231901062

MA23435
PROBABILITY, STATISTICS AND SIMULATION

VII. MARKOV CHAINS ANALYSIS USING MARKOVCHAIN PACKAGE IN R

Creating and Analyzing Markov Chains

```
> states <- c("Healthy", "Fever")
> transition_matrix <- matrix(c(0.7, 0.3, 0.4, 0.6),  nrow = 2, byrow
= TRUE)
> mc <- new("markovchain", states = states, transitionMatrix =
transition_matrix)
> print(mc)
        Healthy Fever
Healthy     0.7   0.3
Fever       0.4   0.6
```

Steady State (Stationary Distribution)

```
> steady_state  <- steadyStates(mc)
> print(steady_state)
       Healthy     Fever
[1,] 0.5714286 0.4285714
```

Simulating Markov Chains

```
> set.seed(123)
> sim <- rmarkovchain(n = 10, object = mc,  t0 = "Healthy")
> print(sim)
 [1] "Healthy" "Fever"   "Fever"   "Healthy" "Fever"   "Fever"
"Fever"   "Healthy"
 [9] "Healthy" "Healthy"
```

Ergodicity and Long-Run Behavior

```
> library(markovchain)
>
> # Define a transition matrix
> states <- c("Healthy", "Fever")
> transition_matrix <- matrix(c(0.7, 0.3,
+                               0.4, 0.6),
+                             nrow = 2, byrow = TRUE)
>
> # Create a Markov chain
> mc <- new("markovchain", states = states, transitionMatrix =
transition_matrix)
>
> # Check for irreducibility
> is_irreducible <- is.irreducible(mc)
>
> # Check for aperiodicity
```

VISHWANATH M
231901062

```
> is_aperiodic <- all(period(mc) == 1)
>
> # Determine ergodicity
> if (is_irreducible && is_aperiodic) {
+  cat("The Markov chain is ergodic.\n")
+ } else {
+  cat("The Markov chain is not ergodic.\n")
+ }
The Markov chain is ergodic.
```

Exercises

Exercise 1: Simple Markov Chain Analysis

o **Task:** Create a Markov chain for a system with three states (e.g., "On", "Off", "Idle") and transition probabilities provided.

```
> library(markovchain)
>
> 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")
> print(sim)
 [1] "On"  "On"  "Off" "On"  "Off" "Off" "Off" "Off" "Off" "Off"
"Off" "Off" "Off"
[14] "On"  "On"  "Off" "On"  "On"  "Off" "Off"
>
> is_irreducible <- is.irreducible(mc)
```

VISHWANATH M
231901062

```
> is_aperiodic <- all(period(mc) == 1)
>
> if (is_irreducible && is_aperiodic) {
+  cat("The Markov chain is ergodic.\n")
+ } else {
+  cat("The Markov chain is not ergodic.\n")
+ }
The Markov chain is ergodic.
```

Exercise 2: Real-World Application

o **Task:** Apply Markov chains to model a practical scenario (e.g., weather patterns, stock market behavior) using data or assumptions. Formulate a Markov chain, analyze its properties, and interpret the results.

```
> 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")
> print(sim_weather)
 [1] "Sunny"  "Sunny"  "Sunny"  "Sunny"  "Sunny"  "Sunny"  "Sunny"
"Sunny"  "Sunny"
[10] "Sunny"  "Sunny"  "Sunny"  "Sunny"  "Sunny"  "Sunny"  "Sunny"
"Sunny"  "Sunny"
[19] "Cloudy" "Cloudy" "Cloudy" "Cloudy" "Cloudy" "Cloudy" "Cloudy"
"Sunny"  "Sunny"
[28] "Sunny"  "Cloudy" "Sunny"
>
> is_irreducible <- is.irreducible(mc_weather)
> is_aperiodic <- all(period(mc_weather) == 1)
>
```

VISHWANATH M
231901062

```
> if (is_irreducible && is_aperiodic) {
+   cat("The weather Markov chain is ergodic.\n")
+ } else {
+   cat("The weather Markov chain is not ergodic.\n")
+ }
The weather Markov chain is ergodic.
```

Exercise 1: Disease Progression Model

```
> library(markovchain)
> disease_matrix <- matrix(
+   c(0.85, 0.10, 0.05,
+     0.10, 0.70, 0.20,
+     0.00, 0.05, 0.95),
+   byrow = TRUE,
+   nrow = 3,
+   dimnames = list(c("Healthy", "Infected", "Recovered"),
+                   c("Healthy", "Infected", "Recovered"))
+ )
>
> disease_chain <- new("markovchain", states = c("Healthy",
"Infected", "Recovered"),
+                      transitionMatrix = disease_matrix)
>
> steady_state <- steadyStates(disease_chain)
> print("Steady-State Probabilities:")
[1] "Steady-State Probabilities:"
> print(steady_state)
       Healthy  Infected Recovered
[1,] 0.1052632 0.1578947 0.7368421
>
> set.seed(123)
> simulation <- rmarkovchain(n = 15, object = disease_chain, t0 =
"Healthy")
> print("Disease Progression Simulation:")
[1] "Disease Progression Simulation:"
> print(simulation)
 [1] "Healthy"  "Healthy"  "Healthy"  "Infected" "Healthy"  "Healthy"
"Healthy"
 [8] "Infected" "Infected" "Infected" "Healthy"  "Healthy"  "Healthy"
"Healthy"
[15] "Healthy"
>
> is_irreducible <- is.irreducible(disease_chain)
> is_aperiodic <- all(period(disease_chain) == 1)
>
> if (is_irreducible && is_aperiodic) {
+   cat("The disease Markov chain is ergodic.\n")
+ } else {
+   cat("The disease Markov chain is not ergodic.\n")
+ }
```

VISHWANATH M
231901062

```
The disease Markov chain is ergodic.
```

Exercise 2: PageRank Algorithm Simulation

```
> pagerank_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(c("A", "B", "C"), c("A", "B", "C"))
+ )
> pagerank_chain <- new("markovchain", states = c("A", "B", "C"),
+                       transitionMatrix = pagerank_matrix)
> pagerank <- steadyStates(pagerank_chain)
> print("PageRank Steady-State Probabilities:")
[1] "PageRank Steady-State Probabilities:"
> print(pagerank)
             A         B         C
[1,] 0.2777778 0.3888889 0.3333333
> set.seed(123)
> navigation <- rmarkovchain(n = 20, object = pagerank_chain, t0 =
"A")
> print("Simulated User Navigation:")
[1] "Simulated User Navigation:"
> print(navigation)
 [1] "B" "C" "C" "B" "C" "A" "B" "C" "C" "C" "B" "A" "C" "C" "A" "C"
"A" "B" "B" "C"
```

Exercise 3: Stock Market Behavior

```
> stock_market_matrix <- matrix(
+   c(0.7, 0.2, 0.1,
+     0.4, 0.4, 0.2,
+     0.3, 0.3, 0.4),
+   byrow = TRUE,
+   nrow = 3,
+   dimnames = list(c("Bull", "Bear", "Stagnant"),
+                   c("Bull", "Bear", "Stagnant"))
+ )
> stock_market_chain <- new("markovchain", states = c("Bull", "Bear",
"Stagnant"),
+                           transitionMatrix = stock_market_matrix)
> set.seed(123)
> simulation <- rmarkovchain(n = 12, object = stock_market_chain, t0
= "Bull")
> print("Stock Market Behavior Simulation:")
[1] "Stock Market Behavior Simulation:"
> print(simulation)
 [1] "Bull"     "Bear"     "Bear"     "Stagnant" "Bear"     "Bull"
"Bull"
```

VISHWANATH M
231901062

```
 [8] "Bear"     "Bear"     "Bear"     "Stagnant" "Bull"
> six_step_matrix <- stock_market_chain^6
> print("Transition Matrix After 6 Steps:")
[1] "Transition Matrix After 6 Steps:"
> print(six_step_matrix)
             Bull     Bear Stagnant
Bull     0.546375 0.272375 0.181250
Bear     0.544750 0.273000 0.182250
Stagnant 0.543750 0.273375 0.182875
```

Exercise 4: Absorbing States - Prisoner Dilemma

```
> prisoner_matrix <- matrix(
+  c(0.5, 0.4, 0.1,
+    0.3, 0.5, 0.2,
+    0.0, 0.0, 1.0),
+  byrow = TRUE,
+  nrow = 3,
+  dimnames = list(c("Free", "Trial", "Jail"), c("Free", "Trial",
"Jail"))
+ )
> prisoner_chain <- new("markovchain", states = c("Free", "Trial",
"Jail"),
+                       transitionMatrix = prisoner_matrix)
> absorbing <- absorbingStates(prisoner_chain)
> print("Absorbing States:")
[1] "Absorbing States:"
> print(absorbing)
[1] "Jail"
> set.seed(123)
> simulation <- rmarkovchain(n = 10, object = prisoner_chain, t0 =
"Free")
> print("Prisoner Dilemma Simulation:")
[1] "Prisoner Dilemma Simulation:"
> print(simulation)
 [1] "Free"  "Trial" "Trial" "Jail"  "Jail"  "Jail"  "Jail"  "Jail"
"Jail"  "Jail"
```

Exercise 5: Manufacturing Defects

```
> manufacturing_matrix <- matrix(
+  c(0.9, 0.1, 0.0,
+    0.3, 0.5, 0.2,
+    0.8, 0.1, 0.1),
+  byrow = TRUE,
+  nrow = 3,
+  dimnames = list(c("Good", "Defective", "Repaired"),
+                  c("Good", "Defective", "Repaired"))
+ )
> manufacturing_chain <- new("markovchain", states = c("Good",
"Defective", "Repaired"),
```

VISHWANATH M
231901062

```
+                             transitionMatrix = manufacturing_matrix)
> steady_state <- steadyStates(manufacturing_chain)
> print("Steady-State Probabilities:")
[1] "Steady-State Probabilities:"
> print(steady_state)
          Good Defective   Repaired
[1,] 0.7962963 0.1666667 0.03703704
> set.seed(123)
> simulation <- rmarkovchain(n = 20, object = manufacturing_chain, t0
= "Defective")
> print("Manufacturing Process Simulation:")
[1] "Manufacturing Process Simulation:"
> print(simulation)
 [1] "Defective" "Good"      "Good"      "Good"      "Defective"
"Defective"
 [7] "Good"      "Good"      "Good"      "Good"      "Defective"
"Defective"
[13] "Good"      "Good"      "Good"      "Good"      "Good"
"Good"
[19] "Good"      "Defective"
```