By the Bayes (STA240 Project)

Noah Obuya & Tamya Davidson

2024-05-01

Proposal

Introduction

Welcome to By the Bayes! A restaurant serving food straight from the Louisiana Bayou. What makes our restaurant special is that every step of our process is backed by probabilistic modeling and inference. It's in our name. Today we'll be taking you through the customer's queuing experience in two different scenarios, providing simulations and explanations to back all of our claims.

Analysis Plan

Scenario One

Our first scenario models simple setting with one dining table and one chef, with operating hours 10am - 10pm. Suppose customers arrive according to a Poisson process with a rate of

$$\lambda_A = 5 \text{ per hour, } 40 \text{ total customers from open to close}$$

Once a customer arrives, their total service time (ordering, cooking and eating) can be modeled by an exponential distribution with rate S = 6.

$$\lambda_S = 6$$

To get a sense of the customer queuing experience, we will run multiple simulations of our exponential distribution. The simulation will model the time it takes to service one customer. Essentially, once one customer comes in, we want to know how long it takes for another customer to be serviced. We only have one table and one chef, so we are only able to handle one customer at a time. We will build upon the exponential distribution, providing two models, as we continue to model its efficacy.

Scenario Two

Our second scenario gives us 5 dining tables and

$$L$$

chefs, with operating hours 10am - 10pm. Customers arrive according to a Poisson process with a rate of

$$\lambda_A = 10 \text{ per hour, } 80 \text{ total customers from open to close}$$

Once a customer arrives, their total service time can be modeled by an exponential distribution with rate S = 3L. Each additional chef cuts down the wait time by $\frac{1}{3}$.

$$\lambda_S = 3L$$

We are assuming a customer spends $50 per meal, and that each chef earns a wage of $40 per hour.

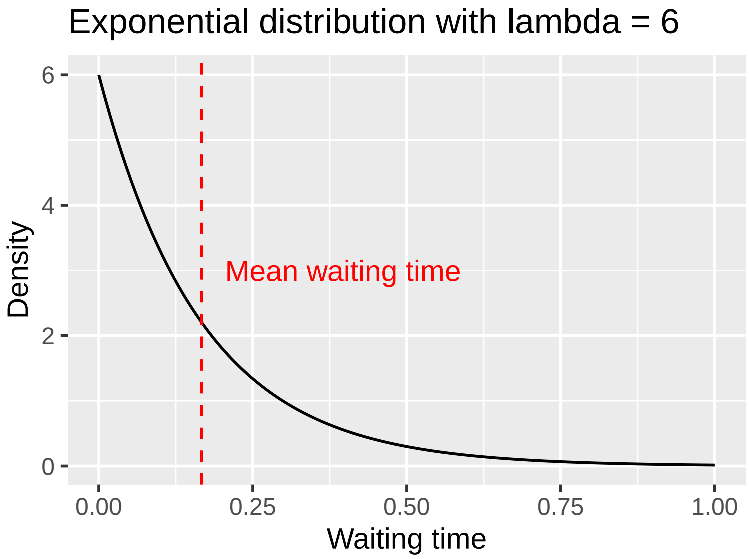
To simulate this scenario, we will build upon the model we created in Scenario One. The main differences being an increase in tables, and the addition of an input variable (number of chefs). To better model the more sophisticated scenario, we will be using the queuing library in R-Studio. The queueing library in R is a package designed to help users model and analyze different types of queueing systems, which are scenarios where tasks, items, or people wait in line to be processed or served. In other words, it helps you understand and simulate situations similar to lines at a grocery store, calls waiting to be answered at a call center, or our very own probabilistic model!

Queueing theory uses a variety of models to describe different systems, and one common way to describe these models is with Kendall's notation (M/M/c). The first M stands for Markovian, meaning memoryless and random, the second M stands for the wait in between service times being modeled by an exponential distribution and the c stands for the amount of servers. In our first model, our Kendall notation would have looked like $M/M/1$ because we only had one server, or in our case, chef. In this model, our c remains a variable because we are trying to find out the best number of chefs to maintain.

After running a simulation using queuing theory, we will create a function that, when given the input of chef number, returns to us average wait time, revenue, wages and of course, profits. This final table will allow us to make a decision on how many chefs we should hire.

Analysis

Scenario One



Part One - The Mean

Figure 1 allows us to explain our mean wait time. Given the PDF (Probability Density Function) of our model, we see that the average wait time is 1/6 of an hour, or 10 minutes. This can also be described by

$$1/\lambda$$

, once again giving us 1/6 of an hour.

Figure 1 showcases the probability density function (PDF) for our basic queuing model's wait times, characterized by an exponential distribution with a mean wait time of $\frac{1}{\lambda} = \frac{1}{6}$ hours, or approximately 10 minutes. This simple model assumes a constant service rate and a single service channel, making it particularly suited for initial planning stages.

However, the exponential distribution's memorylessness property — where the probability of service completion is constant regardless of how long a customer has already waited — does not always align with practical service scenarios. In reality, customer arrivals can be highly variable, and during periods of high demand, this variability can lead to significantly longer wait times than our model suggests.

From the graph, we observe that while many customers may experience wait times close to the mean, a substantial variation is possible, indicative of the potential for both shorter and unexpectedly longer waits. Such insights are crucial for managing customer expectations and improving service delivery strategies. They highlight the need for employing more comprehensive models like the M/M/c system in settings where multiple service points or varied service rates play a critical role in daily operations.

Part Two - First Scenario One Simulation For our first model, we decided to look at the start and end times of each customer's service. The weight time is calculated by the difference between their service start time and their service end time.

Table 1: Average Wait Times for Each Simulation Run

Simulation Run	Average Wait Time (hours)	Average Wait Time (minutes)
1	0.4282327920772	25.693967524632
2	0.43179767708511	25.9078606251066
3	0.405889869983096	24.3533921989857
Average	0.421973446381802	25.3184067829081

The initial simulation attempts to model a single-server queue system where each customer's service and wait time are determined based on exponential distributions for arrivals and services. It assumes random arrivals and service times but lacks detailed dynamics such as the server's operational status (idle or busy) and does not track the length of the queue. This simplistic approach provides basic insights but overlooks several aspects of real-world service environments.

We can demonstrate that this model is missing the mark through observing our means for the queuing simulation

Mean Wait Time in the System (including service time):

$$W = \frac{1}{\mu - \lambda}$$

Mean Wait Time in the Queue (excluding service time):

$$W_q = \frac{\rho}{\mu - \lambda}$$

where

$$\rho = \frac{\lambda}{\mu}$$

represents the server utilization, server utilization being the amount of time a server is "busy". Caculating our two means we get:

$$W = 60 \ minutes$$

and

$$W_q = 50 \; minutes$$

So, clearly, our model is incorrectly averaging means, because it finds that customers are only waiting for an average of 25 minutes ($W = 25 \; minutes$)

This model needs to improve by incorporating more realistic customer behaviors and queue dynamics. Enhancements such as considering customer balking or reneging, and providing metrics for queue length and server utilization, would offer a more comprehensive understanding. Additionally, incorporating variability in arrival and service rates to reflect different operational conditions like peak hours would make the simulation more robust.

Part Three - Second Scenario One Simulation

Table 2: Average Wait Times Improved Simulation

Simulation Run	Average Wait Time (hours)	Average Wait Time (minutes)
1	0.474058159774567	28.443489586474
2	1.24535753790513	74.7214522743075
3	0.72489784123281	43.4938704739686
Average	0.814771179637501	48.88627077825

As we can see, the improved simulation averages much closer to the true mean wait time of $W = 60 \; minutes$. The revised simulation significantly improves upon the first by explicitly calculating the waiting times for each customer, considering the time until the previous customer's service is complete. This approach more accurately reflects actual queue dynamics, where service cannot begin until the previous service has concluded, thus providing a realistic measure of waiting times. Each customer's start time is dynamically adjusted based on the service end time of the preceding customer, which prevents any overlap and simulates a true first-come, first-served queue.

This methodology provides deeper insights into how service efficiency impacts customer waiting experiences, making it a valuable tool for operational planning and optimization. Further improvements could include multi-server configurations, which is what we will tackle next, in Scenario Two.

Scenario Two

In Scenario 2, we used the queueing model package from R. The package allowed us to calculate performance measures such as the average number of customers in the system, the average waiting time, the utilization of servers, and the probability of having a certain number of customers in the system. By simulating different scenarios with varying numbers of chefs (L), we can determine the optimal number of chefs to minimize wait times and maximize profits. The package also allowed us to evaluate the impact of different factors, such as customer arrival rates and service times, on the restaurant's performance.

```r
library(queueing)

simulate_day <- function(L) {
  lambda_A <- 10  # Arrival rate (customers per hour)
  num_tables <- 5  # Number of dining tables
  lambda_S <- 3 * L  # Service rate per hour based on number of chefs

  # M/M/c model where c = number of tables
  # and the service rate is adjusted based on the number of chefs
  Q_mm_c <- NewInput.MMC(lambda = lambda_A, mu = lambda_S, c = num_tables, n = 1000)
  result <- QueueingModel(Q_mm_c)

  # Extract average wait time in the queue (time spent waiting for a table)
  avg_wait_time <- result$Lq / lambda_A * 60  # Convert hours to minutes

  return(list(avg_wait_time = avg_wait_time, utilization = result$U))
}

# Testing with different number of chefs
simulate_day(2)
```

```
## $avg_wait_time
## [1] 0.09082982
##
## $utilization
## NULL
```

```r
simulate_day(1)
```

```
## $avg_wait_time
## [1] 3.920031
##
## $utilization
## NULL
```

The model above uses a queueing system for our restaurant with a varying number of chefs LL. It uses the M/M/c queueing model, where:

- A A is the arrival rate of customers (10 customers per hour),
- **num_tables** is the number of dining tables (5 tables),
- S S is the service rate per hour, which is calculated as 3×L3×L based on the number of chefs.

  By simulating the queueing system with different numbers of chefs, the model helps determine the optimal number of chefs to minimize customer wait times and maximize profits.

```r
library(queueing)

# Function to simulate one day's operation and calculate profits
simulate_day <- function(L) {
  lambda_A <- 10  # Arrival rate (customers per hour)
  num_tables <- 5  # Number of dining tables
  lambda_S <- 3 * L  # Service rate per hour based on number of chefs
  opening_hour <- 10  # Restaurant opening hour
  closing_hour <- 22  # Restaurant closing hour

  # Using M/M/c queue model
  Q_mm_c <- NewInput.MMC(lambda = lambda_A, mu = lambda_S, c = num_tables, n = 1000)
  result <- QueueingModel(Q_mm_c)

  # Calculate hourly revenue based on the minimum of customer arrivals or table capacity
  hourly_revenue <- min(lambda_A, lambda_S * num_tables) * 50  # $50 per customer
  daily_revenue <- hourly_revenue * (closing_hour - opening_hour)  # Calculate hours of operation

  # Calculate daily wages for chefs
  daily_wages <- L * 40 * (closing_hour - opening_hour)  # $40 per hour per chef

  # Calculate profits
  profits <- daily_revenue - daily_wages

  # Get average wait time in minutes
  avg_wait_time <- result$Lq / lambda_A * 60

  return(list(profits = profits, avg_wait_time = avg_wait_time, revenue = daily_revenue, wages = daily_w
}

# Function to find optimal number of chefs
optimize_chefs <- function(max_chefs = 10) {
  results <- data.frame(Chefs = integer(), Profits = numeric(), AvgWaitTime = numeric(), Revenue = numer

  for (L in 1:max_chefs) {
    sim_results <- simulate_day(L)
    results[nrow(results) + 1, ] <- c(L, sim_results$profits, sim_results$avg_wait_time, sim_results$rev
  }

  optimal_chefs <- results[which.max(results$Profits), ]
  return(list(optimal = optimal_chefs, details = results))
}

# Execute the optimization function and print results
optimization_results <- optimize_chefs()
print(optimization_results$details)
```

```
##    Chefs Profits  AvgWaitTime Revenue Wages
## 1      1    5520 3.920031e+00    6000   480
## 2      2    5040 9.082982e-02    6000   960
## 3      3    4560 1.023647e-02    6000  1440
## 4      4    4080 2.095733e-03    6000  1920
## 5      5    3600 6.000825e-04    6000  2400
## 6      6    3120 2.134974e-04    6000  2880
## 7      7    2640 8.847236e-05    6000  3360
## 8      8    2160 4.105378e-05    6000  3840
## 9      9    1680 2.078810e-05    6000  4320
## 10    10    1200 1.128324e-05    6000  4800
```

```
cat("Optimal number of chefs for maximum profits:", optimization_results$optimal$Chefs, "\n")
```

```
## Optimal number of chefs for maximum profits: 1
```

This model simulates the daily operation of a restaurant to determine the optimal number of chefs that maximize profits while considering customer wait times. By varying the number of chefs, the model calculates the restaurant's revenue, chef wages, and profits for each scenario. It uses the M/M/c queuing model to simulate customer arrivals, service times, and queue lengths. The model then identifies the number of chefs that results in the highest profits.

Discussions & Limitations

Conclusion

References

Lab 10 Rossetti, R. (2024, April 4). Sakai.duke.edu. https://sakai.duke.edu/access/content/group/2e691ccc-405a-4982-bf81-42137192402e/Labs/lab10_sol.html

Simulation of Exponential Distribution using R Hernández, C. (2020, September 29). Simulation of Exponential Distribution using R. RPubs. https://rpubs.com/carlosehernandezr/Exponential-Distribution#:~:text=The%20exponential%20distribution%20can%20be,deviation%20is%20also%201%2Flambda