Simulation and Modeling Technique Coursework SMT CW

Thusharkanth_20233168_2410213

2025-04-11

— Model Development —

```
# loading the nessary librearies

library(simmer)

## Warning: package 'simmer' was built under R version 4.4.2

library(dplyr)

##
## Attaching package: 'dplyr'

## The following object is masked from 'package:simmer':
##
##     select

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

library(ggplot2)

## Warning: package 'ggplot2' was built under R version 4.4.1

# Ensures consistent random values for reproducibility

set.seed(2002)

# declearting the varibles/Parameters

simulation_duration <- 480             # Total simulation time in minutes (8
hours = 480 min)
customer_arrival_rate <- 10 / 60       # 10 customers per hour = 1 every 6
mins ( Poisson distribution)
average_service_time <- 5              # Avg service time per customer in
minutes ( Exp distribution)
```

— CUSTOMER TRAJECTORY —

```
customer_flow <- trajectory("Customer Flow") %>%
  seize("service_counter", 1) %>%                                  # Take one
service counter
  timeout(function() rexp(1, 1 / average_service_time)) %>%    # Service time
(randomly generated)
  release("service_counter", 1)                                    # Release
counter after service
```

— FUNCTION: Run Simulation —

```
# Defining a Function for Simulating the bank system for a given number of
counters

simulate_bank_service <- function(number_of_counters) {

  # Create a simulation environment

  bank_simulation <- simmer(paste("Bank ", number_of_counters, "Counter(s)"))
%>%
    add_resource("service_counter", capacity = number_of_counters) %>%
# Add counters
    add_generator("Customer", customer_flow, function() rexp(1,
customer_arrival_rate)) %>%  # Add custmr
    run(until = simulation_duration)
# Run sim

    # Collect logs

  resource_logs <- get_mon_resources(bank_simulation)  # Logs about counter
usage
  arrival_logs <- get_mon_arrivals(bank_simulation)    # Logs about each
customer

    # Calculate customer waiting times

  arrival_logs$total_waiting_time <- arrival_logs$end_time -
arrival_logs$start_time  # Total time in sys
  arrival_logs$pure_waiting_time <- arrival_logs$total_waiting_time -
arrival_logs$activity_time  #Only queue time

    # Calculate key performance metrics
  avg_total_wait <- mean(arrival_logs$total_waiting_time, na.rm = TRUE)    #
Includes service time
  avg_actual_wait <- mean(arrival_logs$pure_waiting_time, na.rm = TRUE)    #
Just waiting time
  avg_queue_length <- mean(resource_logs$queue, na.rm = TRUE)              #
```

```
Avg number in queue



    # Server utilization (how much busy the counters were)
  total_time_busy <- sum(arrival_logs$activity_time, na.rm = TRUE)
  total_time_available <- simulation_duration * number_of_counters
  server_utilization <- total_time_busy / total_time_available

    # PRINT RESULTS -----

  cat("\n--- Results:", number_of_counters, "Counter(s) ---\n")

  cat("Avg Waiting Time (including service time):", round(avg_total_wait, 2),
"mins\n")

  cat("Avg Waiting Time (only in queue, excluding service):",
round(avg_actual_wait, 2), "mins\n")

  cat("Average Queue Length:", round(avg_queue_length, 2), "\n")

  cat("Server Utilization:", round(server_utilization * 100, 2), "%\n")




    # Return all useful outputs

  return(list(
    resource_logs = resource_logs,
    arrival_logs = arrival_logs,
    avg_queue_length = avg_queue_length,
    avg_total_wait = avg_total_wait,
    avg_actual_wait = avg_actual_wait,
    utilization = server_utilization
  ))
}
```

— RUN SIMULATIONS FOR 2 COUNTERS —

```
results_counter_2 <- simulate_bank_service(2)  # Simulate with 2 counters

##
## --- Results: 2 Counter(s) ---
## Avg Waiting Time (including service time): 6.13 mins
## Avg Waiting Time (only in queue, excluding service): 1.45 mins
## Average Queue Length: 0.3
## Server Utilization: 45.38 %
```

— RUN SIMULATIONS FOR 3 COUNTERS —

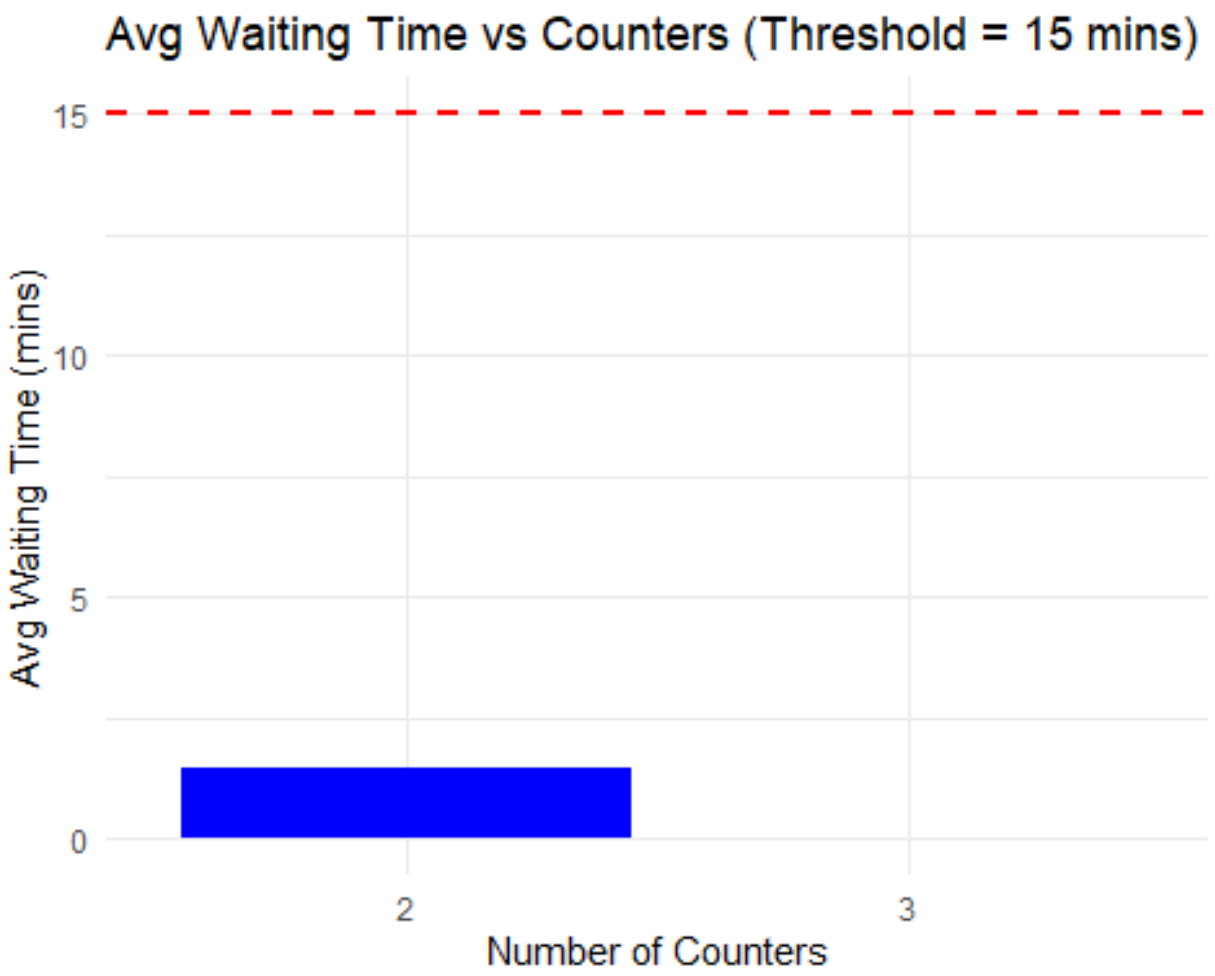
```
results_counter_3 <- simulate_bank_service(3)  # Simulate with 3 counters

##
## --- Results: 3 Counter(s) ---
## Avg Waiting Time (including service time): 5.47 mins
## Avg Waiting Time (only in queue, excluding service): 0 mins
## Average Queue Length: 0.01
## Server Utilization: 33.4 %
```

— RUN SIMULATIONS FOR 1 COUNTERS —

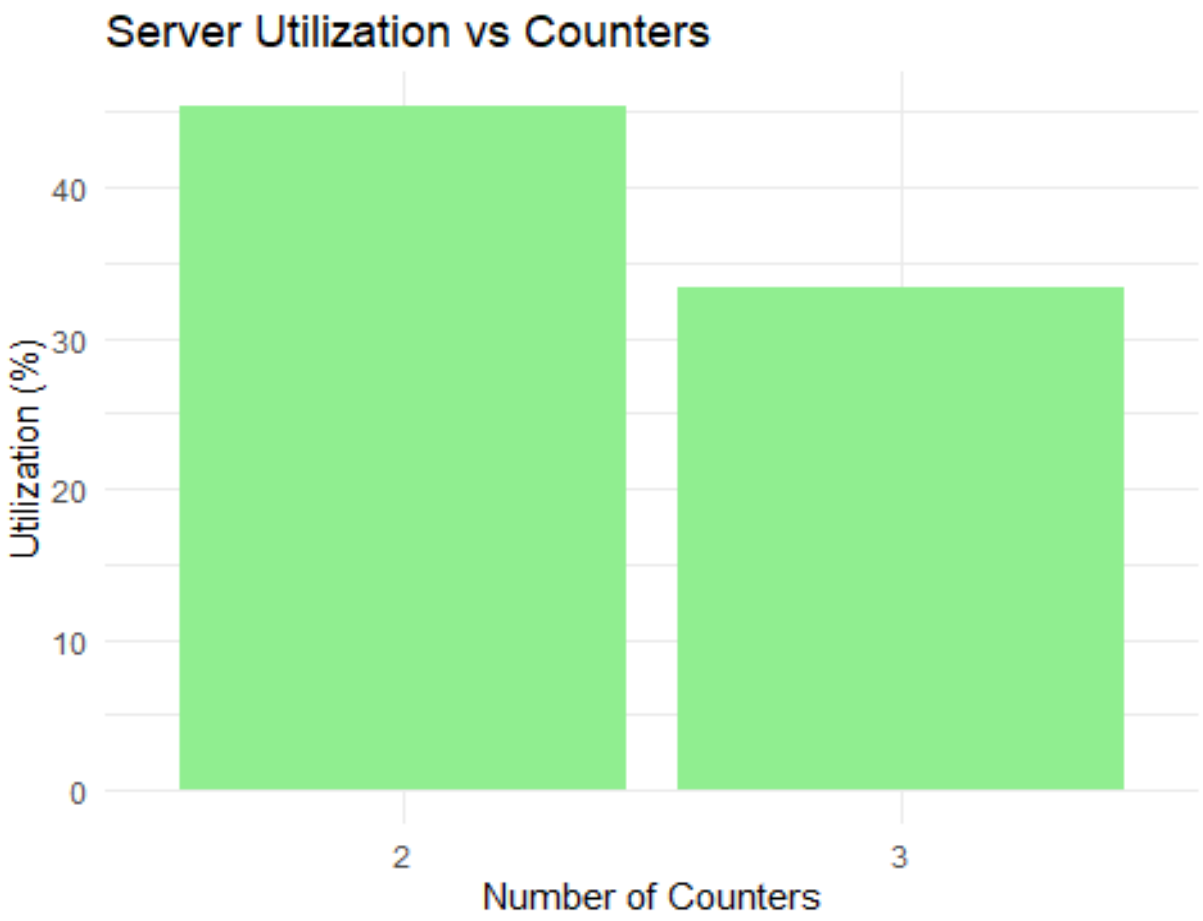
```
results_counter_1 <- simulate_bank_service(1)  # Simulate with 1 counter

##
## --- Results: 1 Counter(s) ---
## Avg Waiting Time (including service time): 8.42 mins
## Avg Waiting Time (only in queue, excluding service): 4.13 mins
## Average Queue Length: 0.7
## Server Utilization: 66.22 %
```

— VISUALIZATION OF COUNTER 2 AND 3 —

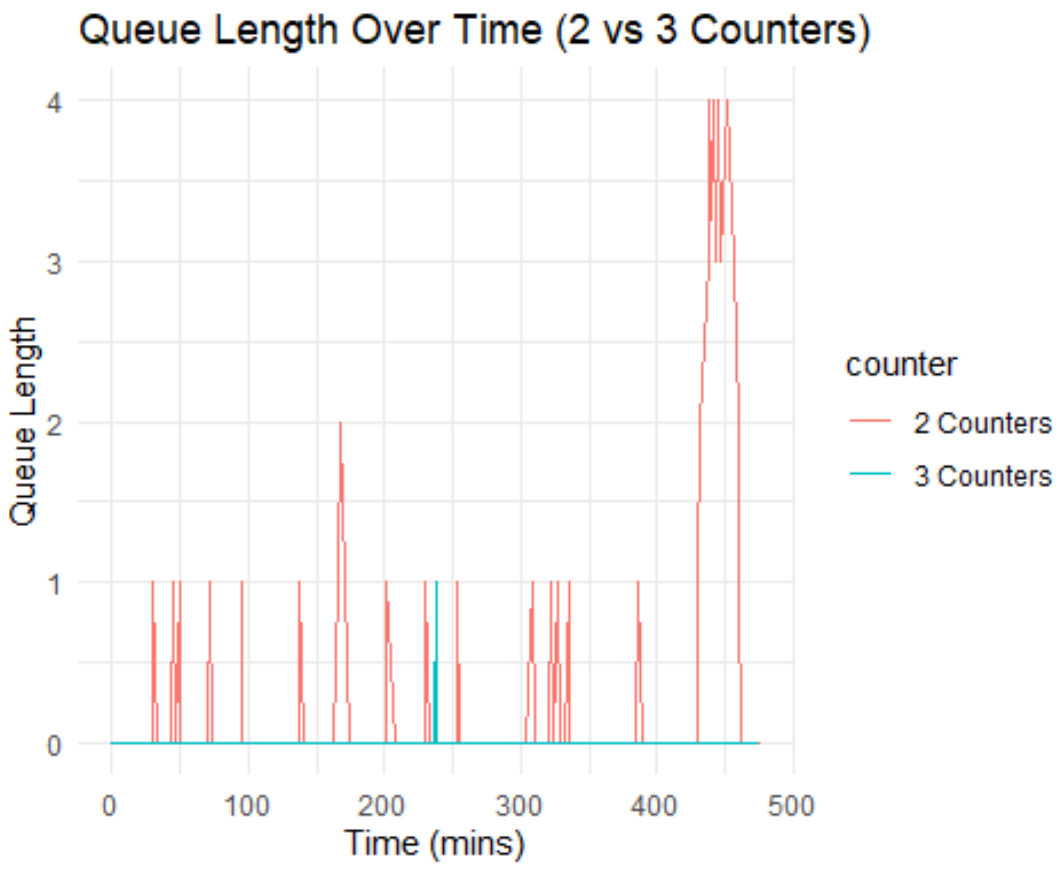
```
# Create DataFrame for 2 and 3 counters

df_23 <- data.frame(
  Counters = c(2, 3),
  Avg_Queue_Length = c(results_counter_2$avg_queue_length,
results_counter_3$avg_queue_length),
  Avg_Waiting_Time = c(results_counter_2$avg_actual_wait,
results_counter_3$avg_actual_wait),
  Utilization = c(results_counter_2$utilization,
results_counter_3$utilization)
)

# Bar Chart: Avg Waiting Time with 15-min Threshold
ggplot(df_23, aes(x = factor(Counters), y = Avg_Waiting_Time)) +
  geom_bar(stat = "identity", fill = "blue") +
  geom_hline(yintercept = 15, color = "red", linetype = "dashed", size = 1) +
  labs(title = "Avg Waiting Time vs Counters (Threshold = 15 mins)",
       x = "Number of Counters", y = "Avg Waiting Time (mins)") +
  theme_minimal()

## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## ℹ Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```




```r
# Bar Chart: Server Utilization
ggplot(df_23, aes(x = factor(Counters), y = Utilization * 100)) +
  geom_bar(stat = "identity", fill = "lightgreen") +
  labs(title = "Server Utilization vs Counters", x = "Number of Counters", y
= "Utilization (%)") +
  theme_minimal()
```



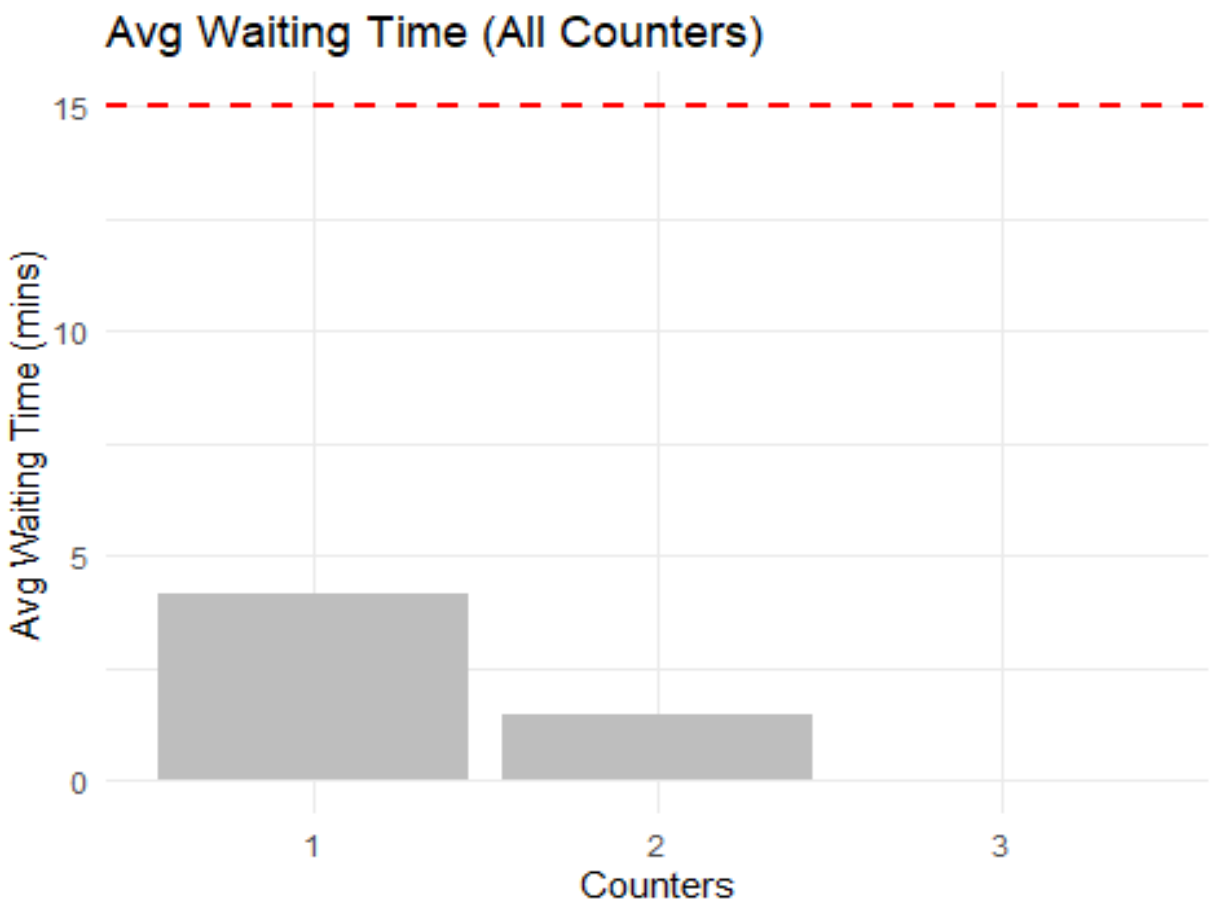

```r
# Line Chart: Avg Queue Length Over Time
r2 <- results_counter_2$resource_logs
r3 <- results_counter_3$resource_logs
r2$counter <- "2 Counters"
r3$counter <- "3 Counters"
queue_df <- rbind(r2, r3)

ggplot(queue_df, aes(x = time, y = queue, color = counter)) +
  geom_line(size = 0.5) +
  labs(title = "Queue Length Over Time (2 vs 3 Counters)",
       x = "Time (mins)", y = "Queue Length") +
  theme_minimal()
```



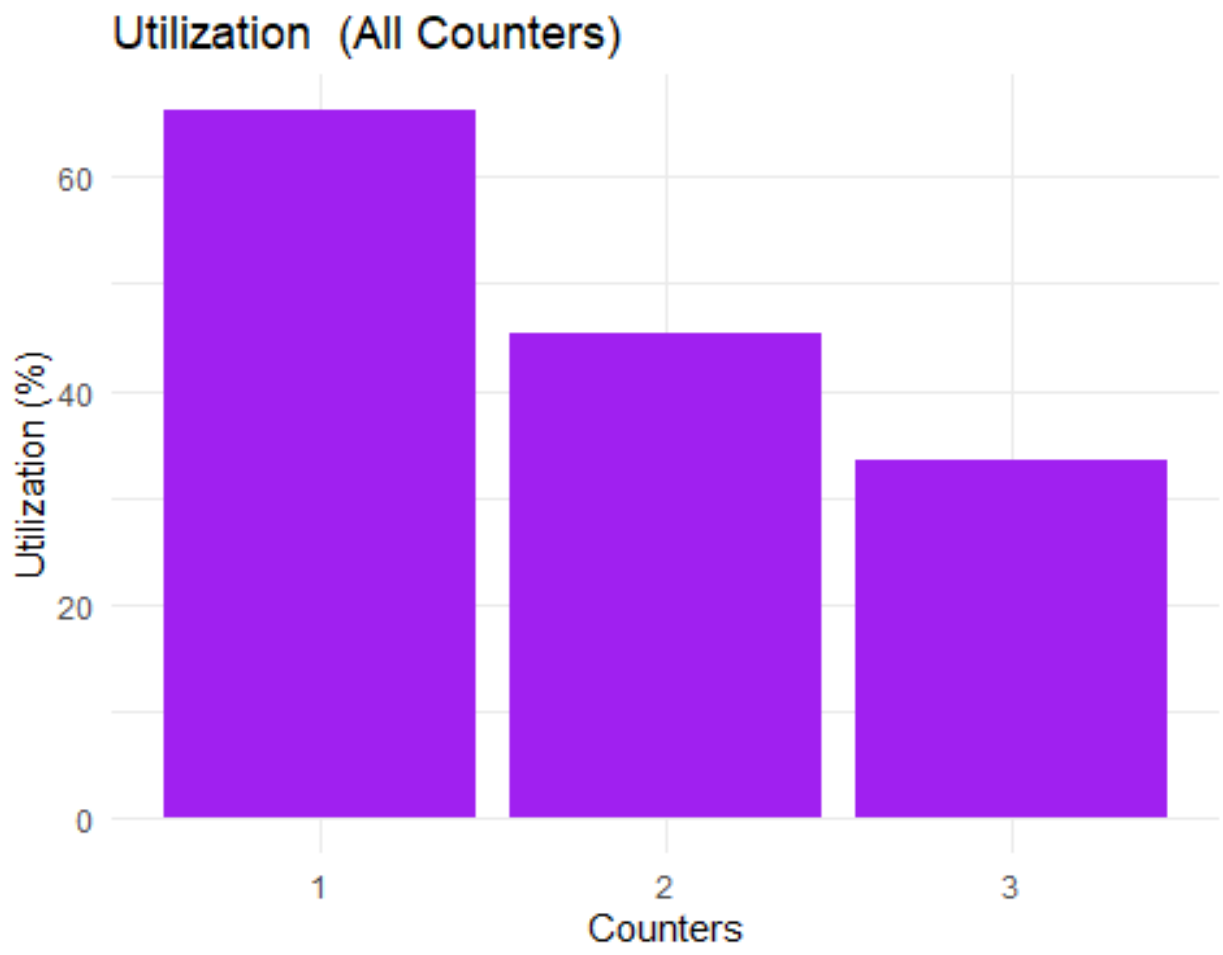
—

VISUALIZATION OF COUNTER 1, 2 AND 3 —

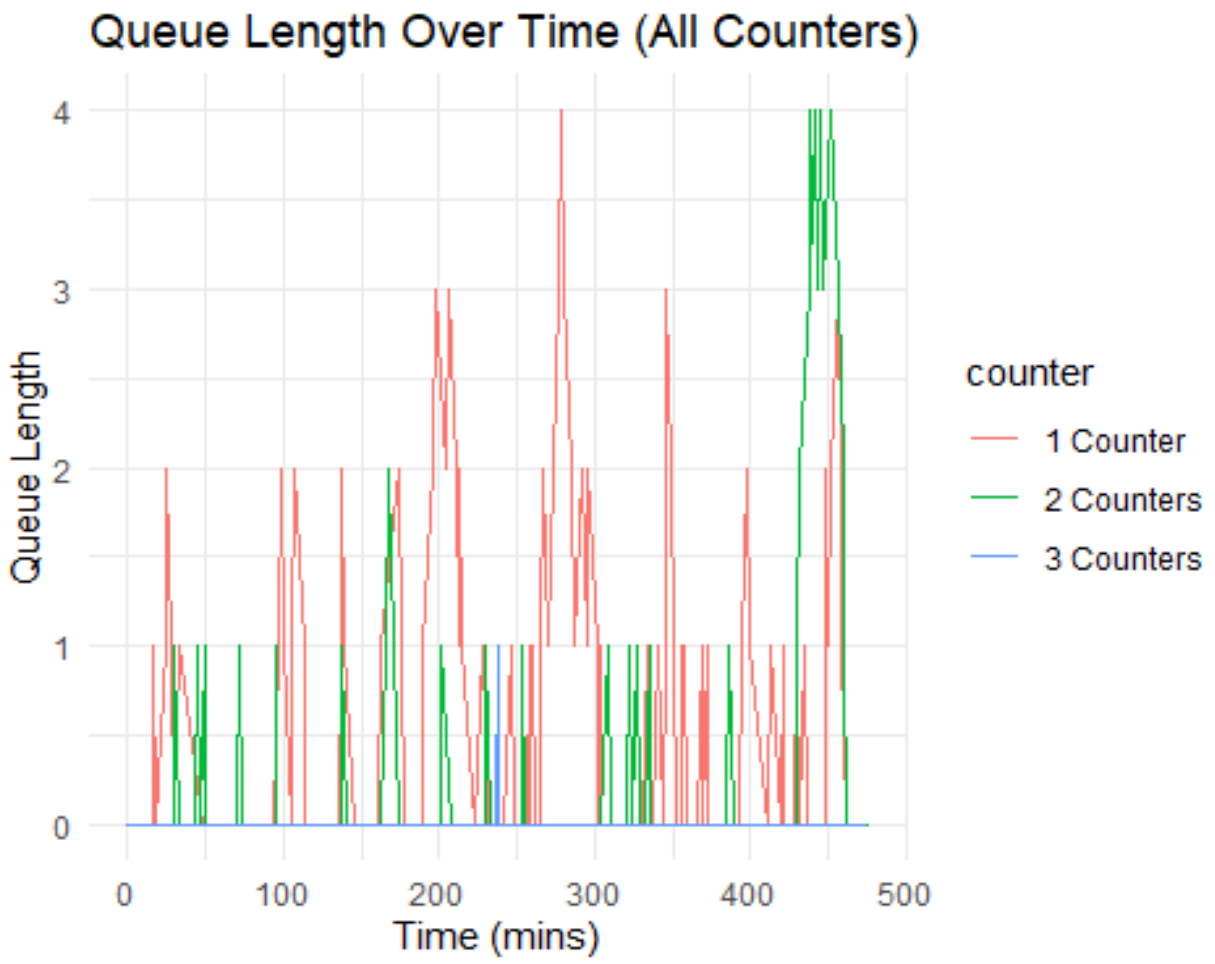
```
# Combined Data for all 3

df_all <- data.frame(
  Counters = c(1, 2, 3),
  Avg_Queue_Length = c(results_counter_1$avg_queue_length,
                       results_counter_2$avg_queue_length,
                       results_counter_3$avg_queue_length),
  Avg_Waiting_Time = c(results_counter_1$avg_actual_wait,
                       results_counter_2$avg_actual_wait,
                       results_counter_3$avg_actual_wait),
  Utilization = c(results_counter_1$utilization,
                  results_counter_2$utilization,
                  results_counter_3$utilization)
)

# Waiting Time with Threshold
ggplot(df_all, aes(x = factor(Counters), y = Avg_Waiting_Time)) +
  geom_bar(stat = "identity", fill = "gray") +
  geom_hline(yintercept = 15, color = "red", linetype = "dashed", size = 1) +
  labs(title = "Avg Waiting Time (All Counters)", x = "Counters", y = "Avg
Waiting Time (mins)") +
  theme_minimal()
```




```
# Utilization
ggplot(df_all, aes(x = factor(Counters), y = Utilization * 100)) +
  geom_bar(stat = "identity", fill = "purple") +
  labs(title = "Utilization  (All Counters)", x = "Counters", y =
"Utilization (%)") +
  theme_minimal()
```




```r
# Queue Length Over Time – All
r1 <- results_counter_1$resource_logs
r1$counter <- "1 Counter"
queue_all <- rbind(r1, r2, r3)

ggplot(queue_all, aes(x = time, y = queue, color = counter)) +
  geom_line() +
  labs(title = "Queue Length Over Time (All Counters)",
       x = "Time (mins)", y = "Queue Length") +
  theme_minimal()
```




```
# --- CONCLUSION ---
cat("\n********** CONCLUSION **********\n")

##
## ********** CONCLUSION **********

if (results_counter_2$avg_actual_wait <= 15)  {
  cat(" 2 counters are ENOUGH.\n Avg wait time is acceptable.\n")
} else {
  cat(" 2 counters are NOT enough. Consider increasing.\n")
}

##  2 counters are ENOUGH.
##  Avg wait time is acceptable.
```

Analysis and Interpretation

The simulation was run for the bank operation with 1,2,3 counters respectively with the given details in the question.

Key performance matrix are summarized below

Metrix	1Counter	2 counter	3 counter
Avg Wait Time (Queue)	4.13 mins	1.45 mins	0 mins
Avg Queue Length	0.7	0.3	0.01
Server Utilization	66.22%	45.38%	33.4 %

Under the current 2-counter system, the average-wait-time is well below the 15-min thresholld, and queue length remains miniimal. Utilization is moderate, suggesting the sysaem handlles currrent demand efficiiently.

an additiional 3rd counter reduces waiting time and queue length even more but causes utilizatiion to falls to 33.4%, indicating potentiial underutiilization of resouirces. Whiile perforrmance is relatiively imprioved, the additiional counter may not be justified under present circumsttances.

2
counters, then, represent the best compromiise between operatiing effiiciency and qualiity of customer serviice. A thiird counter woulld be needed only if future trafffic shows a signifiicant increasse.

Also compared with 1 cuntor in oder to check its performance , which clear state that 1 cuntoer is not enough and have a high queue, which suggest for having another counter

Visualization and Reporting

Visual outputs were generatted to support the analyssis and comparre the systtem under diffferent counter.

A summmary of the key visualls is presentd below:

Visualization	Insights
Bar Chart: Avg Waiting Time	Both 2 and 3 counters stay below the 15-min threshold. 3 counters remove queue completely, but 2 is still sufficient.
Bar Chart: Server Utilization	Utilization drops as counters increase. 2 counters show optimal balance between load and idle time.
Line Chart: Queue Length Over Time	Queue length fluctuates slightly for 2 counters, remains flat for 3. Shows that queues are minimal in both.

Visualizatiions confiirm that 2 countes are suffiicient under curent trafic. The bar chars highligt acceptble waiiting tiimes and server utiilization for 2 counters, and line graps confirm that quaues are small across the simulation.

Introducing a 3rd counter improves the metics but leads to underutlization. Since the marginal gains are not signifiicantly contribuing to customer experiance, the 2-counter setup remains optiimal.

Overall, the visual data is aliigned with performance datia, affirmming the suggestiion to meintain the current setup unless increased future damand requires expansion.