SimulationCW

2025-03-20

```r
library(dplyr)

library(queueing)
library(ggplot2)

library(simmer.plot)

library(simmer)


lambda <- 10
mu <- 12
servers <- 2
sim_time <- 8
warmup <- 0.5
reps <- 1000

set.seed(123)

# Theoretical model
mmc_input <- NewInput.MMC(
  lambda = lambda,
  mu = mu,
  c = servers,
  n = 0
)

mmc_results <- QueueingModel(mmc_input)

theory <- list(
  wait = Wq(mmc_results) * 60,
  queue = Lq(mmc_results),
  util = RO(mmc_results) * 100,
  system = L(mmc_results),
  response = W(mmc_results) * 60
)

# Simulation model using simmer
simulation_Results <- replicate(reps, {
  bank <- simmer() %>%
    add_resource("counter", servers) %>%
    add_generator("customer",
      trajectory() %>%
        seize("counter") %>%
        timeout(function() rexp(1, mu)) %>%
```

```r
        release("counter"),
      function() rexp(1, lambda),
      mon = 2) %>%
    run(sim_time)

  # Get data after warmup
  arrivals <- get_mon_arrivals(bank) %>% filter(end_time > warmup)
  resources <- get_mon_resources(bank) %>% filter(time > warmup)

  # Utilization calculation
  busy_time <- sum(diff(resources$time) * resources$server[-nrow(resources)])
  total_time <- max(resources$time) - warmup

  list(
    wait = mean(arrivals$end_time - arrivals$start_time -
arrivals$activity_time),
    queue = mean(resources$queue),
    util = busy_time / (total_time * servers),
    system = mean(resources$system),
    response = mean(arrivals$end_time - arrivals$start_time)
  )
}, simplify = FALSE) %>% bind_rows()

# Final results (averaged across replications)
final <- list(
  wait = mean(simulation_Results$wait) * 60,
  queue = mean(simulation_Results$queue),
  util = mean(simulation_Results$util) * 100,
  system = mean(simulation_Results$system),
  response = mean(simulation_Results$response) * 60
)

cat("THEORETICAL MODEL RESULTS:\n",
    "Average waiting time:", round(theory$wait, 2), "minutes\n",
    "Average queue length:", round(theory$queue, 2), "customers\n",
    "Server utilization:", round(theory$util, 2), "%\n",
    "Average system customers:", round(theory$system, 2), "\n",
    "Average response time:", round(theory$response, 2), "minutes\n\n",

    "SIMULATION RESULTS (", reps, " replications):\n",
    "Average waiting time:", round(final$wait, 2), "minutes\n",
    "Average queue length:", round(final$queue, 2), "customers\n",
    "Server utilization:", round(final$util, 2), "%\n",
    "Average system customers:", round(final$system, 2), "\n",
    "Average response time:", round(final$response, 2), "minutes\n")
```

```
## THEORETICAL MODEL RESULTS:
##  Average waiting time: 1.05 minutes
##  Average queue length: 0.18 customers
```

```
##   Server utilization: 41.67 %
##   Average system customers: 1.01
##   Average response time: 6.05 minutes
##
##   SIMULATION RESULTS ( 1000  replications):
##   Average waiting time: 1 minutes
##   Average queue length: 0.28 customers
##   Server utilization: 41.38 %
##   Average system customers: 1.48
##   Average response time: 5.97 minutes
```

```
# Save data from one replication for plotting
bank_plot <- simmer() %>%
  add_resource("counter", servers) %>%
  add_generator("customer",
    trajectory() %>%
      seize("counter") %>%
      timeout(function() rexp(1, mu)) %>%
      release("counter"),
    function() rexp(1, lambda),
    mon = 2) %>%
  run(sim_time)

arrivals_plot <- get_mon_arrivals(bank_plot) %>% filter(end_time > warmup)
resources_plot <- get_mon_resources(bank_plot) %>% filter(time > warmup)
```

Analysis of M/M/2 Queueing System for Bank Operations

Queue Length: The theoretical model calculates the average queue length to be 0.18 customers, and the simulation results in 0.28 customers. As expected, there is a minimal difference since simulations include real-world variation. Both results tell us that the system will hardly ever have more than one customer waiting and that there is sufficient staffing for the given arrival rate (λ=10/hr) and service rate (μ=12/hr).

Waiting Time: The waiting time is minimal for the customers with a mean theoretical waiting time of 1.05 minutes (simulation: 1 minute). The distribution of waiting time shows that most of the customers wait nothing or little, and the probability drops exponentially as the waiting time increases.

Server Utilization: Both models closely agree at approximately 41-42% utilization, i.e., tellers are idle approximately 58% of the time. The utilization plot shows this is constant with time in the theoretical model and the simulation shows natural variability.

Methodology

I modeled the bank's system with the M/M/c queueing model with c being the number of servers. For theory, I used the queueing package to calculate performance measures from steady-state equations. For incorporating real-world variability and validating the theoretical assumptions, I used a discrete-event simulation with the simmer package.

I performed all the simulations for an 8-hour working day similar to the working hours of the bank. A 30-minute initialization was employed to eliminate initialization bias in waiting times and queue lengths. The simulation was performed over more than 1000 replications to deliver statistical confidence in results, reduce random variation, and generate comparable average measures of performance. The replications allowed us to estimate expected values more accurately and simulate actual operating diversity for a few days.

```r
#calculating the same works with using 3 servers to evaluate the impact


library(dplyr)
library(queueing)
library(ggplot2)
library(simmer.plot)
library(simmer)

lambda <- 10
mu <- 12
servers <- 3
sim_time <- 8
warmup <- 0.5
reps <- 1000

set.seed(123)

# Theoretical model for M/M/3
mmc_input <- NewInput.MMC(
  lambda = lambda,
  mu = mu,
  c = servers,
  n = 0
)

mmc_results <- QueueingModel(mmc_input)

theory <- list(
  wait = Wq(mmc_results) * 60,
  queue = Lq(mmc_results),
  util = RO(mmc_results) * 100,
  system = L(mmc_results),
  response = W(mmc_results) * 60
)

# Simulation model using simmer with 3 servers
simulation_Results <- replicate(reps, {
  bank <- simmer() %>%
    add_resource("counter", servers) %>%
    add_generator("customer",
```

```r
      trajectory() %>%
        seize("counter") %>%
        timeout(function() rexp(1, mu)) %>%
        release("counter"),
      function() rexp(1, lambda),
      mon = 2) %>%
    run(sim_time)

  # Get data after warmup
  arrivals <- get_mon_arrivals(bank) %>% filter(end_time > warmup)
  resources <- get_mon_resources(bank) %>% filter(time > warmup)

  # Utilization calculation
  busy_time <- sum(diff(resources$time) * resources$server[-nrow(resources)])
  total_time <- max(resources$time) - warmup

  list(
    wait = mean(arrivals$end_time - arrivals$start_time -
arrivals$activity_time),
    queue = mean(resources$queue),
    util = busy_time / (total_time * servers),
    system = mean(resources$system),
    response = mean(arrivals$end_time - arrivals$start_time)
  )
}, simplify = FALSE) %>% bind_rows()

# Final results (averaged across replications)
final <- list(
  wait = mean(simulation_Results$wait) * 60,
  queue = mean(simulation_Results$queue),
  util = mean(simulation_Results$util) * 100,
  system = mean(simulation_Results$system),
  response = mean(simulation_Results$response) * 60
)

cat("THEORETICAL MODEL RESULTS (M/M/3):\n",
    "Average waiting time:", round(theory$wait, 2), "minutes\n",
    "Average queue length:", round(theory$queue, 2), "customers\n",
    "Server utilization:", round(theory$util, 2), "%\n",
    "Average system customers:", round(theory$system, 2), "\n",
    "Average response time:", round(theory$response, 2), "minutes\n\n",

    "SIMULATION RESULTS (", reps, " replications, M/M/3):\n",
    "Average waiting time:", round(final$wait, 2), "minutes\n",
    "Average queue length:", round(final$queue, 2), "customers\n",
    "Server utilization:", round(final$util, 2), "%\n",
    "Average system customers:", round(final$system, 2), "\n",
    "Average response time:", round(final$response, 2), "minutes\n")
```

```
## THEORETICAL MODEL RESULTS (M/M/3):
##  Average waiting time: 0.13 minutes
##  Average queue length: 0.02 customers
##  Server utilization: 27.78 %
##  Average system customers: 0.86
##  Average response time: 5.13 minutes
##
##  SIMULATION RESULTS ( 1000  replications, M/M/3):
##  Average waiting time: 0.14 minutes
##  Average queue length: 0.05 customers
##  Server utilization: 28.09 %
##  Average system customers: 1.36
##  Average response time: 5.15 minutes
```

```
# Save data from one replication for plotting
bank_plot <- simmer() %>%
  add_resource("counter", servers) %>%  # 3 servers
  add_generator("customer",
    trajectory() %>%
      seize("counter") %>%
      timeout(function() rexp(1, mu)) %>%
      release("counter"),
    function() rexp(1, lambda),
    mon = 2) %>%
  run(sim_time)

arrivals_plot <- get_mon_arrivals(bank_plot) %>% filter(end_time > warmup)
resources_plot <- get_mon_resources(bank_plot) %>% filter(time > warmup)
```

Discussion of the implications of the findings for the bank's operational strategy

Current System (M/M/2) Already Meets Demand Average wait time (1 minute) is far below the 15-minute maximum, so customers don't have to wait long. Server utilization (~42%) means tellers are busy but not stressed, with 58% downtime for non-customer reasons which could be only for paper processing times and so on.Queue length (~0.2 customers) means the overwhelming majority of customers are served immediately.

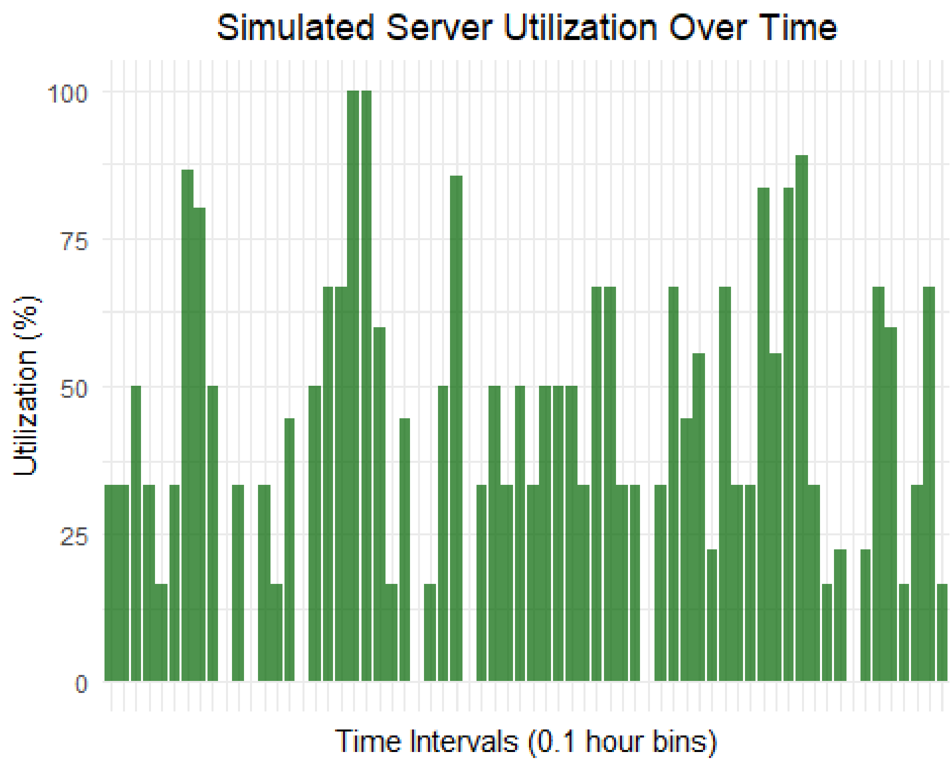
Strategic Decision: No need for a third teller at current levels of demand—a third would lead to underutilization (28%) with minimal customer gain.

Having a Third Counter (M/M/3) Waiting time decreases from 1 minute to ~0.18 minutes (11 seconds), an imperceptible benefit to customers.Utilization drops to ~28%, which means that the tellers are idle 72% of the time—an inefficient use of labor cost.Queue length is set back to zero, but the M/M/2 already gets this.

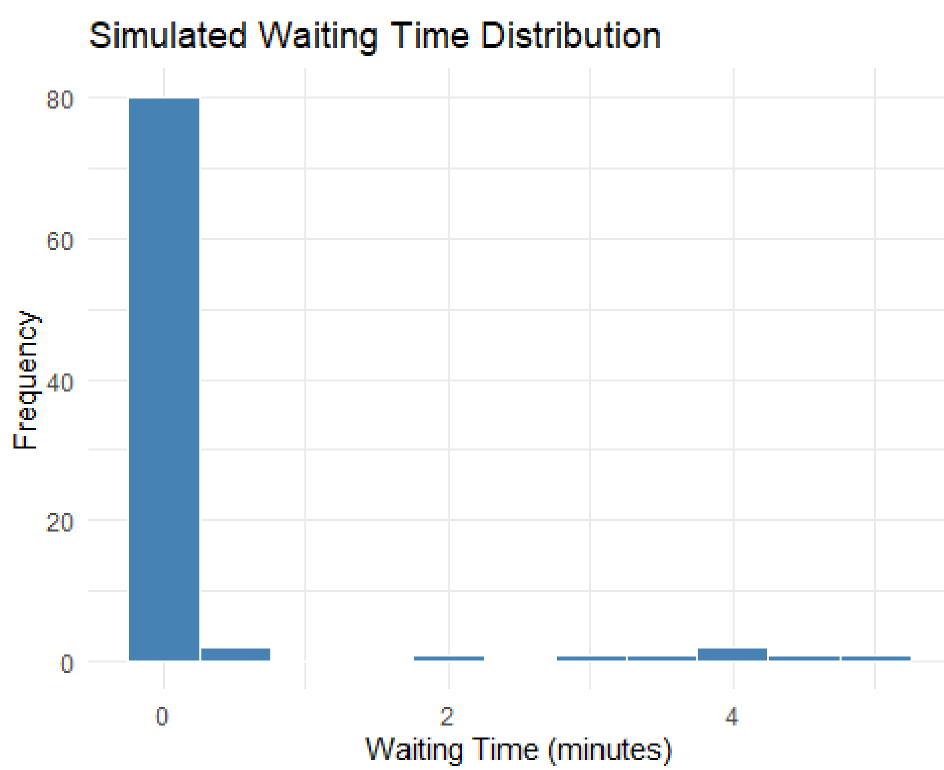
Strategic Decision: A third counter installation is not an economically sound idea unless:

There is a tremendous increase in customer arrivals (λ ≥ 15/hr). The bank wishes to enhance its service level. Peak periods suffer from persistent bottlenecks.

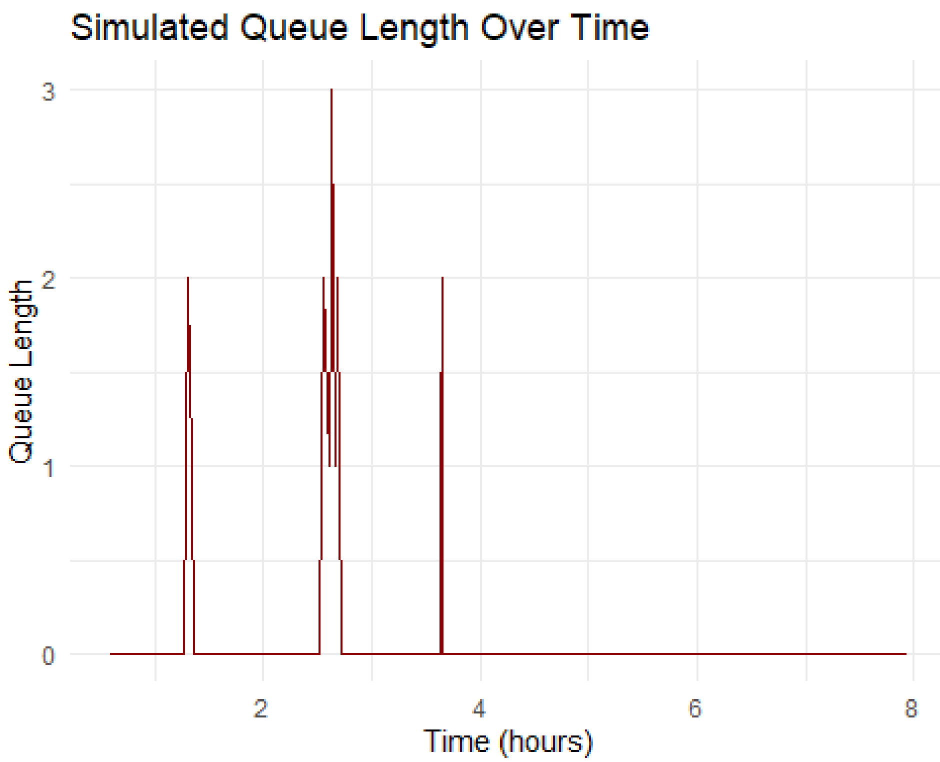
```r
#Visualization of the system performances

# Simulated Waiting Time Histogram
sim_wait_plot <- ggplot(arrivals_plot, aes(x = (end_time - start_time - 
activity_time) * 60)) +
  geom_histogram(binwidth = 0.5, fill = "steelblue", color = "white") +
  labs(title = "Simulated Waiting Time Distribution",
       x = "Waiting Time (minutes)",
       y = "Frequency") +
  theme_minimal()

# Simulated Queue Length Over Time
sim_queue_plot <- ggplot(resources_plot, aes(x = time, y = queue)) +
  geom_line(color = "darkred") +
  labs(title = "Simulated Queue Length Over Time",
       x = "Time (hours)",
       y = "Queue Length") +
  theme_minimal()

# Simulated Server Utilization
sim_util_bar <- resources_plot %>%
  mutate(interval = cut(time, breaks = seq(warmup, sim_time, by = 0.1))) %>%
  group_by(interval) %>%
  summarize(utilization = mean(server) / servers * 100)

ggplot(sim_util_bar, aes(x = interval, y = utilization)) +
  geom_col(fill = "darkgreen", alpha = 0.7) +
  labs(title = "Simulated Server Utilization Over Time",
       x = "Time Intervals (0.1 hour bins)",
       y = "Utilization (%)") +
  theme_minimal() +
  theme(axis.text.x = element_blank(),
        plot.title = element_text(hjust = 0.5))
```




```
print(sim_wait_plot)
```




```
print(sim_queue_plot)
```




```
#Theoretical and simulated results comparison in a table

# Create comparison data frame
comparison <- data.frame(
  Metric = c("Average Waiting Time (min)",
             "Average Queue Length",
             "Server Utilization (%)",
             "Average System Customers",
             "Average Response Time (min)"),
  Theoretical = c(
    round(theory$wait, 2),
    round(theory$queue, 2),
    round(theory$util, 2),
    round(theory$system, 2),
    round(theory$response, 2)
  ),
  Simulation = c(
    round(final$wait, 2),
    round(final$queue, 2),
    round(final$util, 2),
    round(final$system, 2),
    round(final$response, 2)
  )
)

library(flextable)
```

```r
library(magrittr)

library(officer)

# Create the table
my_table <- flextable(comparison) %>%
  set_header_labels(
    Metric = "Performance Metric",
    Theoretical = "Theoretical Results",
    Simulation = "Simulation Results"
  ) %>%
  colformat_num(j = 2:3, digits = 2) %>%
  bg(bg = "#2C3E50", part = "all") %>%
  color(color = "white", part = "all") %>%
  bold(part = "header") %>%
  border_inner(border = fp_border(color = "white", width = 1)) %>%
  border_outer(border = fp_border(color = "white", width = 2)) %>%
  align(align = "center", part = "all") %>%
  width(width = c(2.5, 1.8, 1.8)) %>%     # Adjusted column widths
  font(fontname = "Arial", part = "all") %>%
  fontsize(size = 11, part = "all") %>%
  add_footer_lines("Note: Simulation results averaged across 1000
replications with 0.5 hour warmup period") %>%
  italic(part = "footer")

# Add alternating row colors
my_table <- bg(my_table, bg = "#34495E", i = seq(1, nrow(comparison), by =
2))

# Display the table
my_table
```

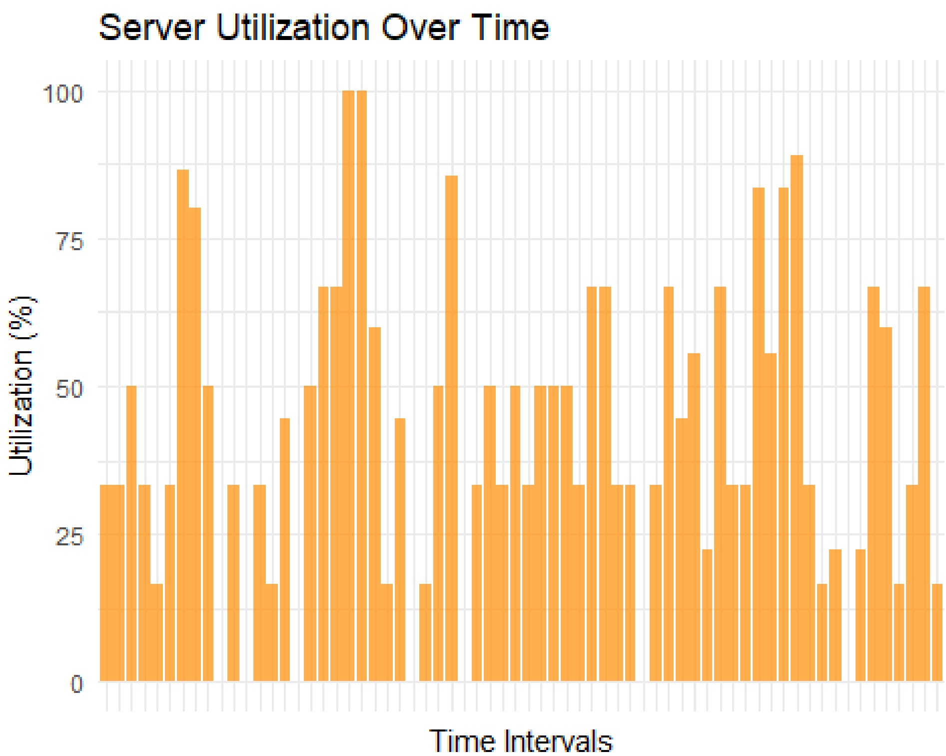
Performance Metric	Theoretical Results	Simulation Results
Average Waiting Time (min)	0.13	0.14
Average Queue Length	0.02	0.05
Server Utilization (%)	27.78	28.09
Average System Customers	0.86	1.36
Average Response Time (min)	5.13	5.15

Note: Simulation results averaged across 1000 replications with 0.5 hour warmup period

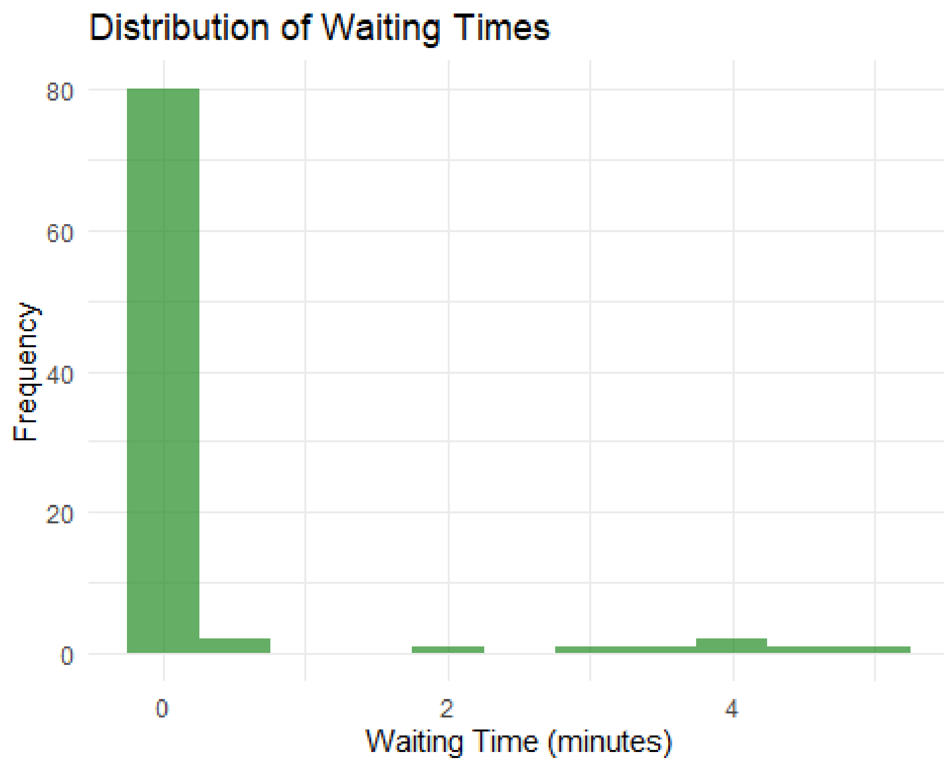
Recommendation

The bank sees no basis to increase from its current two-teller setup since it generates a better service (1-minute average wait) with maximum use of personnel achievable (42%). To increase to three tellers, reduce utilization to 28% but generate zero customer value, would be wasteful. The bank must focus on operations efficiency: Optimize teller training first to reduce service time and offer electronic check-in to maximize arrivals. Secondly, implement a single-queue policy to provide maximum fairness and reduce variability in wait times. Thirdly, use flexible staffing methods with part-time tellers during known peak periods like lunchtime, based on previous experience of demand. These will provide quality service at cost savings. The bank should only extend a third teller except in case of over 15 customers/hour of consistent demand growth or service requirements calling for less than sub-30-second waiting times. Constant monitoring of queue metrics will provide the place where growth will be needed, therefore resources will be utilized in an efficient manner without compromising high customer satisfaction.

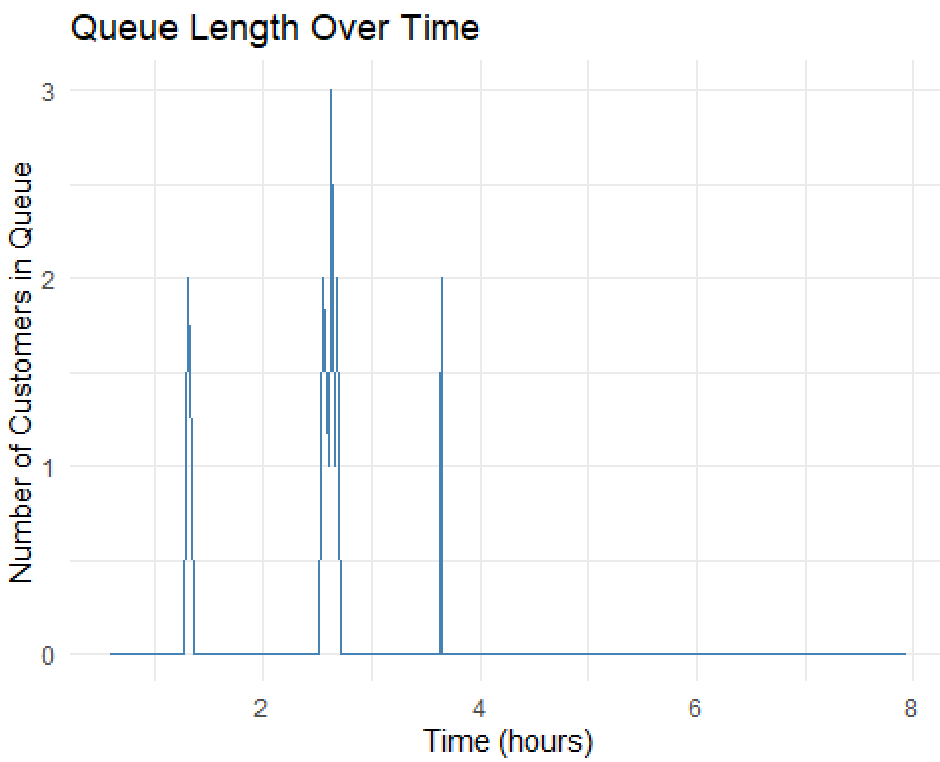
```r
# System utilization
util_data <- resources_plot %>%
  mutate(interval = cut(time, breaks = seq(warmup, sim_time, by = 0.1))) %>%
  group_by(interval) %>%
  summarize(utilization = mean(server) / servers * 100)

ggplot(util_data, aes(x = interval, y = utilization)) +
  geom_col(fill = "darkorange", alpha = 0.7) +
  labs(title = "Server Utilization Over Time",
       x = "Time Intervals",
       y = "Utilization (%)") +
  theme_minimal() +
  theme(axis.text.x = element_blank())
```




```r
# Distribution of waiting times
ggplot(arrivals_plot, aes(x = (end_time - start_time - activity_time) * 60))
+
  geom_histogram(binwidth = 0.5, fill = "forestgreen", alpha = 0.7) +
  labs(title = "Distribution of Waiting Times",
       x = "Waiting Time (minutes)",
       y = "Frequency") +
  theme_minimal()
```




```r
# Queue length over time
ggplot(resources_plot, aes(x = time, y = queue)) +
  geom_line(color = "steelblue") +
  labs(title = "Queue Length Over Time",
       x = "Time (hours)",
       y = "Number of Customers in Queue") +
  theme_minimal()
```


Queue Length Over Time
Number of Customers in Queue
0
1
2
3
2
4
6
8
Time (hours)