CTMC and Queueing Lab: Vaccine Clinic Operations

Table of contents

1 Introduction

In this short lab, we'll explore how Continuous-Time Markov Chains (CTMCs) and Queueing Theory apply to healthcare settings using realistic examples. The goal is to build intuition and hands-on familiarity through code you can run and interpret.

We'll use the example of a vaccine clinic with limited observation space, followed by a queueing system involving patients waiting for service.

---

2 Part I: CTMC – Vaccine Observation Room

In a rural health facility, adults are being vaccinated against rubella. After receiving their shots, each patient must be observed for side effects in a dedicated room. Due to space limits, only two patients can be monitored at a time. Initially, patients arrive approximately once every minute. If there is space,

they are admitted; otherwise, they are turned away. When alone in the room, each person is monitored for 10 minutes. If two are observed together, both are monitored for 8 minutes due to resource constraints.

2.1 Task 1. Construct the CTMC Generator Matrix

Can you build a model of the possible occupancy changes in this small clinic? Think about transitions between 0, 1, or 2 patients and how often they happen.

```
Q <- matrix(c(
  __, __, __,
  __, __, __,
  __, __, __
), nrow = 3, byrow = TRUE)
Q
```

2.2 Task 2. How much time does the clinic spend in each state?

Over a long day, how much time does the clinic spend in each state — 0, 1, or 2 patients in the observation room?

```
steady_state <- function(Q) {
  A <- t(Q) # Transpose Q so rows become equations
  A[nrow(A), ] <- rep(1, ncol(Q))  # Replace last row with 1s for normalization
  b <- c(__, __, __) # RHS of the linear system (0s and 1
  solve(__, __) #Solve the linear system A * pi = b
}

pi <- steady_state(Q)
names(pi) <- paste0("State ", 0:2)
pi
```

2.3 Task 3: How Does the System Converge to Steady State?

Let's visualize the convergence to the steady-state distribution by tracking the probability of each state over time.

2.3.1 Understanding Convergence in a CTMC

Before we plot how the system converges, let's understand the math behind it.

In a **Continuous-Time Markov Chain (CTMC)**, the system moves between states randomly over continuous time. To know how the system evolves, we want to compute:

What is the probability of being in each state after time t?

This is given by the formula:

> **State distribution at time t = Initial distribution × exp(Q × t)**

- The **initial distribution** is a row vector (e.g., `p0 <- c(1, 0, 0)` if the system starts in State 0).
- The **matrix exponential**, written as `exp(Q × t)` in R, gives the **transition probability matrix** at time `t`.
- Multiplying the two gives us the **probability distribution across states** at that time.

We repeat this calculation for many time points (e.g., from 0 to 100 minutes) to see **how the system changes over time** and eventually **settles into a steady state**.

In the next task, you'll:

- Define the time points
- Set the initial condition
- Use a loop to calculate the probability vector at each time
- Plot how the state probabilities evolve

Let's implement this now.

```
# Define a sequence of time points
times <- seq(___, ___, by = ___)

# Set the initial state distribution (starts in State 0)
p0 <- c(___, ___, ___)

# Compute how probabilities evolve over time
probs_over_time <- t(sapply(times, function(t) p0 %*% expm(___)))

# Plot the result
matplot(times, probs_over_time, type = "l", lty = 1, lwd = 2,
        col = c("blue", "orange", "red"),
        ylab = "Probability", xlab = "Time",
        main = "Convergence to Steady State")

legend("right", legend = c("State 0", "State 1", "State 2"),
        col = c("blue", "orange", "red"), lty = 1, lwd = 2)
```

Explain: How long does it take for the system to reach equilibrium? Which state dominates during the transient phase?

Answer:

2.4 Task 4. How many people are typically in the room?

On average, how many people are being observed in the room at any given time? This gives a sense of how often the room is full or idle.

```
sum(0:2 * __)
```

2.5 Task 5. What if more patients start coming in?

Due to a new outreach campaign and the arrival of a donor-funded vaccine shipment, community interest in rubella vaccination has surged. As a result, patients are now arriving every 30 minutes, doubling the previous arrival rate. Observation lasts 20 minutes when one person is present. If two patients are present, they both stay for 17 minutes.

How does this change the expected number of patients inside the room?

```
Q2 <- matrix(c(
  __, __, __,
  __, __, __,
  __, __, __
), nrow = 3, byrow = TRUE)

pi2 <- steady_state(__)
round(pi2, 4)

sum(0:2 * __)  # new expected number
```

3 Part II: Queueing – Patient Check-in Desk

In the same rural vaccine clinic, patients begin their visit at a **check-in desk** staffed by a single nurse. Patients arrive randomly, on average once every two minutes. Each check-in takes about 1.5 minutes and is handled one at a time. If the nurse is already helping someone, the remaining patients wait their turn. There is no upper limit on how many patients can wait in line.

3.1 Task 6. How busy is the nurse, and how long do patients wait?

The clinic administrator wants to know: How busy is the nurse? How long does each patient spend waiting and checking in? And how many people are typically in the system?

```r
lambda <- __  # arrivals per minute
mu <- __      # service rate per minute (1/1.5)

rho <- __ / __
L <- __ / (1 - __)
Lq <- __^2 / (1 - __)
W <- 1 / (mu * (1 - __))
Wq <- __ / (mu * (1 - __))

list(
  Utilization = round(rho, 2),
  "Avg in System (L)" = round(L, 2),
  "Avg in Queue (Lq)" = round(Lq, 2),
  "Time in System (W)" = round(W, 2),
  "Time in Queue (Wq)" = round(Wq, 2)
)
```

3.2 Task 7. What happens when more patients keep coming?

Imagine the clinic gets busier over time — more patients show up without any increase in staff. Let's see how the average number of patients and the uncertainty around that number change.

```r
# Define a sequence of traffic intensity ( ) values from 0.01 to 0.99
rho_vals <- seq(0.01, 0.99, by = 0.01)

# Calculate the average number of patients in the system (L)
L_vals <- ___ / (1 - ___)   # FILL IN: Use rho_vals

# Calculate the variance of the number of patients in the system
# We calculate the variance to understand how unpredictable the system is.
# A high variance means it's harder to estimate how many patients are actually there at any
Var_vals <- ___ * (1 + ___ - ___^2) / (1 - ___)^2   # FILL IN: Use rho_vals multiple times

# Plot the results
plot(rho_vals, L_vals, type = "l", col = "red", lwd = 2,
     ylim = c(0, 40), ylab = "Mean & Variance in System", xlab = "Traffic Intensity ( )")
lines(rho_vals, Var_vals, col = "blue", lwd = 2, lty = 2)
legend("topleft", legend = c("Mean", "Variance"), col = c("red", "blue"), lty = c(1, 2), lwd
```

Explain: What does the graph suggest about the workload and predictability of the check-in desk as it gets busier?

Answer
