STA240 Final Project

Anthony Zhao, Abby Li, William Yan

Introduction

In 2024 alone, the restaurant industry was predicted to reach $1 trillion in sales; however, with rising costs and increased competition, owners must find innovative ways to keep up profit (National Restaurant Association, 2024). With increased technology, many restaurants are turning to data to provide insights into operations. Therefore, in this project, we will be analyzing performance metrics in three different scenarios to inform our decision on profit maximization. The goal for this project is three-fold: to utilize probabilistic modeling to optimize restaurant operations, to analyze customer behavior to maximize consumer satisfaction, and to provide business recommendations by considering key performance metrics such as waiting times, the number of chefs, and more. In the following scenarios, we assume the restaurant operates between the hours of 10am - 10pm.

Scenario 1

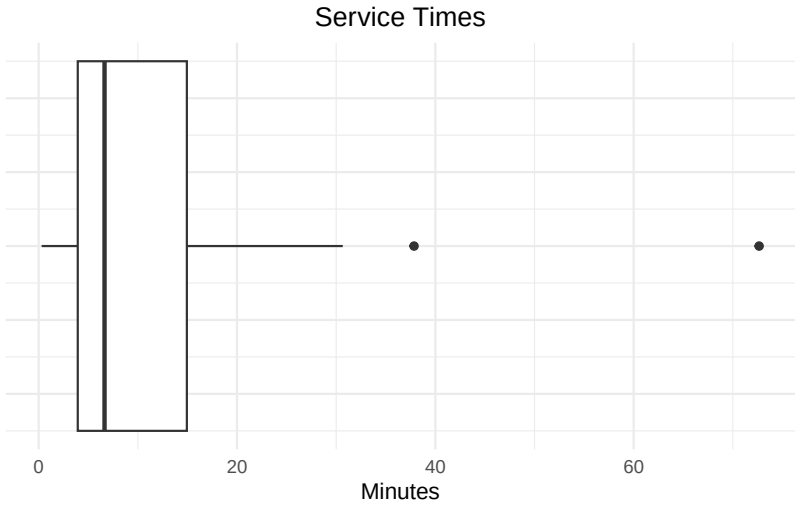
The first scenario considers a setting with 1 table and 1 chef. Customer arrival will be modeled to $W_k \sim Pois(\lambda)$ with $\lambda = 5$ and customer service time will be modeled by $S_k \sim Exp(\lambda)$ where $\lambda = 6$.

- T_k: Arrival time of the kth customer
- W_k: Time between the $k-1$th arrival and the kth arrival

$$W_k = T_k - T_{k-1}.$$

Arrival Times

In this simulation, on average, the number of customers arriving is 5 per hour, so around every 12 minutes. The number of customers that will be arriving within the operating hours is 60, with the first customer arriving 15 minutes after opening at 10:15 AM and the last customer arriving 67 minutes before closing at 8:54 PM.



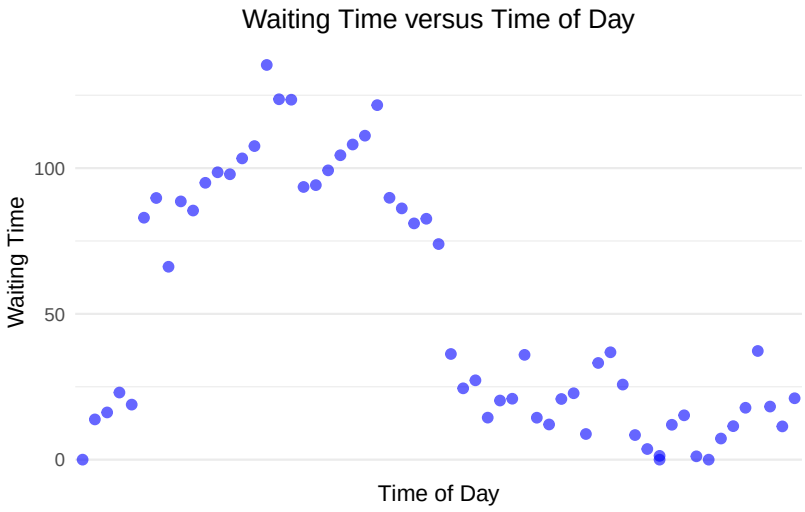

```
        Min       Q1   Median       Q3      Max
1 0.2995134 3.941274 6.641874 14.94373 72.64854
```

The median service time is 7 minutes, with the shortest service time being 0 and the longest service time being 73, which appears to be an outlier. Overall, the data is skewed right, consistent with an exponential distribution, and this indicates that service times tend to be quick.


```
  Min       Q1   Median       Q3     Max       SD
1   0 14.40915 30.18981 90.74413 135.378 42.37374
```

The median waiting time is 30 minutes, with the lowest waiting time being 0 and the highest waiting time being 135. Overall, waiting times tends to be fairly symmetric with a slight positive skew and there are no visible outliers. Wait times are spread out with times being, on average, 42 minutes from the mean of 51 minutes.

	Customer	Service_Length	Service_Start	Service_End	Waiting_Time	Time
1	1	18.322653	15.48044	33.80309	0.00000	10:15
2	2	3.611812	33.80309	37.41490	13.80485	10:20
3	3	23.350001	37.41490	60.76490	16.19654	10:21
4	4	6.156491	60.76490	66.92140	23.01963	10:38
5	5	72.648544	66.92140	139.56994	18.88113	10:48
6	6	11.892555	139.56994	151.46249	82.98774	10:57
7	7	2.496301	151.46249	153.95880	89.74933	11:02
8	8	25.231974	153.95880	179.19077	66.16318	11:28
9	9	4.531018	179.19077	183.72179	88.58002	11:31
10	10	15.649534	183.72179	199.37132	85.44248	11:38



Overall, waiting times appear to be highest around noon or lunch when customer satisfication is lowest. Waiting times decrease steadily 3 pm onwards until closing, suggesting a less busy dinner time where the restaurant has more down time. In our simulation, the maximum service time, 73 minutes, occured an hour before the rush hour, and as a result, waiting times increased significantly after this customer. Although overall service times are fairly quick, the restaurant should do their best to ensure service times are just as fast or even faster during the busy hours.

Scenario 2

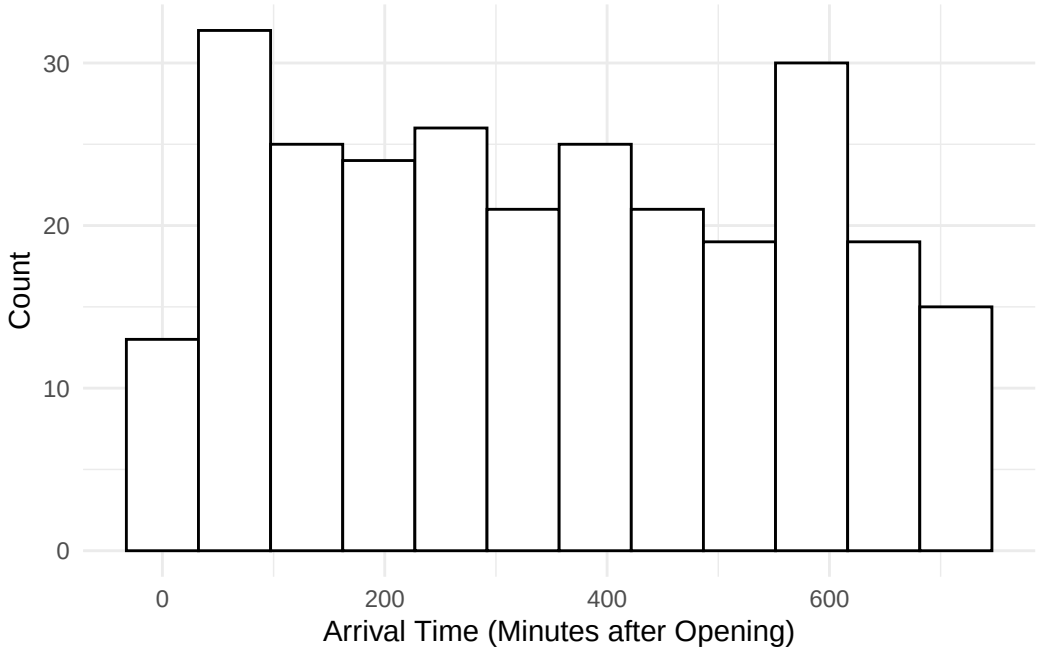
Assumptions:

1. 5 dining tables and L chefs with operating hours 10am - 10pm. We choose here that $L = 2$
2. each table only seats one customer
3. service time modeled by an exponential distribution with rate $S = 3L$, so that the more chefs there are, the faster the service times become
4. 24 customers arrive every hour

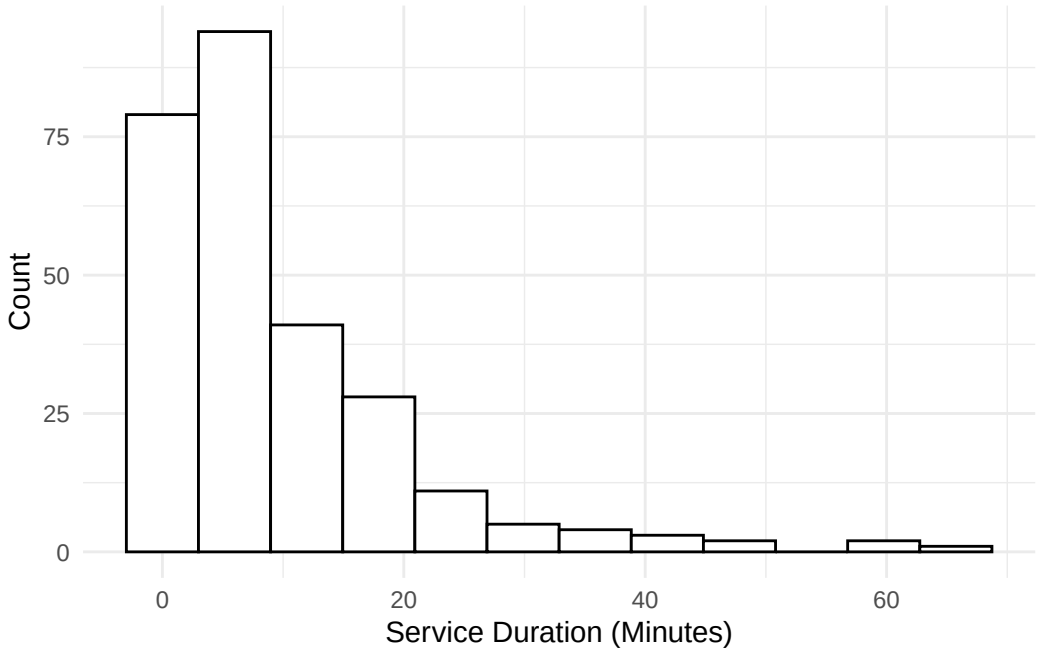
Waiting Times

To model waiting times, we iterate through the day minute by minute.

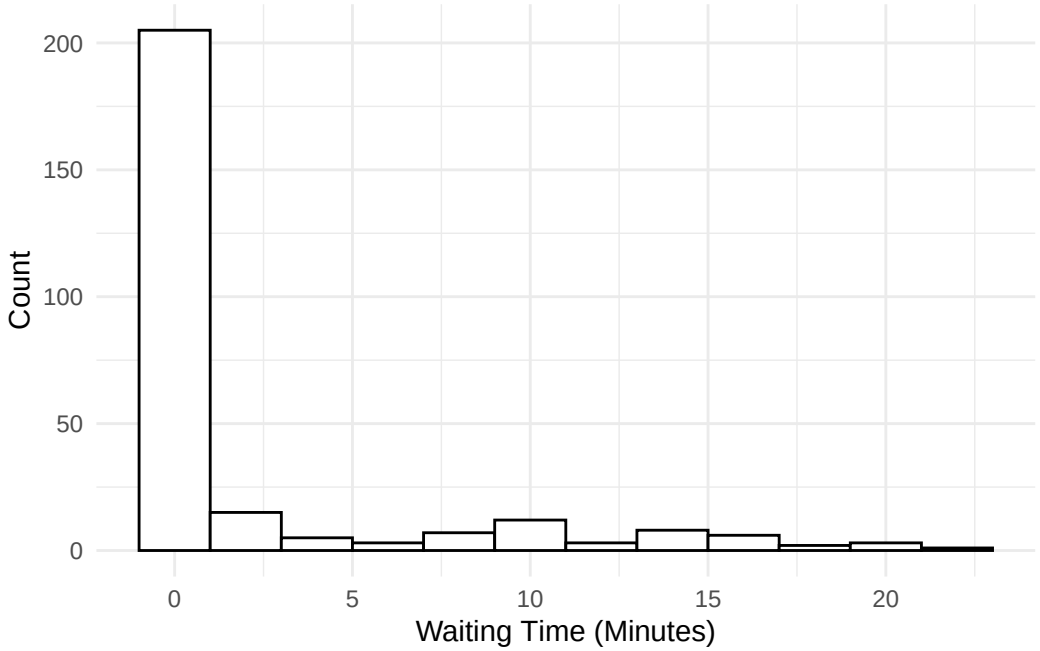
```
scen2_sim_results_by_customer |>
  ggplot(aes(x = arrival_time)) +
  geom_histogram(bins = 12, color = "black", fill = "white") +
  labs(
    x = "Arrival Time (Minutes after Opening)",
    y = "Count"
  ) +
  theme_minimal()
```



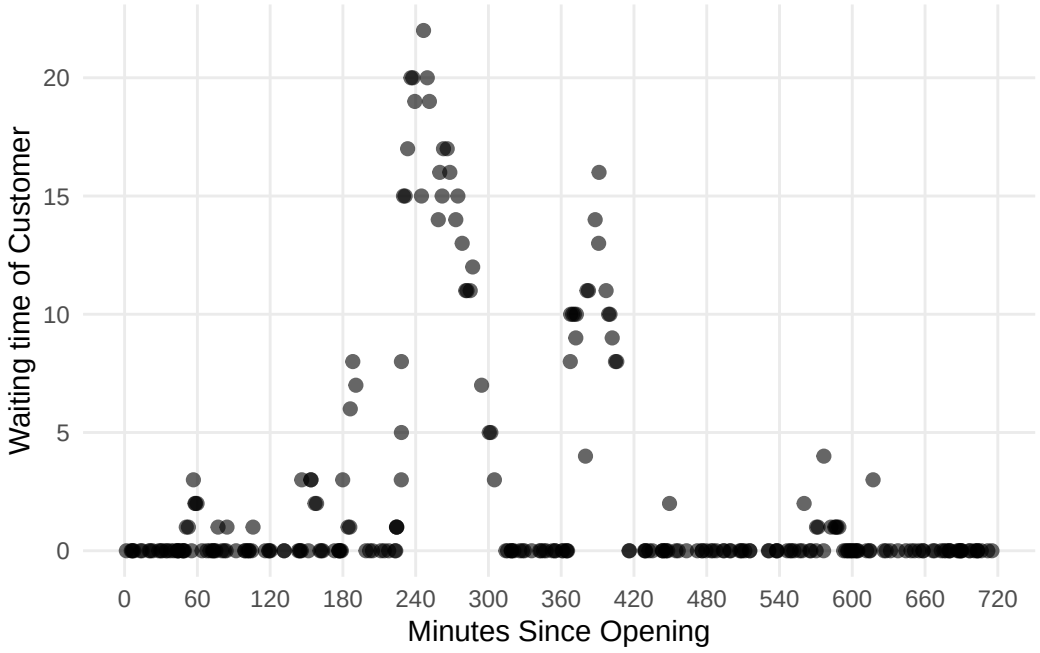

```
scen2_sim_results_by_customer |>
  ggplot(aes(x = service_length)) +
  geom_histogram(bins = 12, color = "black", fill = "white") +
  labs(
    x = "Service Duration (Minutes)",
    y = "Count"
  ) +
  theme_minimal()
```



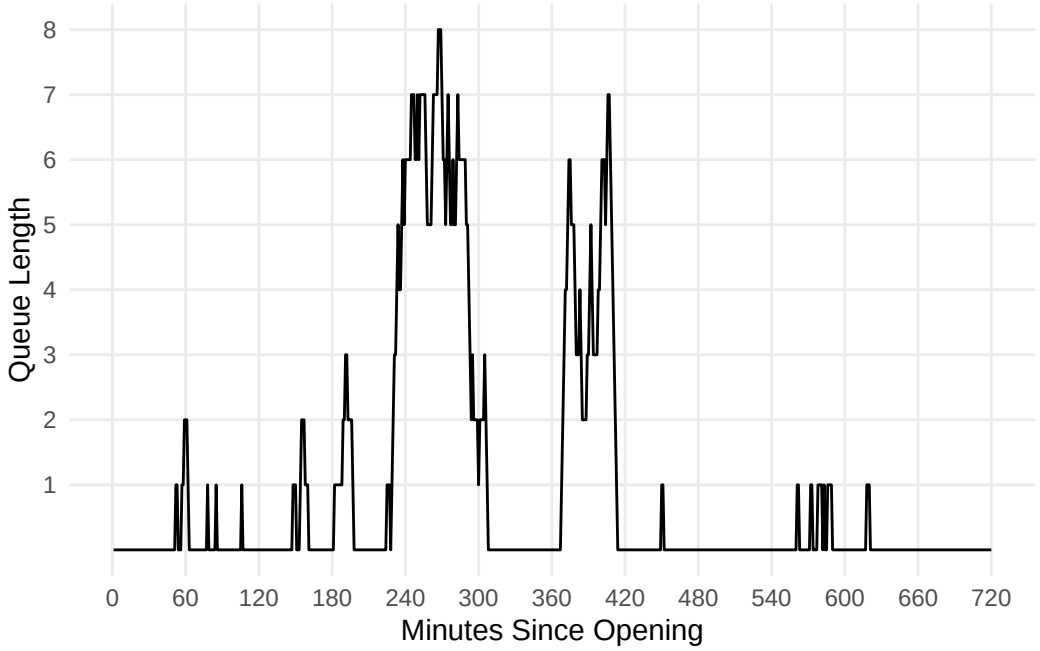

```
scen2_sim_results_by_customer |>
  ggplot(aes(x = waiting_time)) +
  geom_histogram(bins = 12, color = "black", fill = "white") +
  labs(
    x = "Waiting Time (Minutes)",
    y = "Count"
  ) +
  theme_minimal()
```



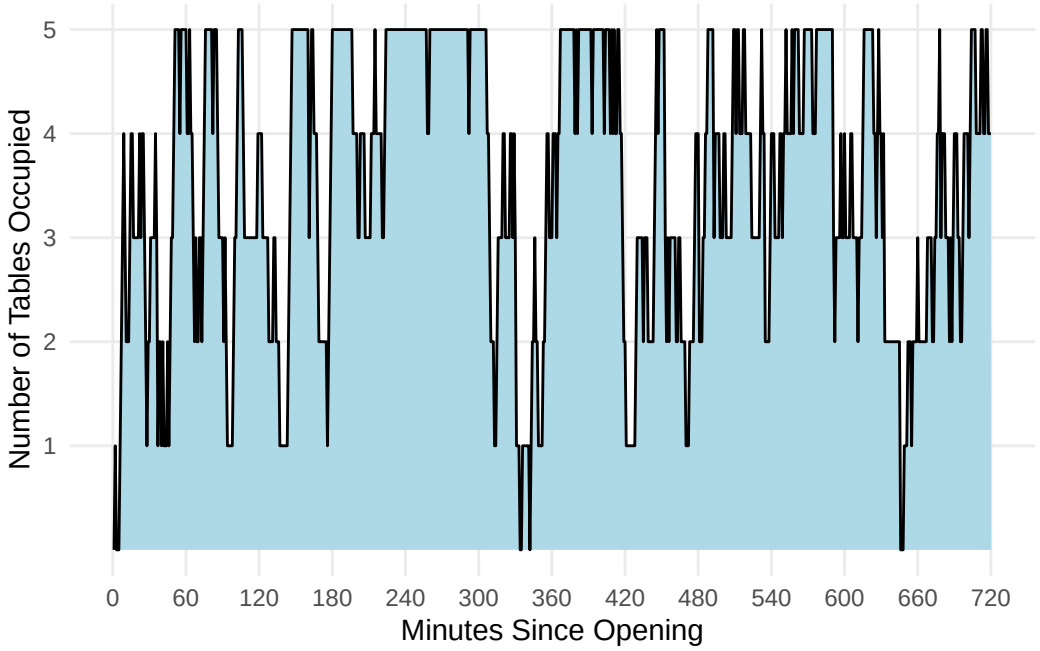
```
scen2_sim_results_by_customer |>
  ggplot(aes(x = arrival_time, y = waiting_time)) +
  geom_point(size = 2, alpha = 0.6) +
  scale_x_continuous(breaks = seq(0, as.numeric(total_time_s2), by = 60)) +
  labs(
    x = "Minutes Since Opening",
    y = "Waiting time of Customer"
  ) +
  theme_minimal() +
  theme(panel.grid.minor = element_blank())
```




```
scen2_sim_results_by_minute |>
  ggplot(aes(x = minutes_since_opening, y = queue_size)) +
  geom_line() +
  scale_y_continuous(breaks = seq(1, max(sim_s2$queue_size_history), by = 1)) +
  scale_x_continuous(breaks = seq(0, as.numeric(total_time_s2), by = 60)) +
  labs(
    x = "Minutes Since Opening",
    y = "Queue Length"
  ) +
  theme_minimal() +
  theme(panel.grid.minor = element_blank())
```




```
# OCCUPIED TABLES
scen2_sim_results_by_minute |>
  ggplot(aes(x = minutes_since_opening, y = occupied_tables)) +
  geom_area(fill = "lightblue") +
  geom_line() +
  scale_y_continuous(breaks = seq(1, num_tables_s2, by = 1)) +
  scale_x_continuous(breaks = seq(0, as.numeric(total_time_s2), by = 60)) +
  labs(
    x = "Minutes Since Opening",
    y = "Number of Tables Occupied"
  ) +
  theme_minimal() +
  theme(panel.grid.minor = element_blank())
```



Restaurant Profits

Assumptions:

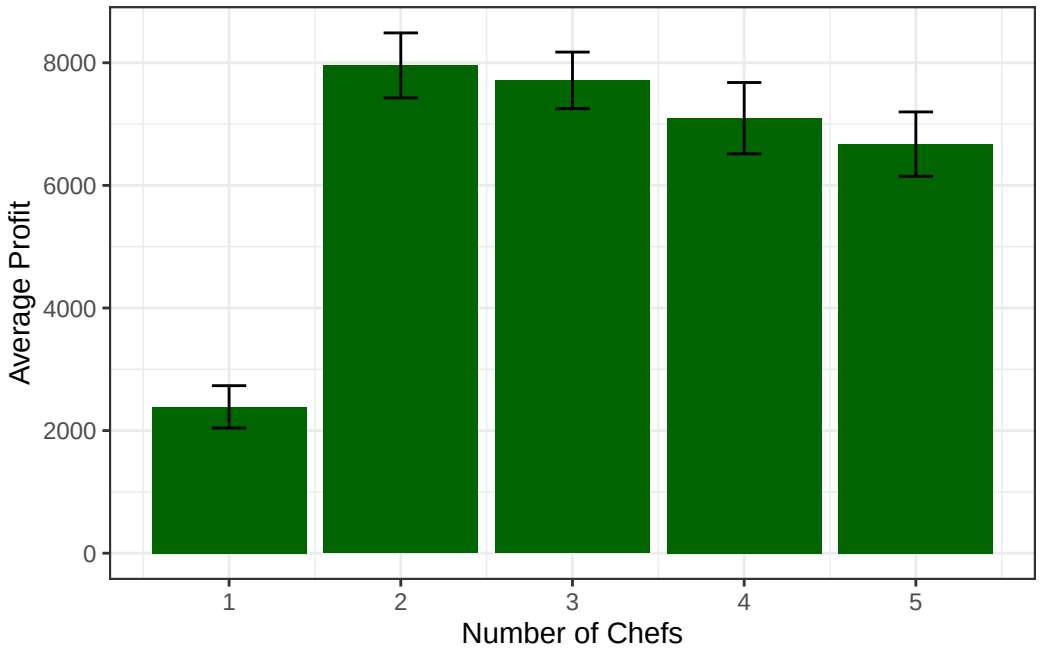
1. each customer spends $50 per meal (customers who are still in the queue when the restaurant closes won't pay)
2. each chef earns a wage of $40 per hour (paid for the entire duration of the restaurant's operating hours)
3. Each table cost $1000 per day (extra service cost, rent, etc.)
4. For customers who waited more than 30 minutes, they earn the restaurant half the amount of customers who didn't.

Maximizing Profits

With 5 tables, 24 customers arriving per hour, and these dollar amounts, how many chefs should we hire? We will run our simulation 100 times with 1 to 5 chefs on staff, to see which will maximize the expected profit.

	total_customers	profit	num_chefs	num_tables	avg_waiting_time	long_waits
1	288	1995	1	5	187.55208333	277
2	282	7180	4	5	0.10638298	0
3	282	6700	5	5	0.00000000	0
4	264	7240	2	5	3.57196970	0
5	288	7480	4	5	0.05555556	0
6	288	2320	1	5	164.96527778	264
7	284	7280	4	5	0.04577465	0

```
8             288   7960         3          5       0.56250000          0
  avg_queue_length max_queue_length avg_tables_occupied
1       75.02083333              129            4.966667
2        0.04166667                3            2.023611
3        0.00000000                0            1.495833
4        1.30972222               12            3.588889
5        0.02222222                2            1.806944
6       65.98611111              117            4.925000
7        0.01805556                2            2.134722
8        0.22500000                4            2.783333
```

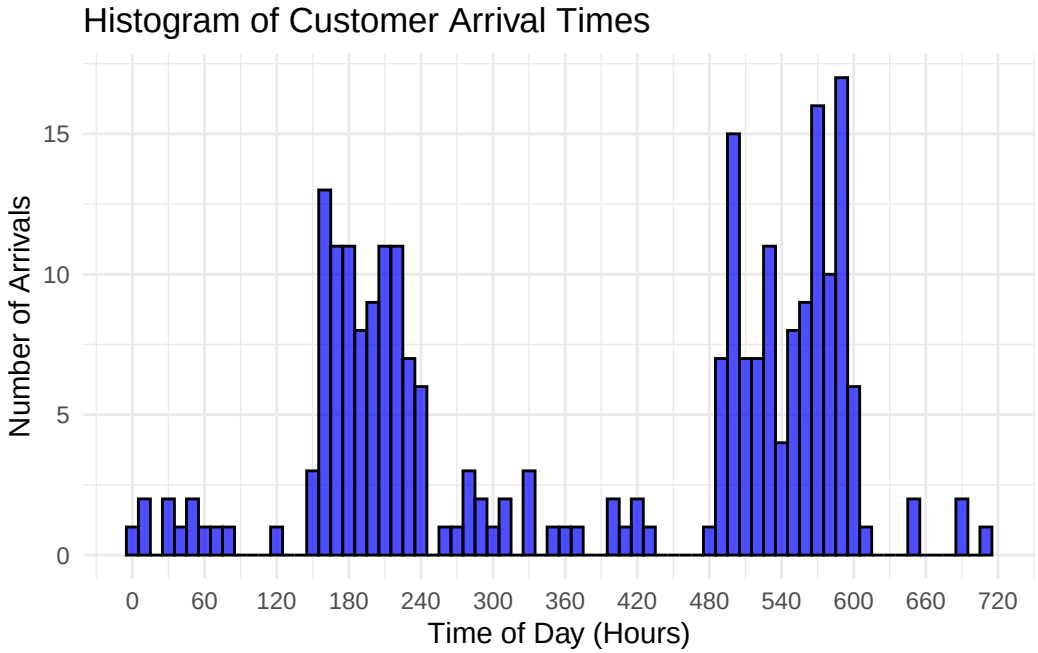


Scenario 3

To make the simulation more realistic, we have a third scenario.

Assumptions: 1. Open at 10am, close at 10pm 2. From 12pm to 2pm and 6pm to 8pm, 60 customers arrive every hour. Otherwise, 6 arrive every hour. 3. Instead of simulating service times with Exp( ) where = 3 times the number of chefs, we do = ln(chefs + 1), so that additional chefs beyond 2 make more of an impact. 4. Each customer will sit for a minimum of 45 minutes. This flat value will be added to the simulated service time, and is unaffected by staffing. 5. In the profit calculation, there is a cost of adding additional tables (which are now variable), which is $40 per table. 6. Chefs still cost $40 per hour to hire, and each customer earns $50.

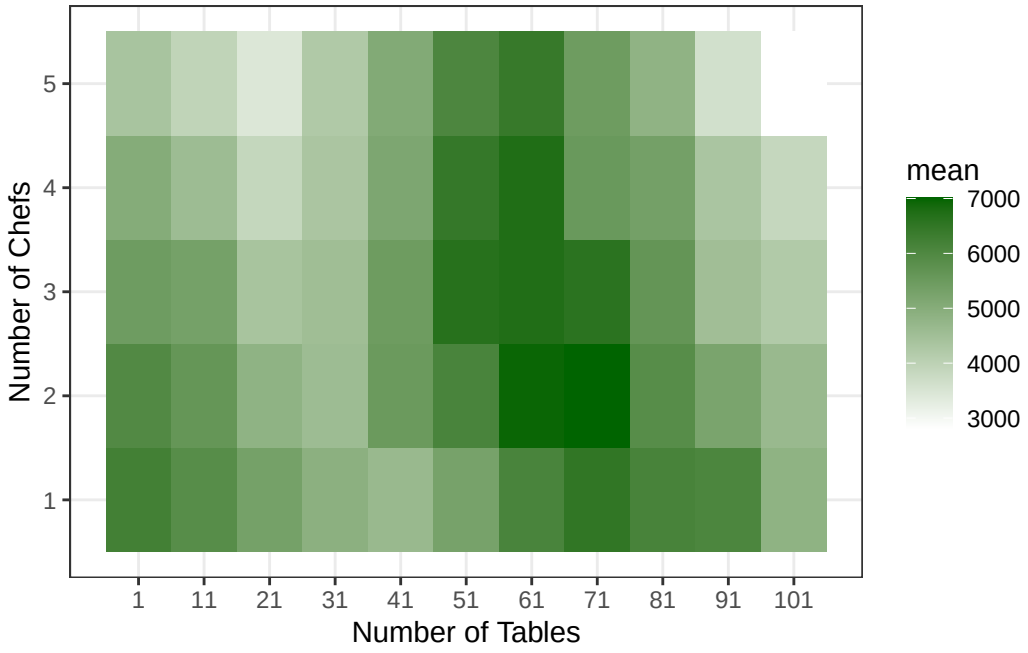
Arrival Times



Maximizing Profits

Under this scenario, how can we maximize profits?

```
`summarise()` has grouped output by 'num_chefs'. You can override using the
`.groups` argument.
```



```
`summarise()` has grouped output by 'num_chefs'. You can override using the
`.groups` argument.

# A tibble: 10 x 4
# Groups:   num_chefs [5]
   num_chefs num_tables mean_profit    sd
       <dbl>      <dbl>       <dbl> <dbl>
 1         2         71       7009. 1171.
 2         2         61       6948.  978.
 3         4         61       6735  1041.
 4         3         61       6730  1088.
 5         3         51       6652.  491.
 6         3         71       6595   868.
 7         1         71       6504.  462.
 8         4         51       6448.  567.
 9         5         61       6419. 1131.
10         1          1       6225   474.

`summarise()` has grouped output by 'num_chefs'. You can override using the
`.groups` argument.

# A tibble: 3 x 5
# Groups:   num_chefs [2]
  num_chefs num_tables profit mean_waiting_time    sd
      <dbl>      <dbl>  <dbl>             <dbl> <dbl>
1         2         71  7009.              3.63  1.93
2         2         61  6948.             10.5   3.72
3         4         61  6735               5.32  2.54

`summarise()` has grouped output by 'num_chefs'. You can override using the
`.groups` argument.

# A tibble: 3 x 5
# Groups:   num_chefs [2]
  num_chefs num_tables profit mean_long_waits    sd
      <dbl>      <dbl>  <dbl>           <dbl> <dbl>
1         2         71  7009.            1.75  3.68
2         2         61  6948.           38.3  25.3
3         4         61  6735             6.3  12.3

`summarise()` has grouped output by 'num_chefs'. You can override using the
`.groups` argument.

# A tibble: 3 x 5
# Groups:   num_chefs [2]
  num_chefs num_tables profit mean_queue_length    sd
```

```
      <dbl>      <dbl>  <dbl>             <dbl> <dbl>
1         2         71  7009.              1.43 0.829
2         2         61  6948.              4.13 1.75
3         4         61  6735              2.09 1.12
```

```
`summarise()` has grouped output by 'num_chefs'. You can override using the
`.groups` argument.
```

```
# A tibble: 3 x 5
# Groups:   num_chefs [2]
  num_chefs num_tables profit mean_max_queue    sd
      <dbl>      <dbl>  <dbl>          <dbl> <dbl>
1         2         71  7009.           17.6  5.63
2         2         61  6948.           28.4  5.48
3         4         61  6735            19.8  6.12
```

```
`summarise()` has grouped output by 'num_chefs'. You can override using the
`.groups` argument.
```

```
# A tibble: 3 x 5
# Groups:   num_chefs [2]
  num_chefs num_tables profit mean_occupied    sd
      <dbl>      <dbl>  <dbl>         <dbl> <dbl>
1         2         71  7009.          36.1  2.92
2         2         61  6948.          35.4  3.60
3         4         61  6735           30.5  2.56
```