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.

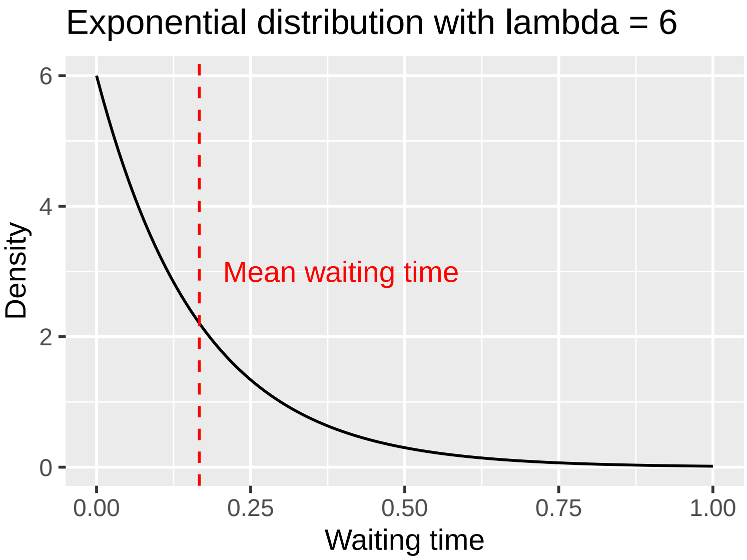
To simulate this scenario, we will expand upon the model developed in Scenario One with increased tables and an additional input for the number of chefs. We'll use the queueing library in R-Studio, designed to model and analyze queueing systems where tasks or people wait to be processed. This allows us to simulate real-world queuing situations like our own@

Queueing theory employs various models to represent systems, commonly described using Kendall's notation (M/M/c), where 'M' stands for Markovian (memoryless and random), and 'c' represents the number of servers. Our initial model was an M/M/1, indicating a single server or chef. In the current model, 'c' varies as we explore the optimal number of chefs needed.

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 highlights the mean wait time as indicated by our model's Probability Density Function (PDF), where the average wait time is shown as $\frac{1}{\lambda} = \frac{1}{6}$ hours, or approximately 10 minutes. This model, ideal for initial planning, assumes a constant service rate and single service channel but its memorylessness—constant probability of service completion regardless of waiting time—may not align well with variable real-world customer arrivals.

The observed variability in wait times from the graph, ranging from shorter to unexpectedly longer, illustrates the limitations of the exponential distribution and the importance of employing more comprehensive models like M/M/c in complex settings. This adaptation is crucial for effectively managing customer expectations and optimizing service delivery strategies.

Part Two - First Scenario One Simulation

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

Table 3: Average Wait Times predicted by Chef Number

Number of Chefs	Average Wait Time (Hours)
1	3.9200314
2	0.0908298
3	0.0102365
4	0.0020957
5	0.0006001
6	0.0002135
7	0.0000885
8	0.0000411
9	0.0000208
10	0.0000113

As the number of chefs L increases, the service rate λ_S multiplies accordingly, which enhances the restaurant's ability to manage customer queues effectively. This directly reduces the average wait time, thereby potentially increasing customer satisfaction and retention. However, diminishing returns are likely as the number of chefs increases beyond the point where service rate far exceeds the arrival rate of new customers, leading to unnecessary labor costs without corresponding increases in customer throughput. I would say, at this stage of analysis, we reach that point past 6 chefs.

Table 4: Optimal Number of Chefs

Chefs	Profits	AvgWaitTime	Revenue	Wages
1	5520	3.9200314	6000	480
2	5040	0.0908298	6000	960
3	4560	0.0102365	6000	1440
4	4080	0.0020957	6000	1920

Chefs	Profits	AvgWaitTime	Revenue	Wages
5	3600	0.0006001	6000	2400
6	3120	0.0002135	6000	2880
7	2640	0.0000885	6000	3360
8	2160	0.0000411	6000	3840
9	1680	0.0000208	6000	4320
10	1200	0.0000113	6000	4800

From a business perspective, the optimal number of chefs L should be selected to balance the cost of hiring additional staff against the generated revenue from serving more customers efficiently. The profits calculated suggest that maximizing profits involves not merely increasing the number of chefs to reduce wait times but also considering the point where additional chefs no longer contribute to significant increases in customer throughput or revenue. At this stage of analysis, one could go with two perspectives, pure profit or customer experience. For the purposes of pure profit, it is clear that one chef takes the cake. Yet, the customer wait time for one chef is exponentially higher than all the others, detailing the stark need for more than one chef. Two chefs gets you the perfect balance of profit and wait time. $\frac{9}{100}$ of an hour is only 5.4 minutes, which is a very reasonable time to wait for food.

```r
library(queueing)

# Function to simulate a day and calculate downtime
simulate_day_with_downtime <- function(L) {
  lambda_A <- 10  # Customers arriving per hour
  num_tables <- 5  # Number of available tables
  lambda_S <- 3 * L  # Service rate, dependent on number of chefs

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

  # Calculate downtime as the proportion of time when chefs are underutilized
  expected_customers_served_per_hour <- min(lambda_A, lambda_S)
  expected_idle_chefs_per_hour <- max(0, L - expected_customers_served_per_hour / 3)
  downtime_proportion <- expected_idle_chefs_per_hour / L

  # Check if utilization is set in the result, otherwise set to 0
  utilization <- ifelse(is.null(result$U), 0, result$U)

  return(list(downtime_proportion = downtime_proportion, utilization = utilization))
}

# Simulating different numbers of chefs and recording downtime
chef_values <- 1:10
downtime_results <- numeric(length(chef_values))
utilizations <- numeric(length(chef_values))

for (i in seq_along(chef_values)) {
  sim_results <- simulate_day_with_downtime(chef_values[i])
  downtime_results[i] <- sim_results$downtime_proportion
  utilizations[i] <- sim_results$utilization
}

# Create a table of results
```

```r
downtime_table <- data.frame(
  "Number of Chefs" = chef_values,
  "Downtime Proportion" = downtime_results)

names(downtime_table)[1] <- "Number of Chefs"
names(downtime_table)[2] <- "Downtime Proportion"

# Use kable from knitr to create a nicely formatted table
library(knitr)
kable(downtime_table, format = "markdown", caption = "Proportion of Downtime and Utilization by Number
```

Table 5: Proportion of Downtime and Utilization by Number of Chefs

Number of Chefs	Downtime Proportion
1	0.0000000
2	0.0000000
3	0.0000000
4	0.1666667
5	0.3333333
6	0.4444444
7	0.5238095
8	0.5833333
9	0.6296296
10	0.6666667

```r
names(downtime_table)[1] <- "Number of Chefs"
names(downtime_table)[2] <- "Downtime Proportion"
```

One final thing that is essential to factor in is downtime for the number of chefs we possess. Without proper downtime, no cleaning, breaks, or just time to catch a breath can occur. This will lead to decreased levels of customer satisfaction when they see dirty counter tops or encounter peeved chefs. With downtime as an added variable of interest, we see that 2 chefs is no longer the optimal number. Only having two chefs allows for no downtime. *Four chefs* seems to be the sweet spot, with 10 minutes of downtime (for all chefs collectively) per hour. That amount also maximizes profits the most and ensures a very speedy customer wait time.

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