# Pokémon Alfresco Restaurant

## Simulation Tutorial

Team Members: Nicole, Faisal, Jake and Brandon

# Project Objective

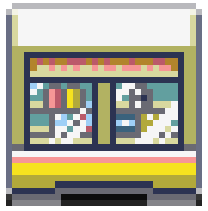
To **simulate a queueing system** of a fictional Pokémon Alfresco Restaurant in order to determine the **optimal number of tables** to have in order **to achieve maximum profits**

# Elements of the simulation

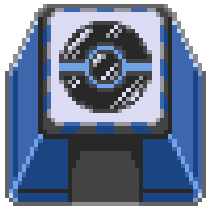
**Types of customers**

**Cashier**

**Drink dispenser**



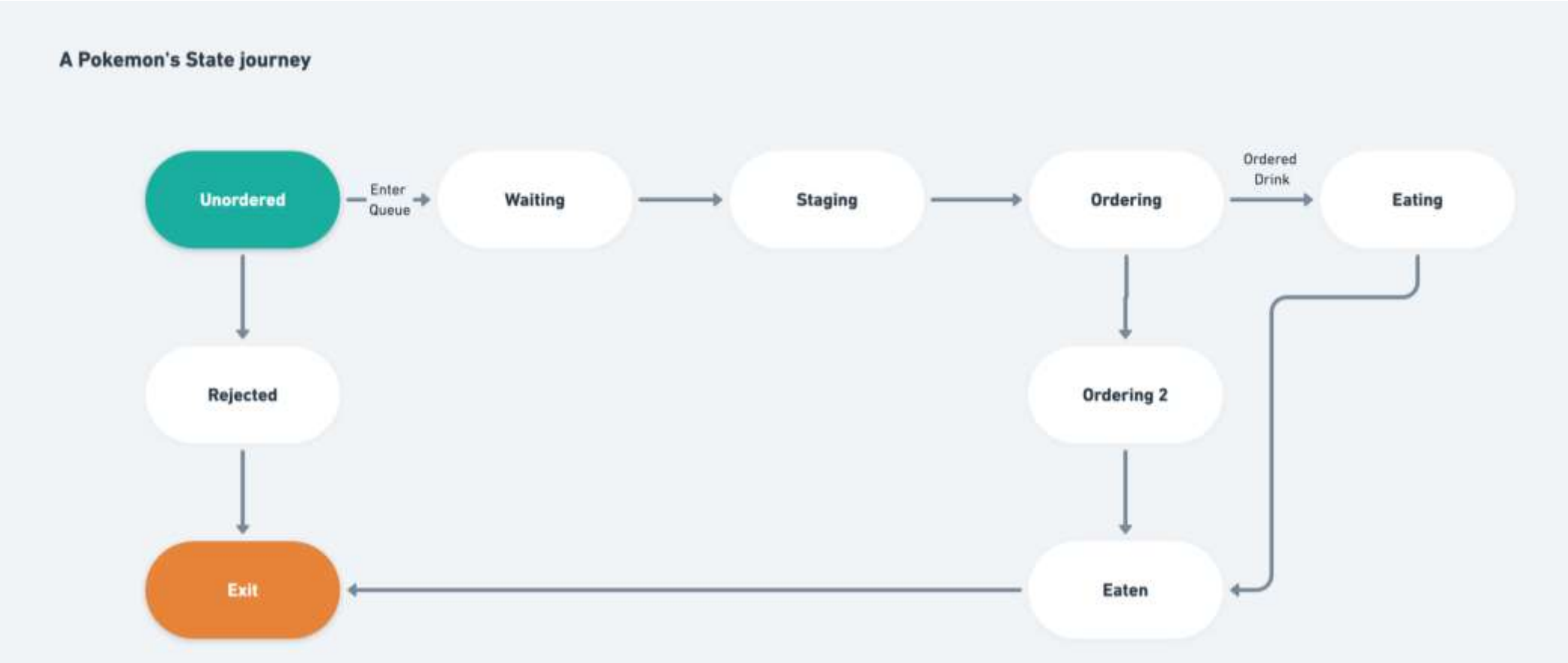
**Entrance**



**Table**

**Chair**

# Flow of the simulation



# Unordered


This is where new customer **arrive** through the entrance.

State: ‘UNORDERED’

# Waiting


New customers start queueing up to order their food.

State: ‘UNORDERED’ to ‘WAITING’

# Staging


A customer here is the **next one to be served**

State: ‘WAITING’ to ‘STAGING’

# Ordering


A customer here starts **ordering** from cashier Jenny

State: ‘STAGING’ to ‘ORDERING’

# Ordered → Eating or Ordering2


Having **ordered**, the customer is then randomly assigned one of two cases.

They could either go straight to their table to start **eating** OR grab an **additional drink** from the dispenser

State: 'ORDERING' to 'ORDERED'

# Case 1: Eating


In this case, the customer goes directly to their seat to start **eating**

State: ‘ORDERED’ to ‘EATING’

# Case 2: Ordering2


Alternatively, the customer could proceed to the **drinks dispenser** for a drink before heading to their seat to start **eating**

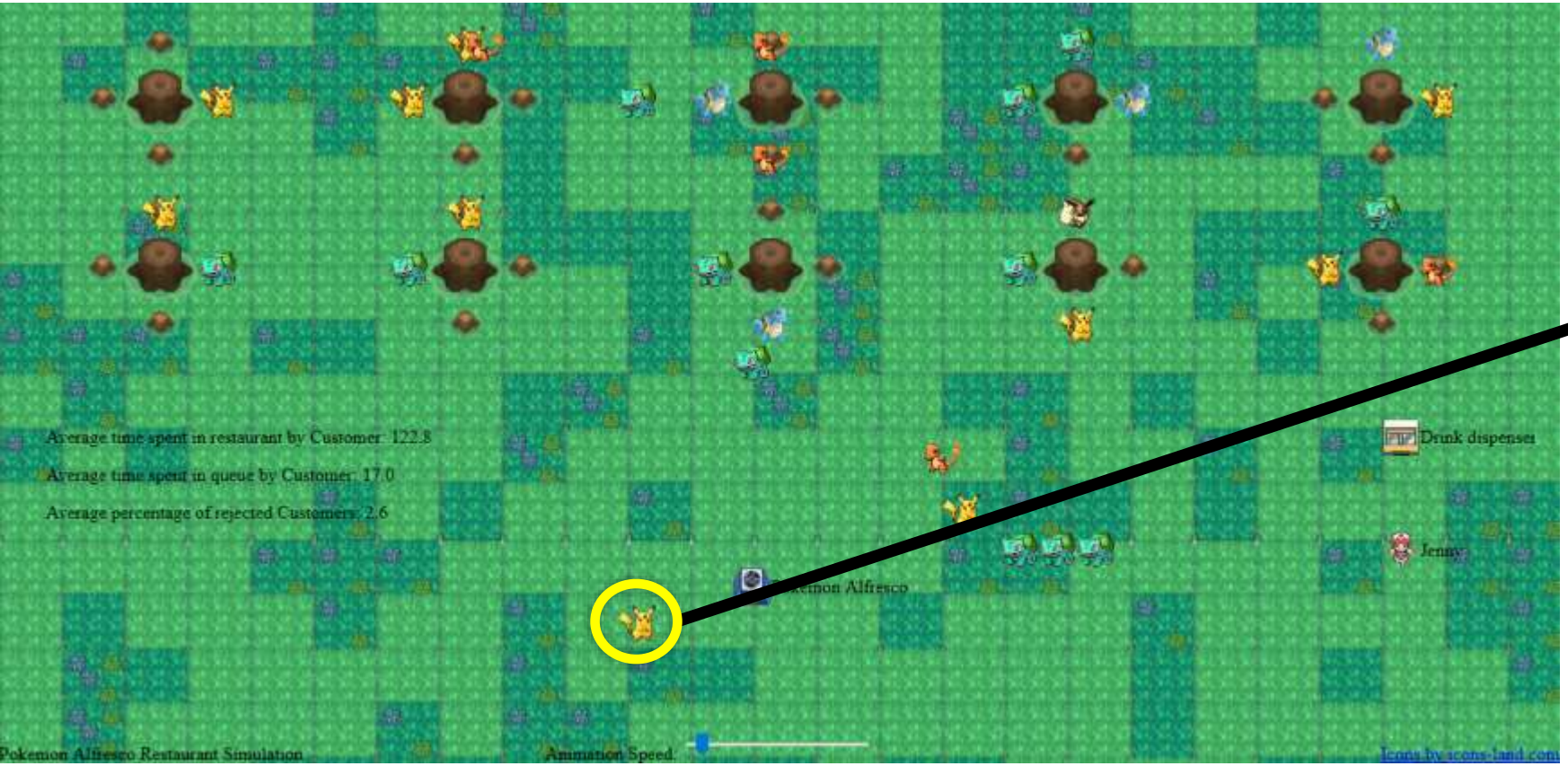
State: ‘ORDERED’ to ‘ORDERING2’ to ‘EATING’

# Eaten→ Exited


Having **eaten** their meals, customer would then **exit** the restaurant.

State: ‘EATING’ to ‘EATEN’ to ‘EXITED’

# Rejected

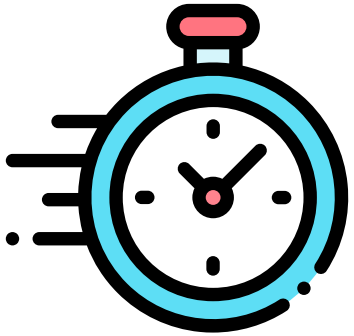



Whenever the maximum capacity of the restaurant is reached, any newly arrived customer will be **rejected**

State: ‘EXITED’

# Statistics of interest

Here are some of the statistics we are generated from our simulation:



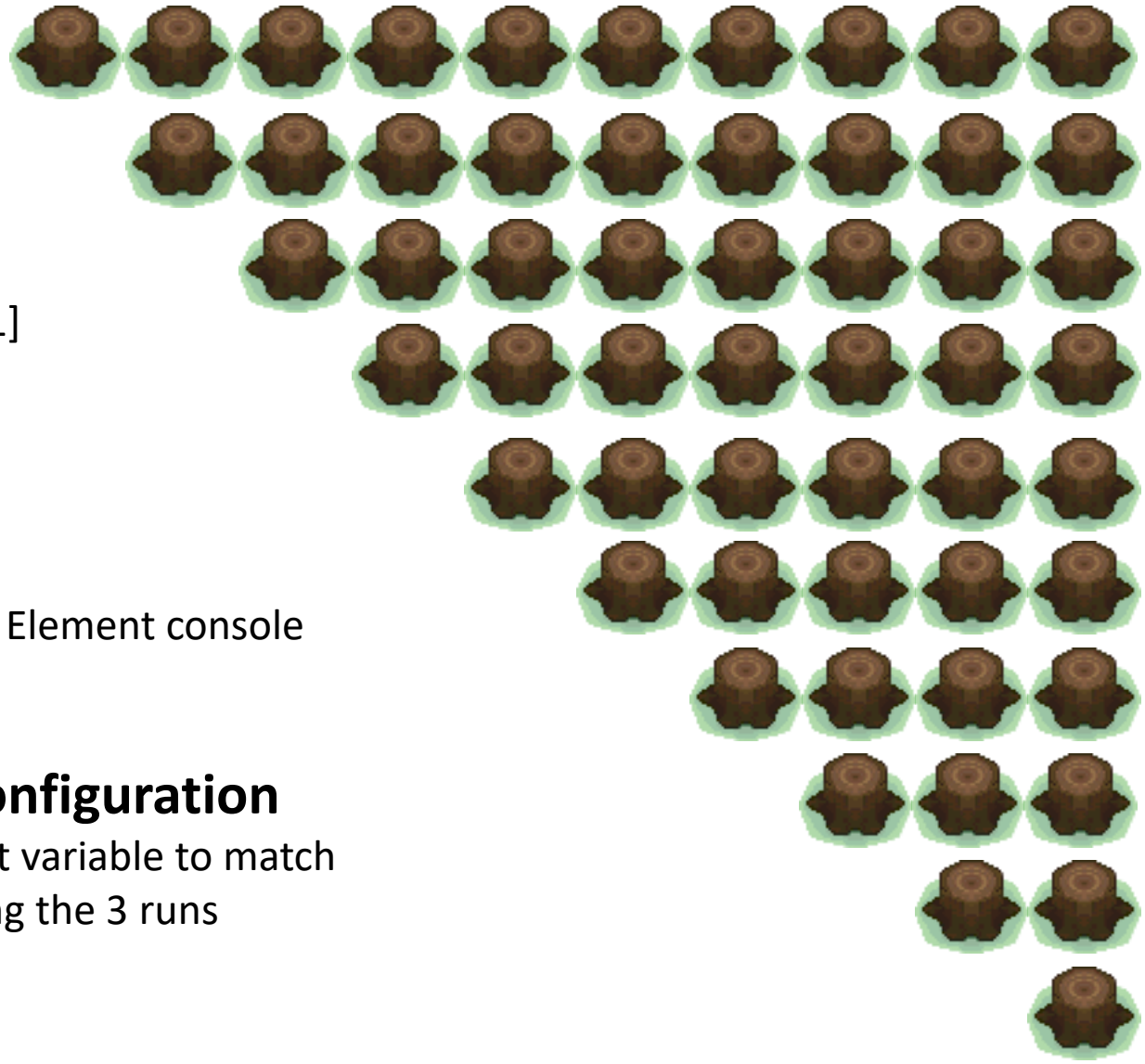
Average time spent in restaurant

Average queueing time

Percentage of customers lost due to maximum capacity

Customer count

# Data collection process



**Step 1: Initial conditions:**

- # of Tables: 10
- Max customerInRestaurant= [(# of tables x 4) + 1]
- e.g 10 tables → 41 customers max

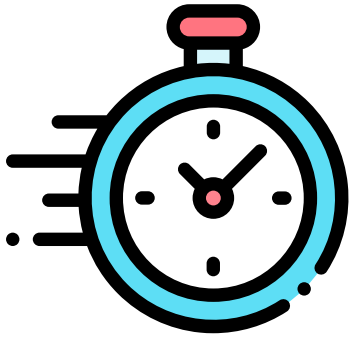
**Step 2: Perform simulation:**

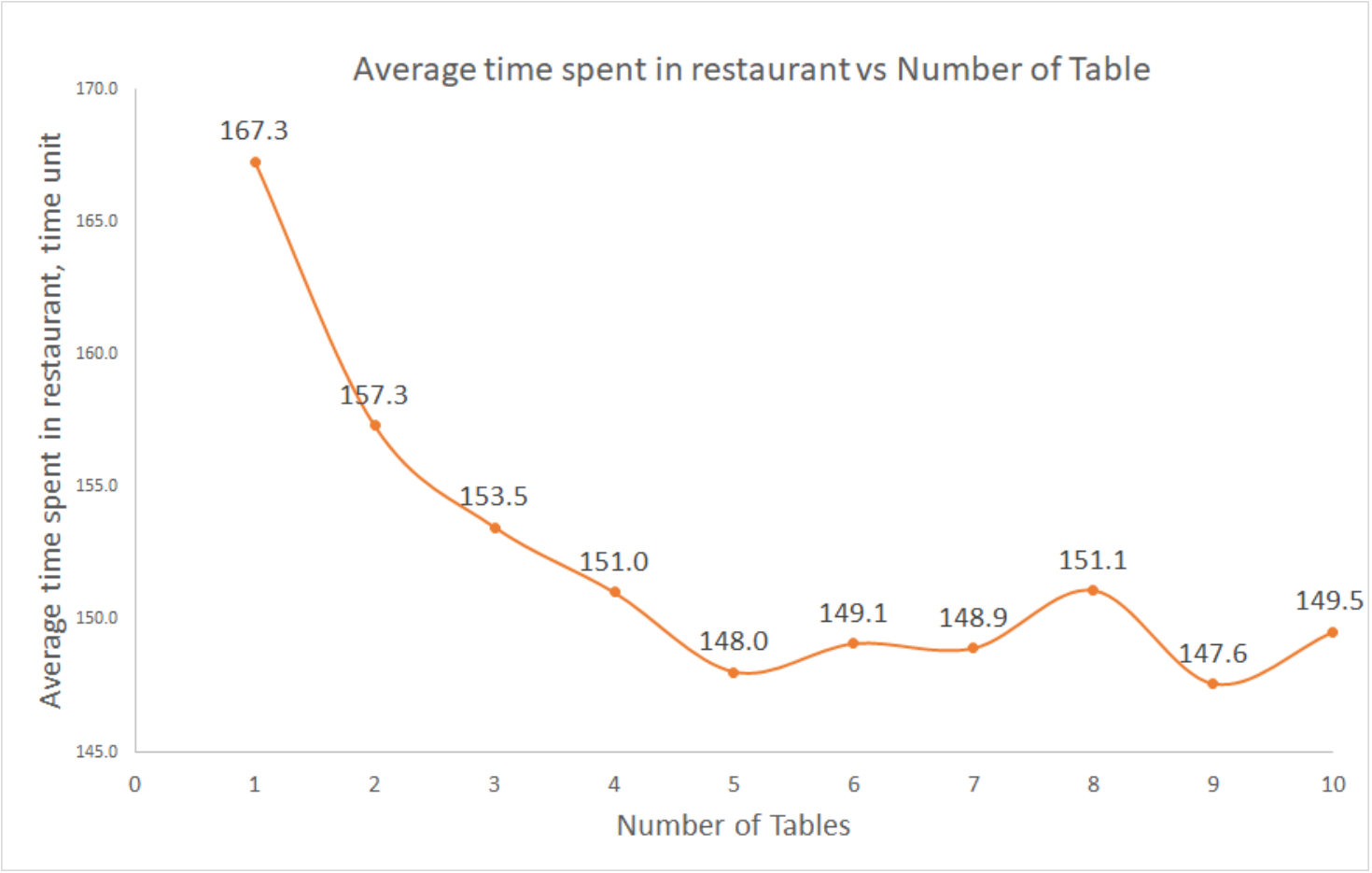
- 3 runs
- Collect the respective statistics from the Inspect Element console
- Tabulate in EXCEL

**Step 3: Repeat step 2 for next table configuration**

- Remember to change the customerInRestaurant variable to match the updated number of tables before performing the 3 runs

# Findings - Average Time spent in restaurant

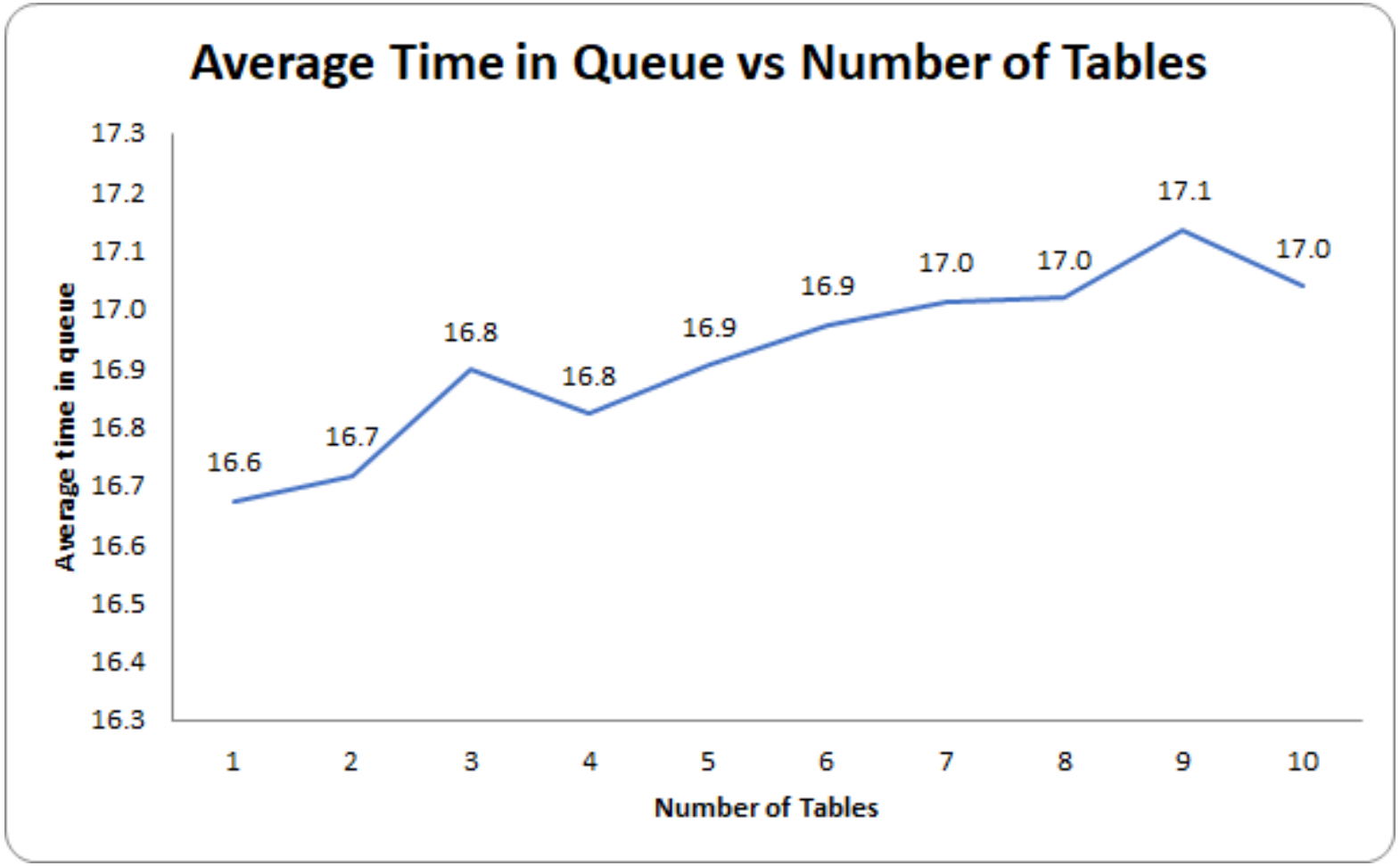





Observations

As the number of tables increases, we could see a general decrease in the time spent by customers in the restaurant

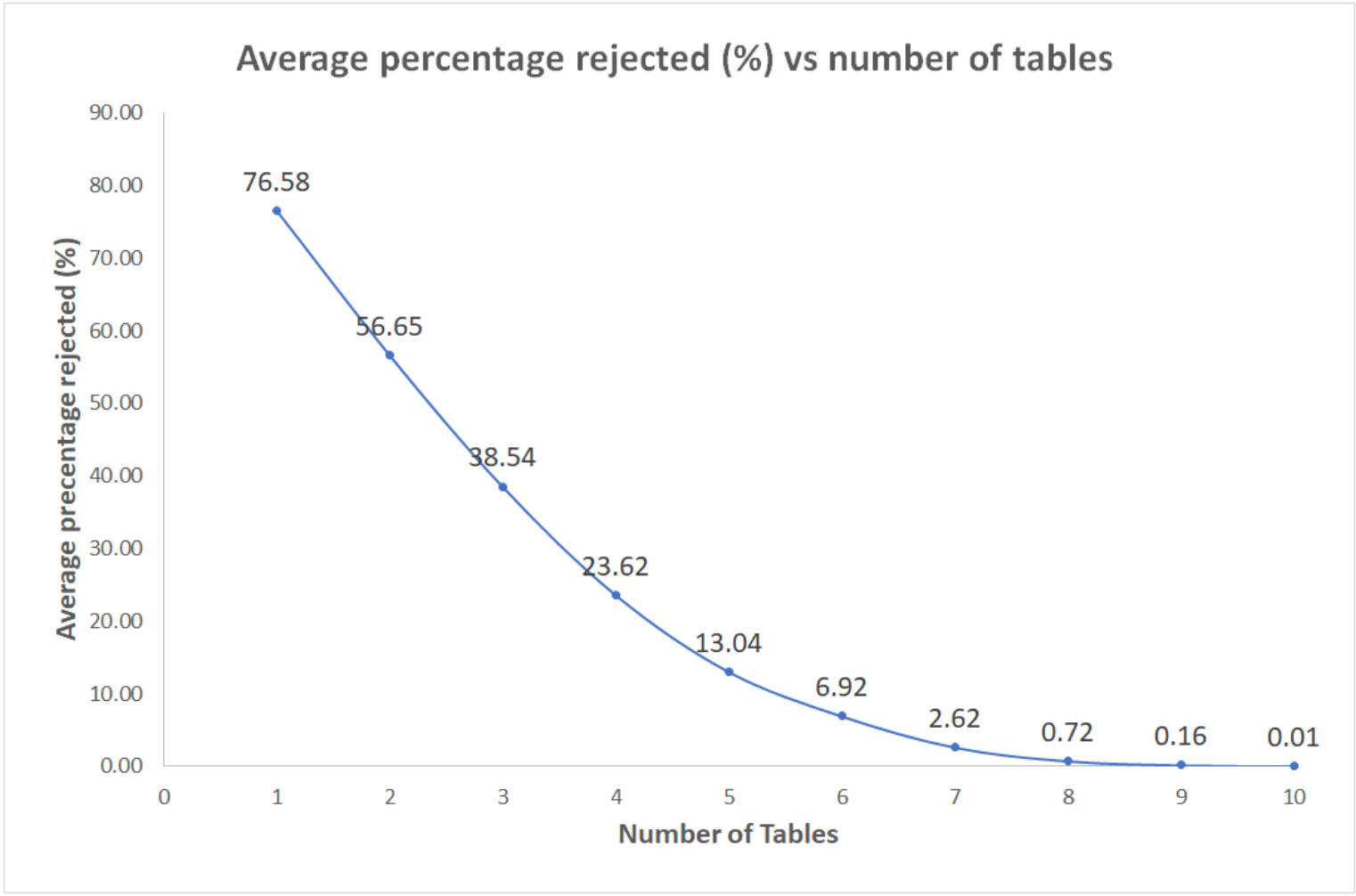
# Findings – Average Queueing Time




Observations

As the number of tables increases, we can see that there is a general increasing trend in terms of the average queueing time

# Findings - Average Percentage Rejected

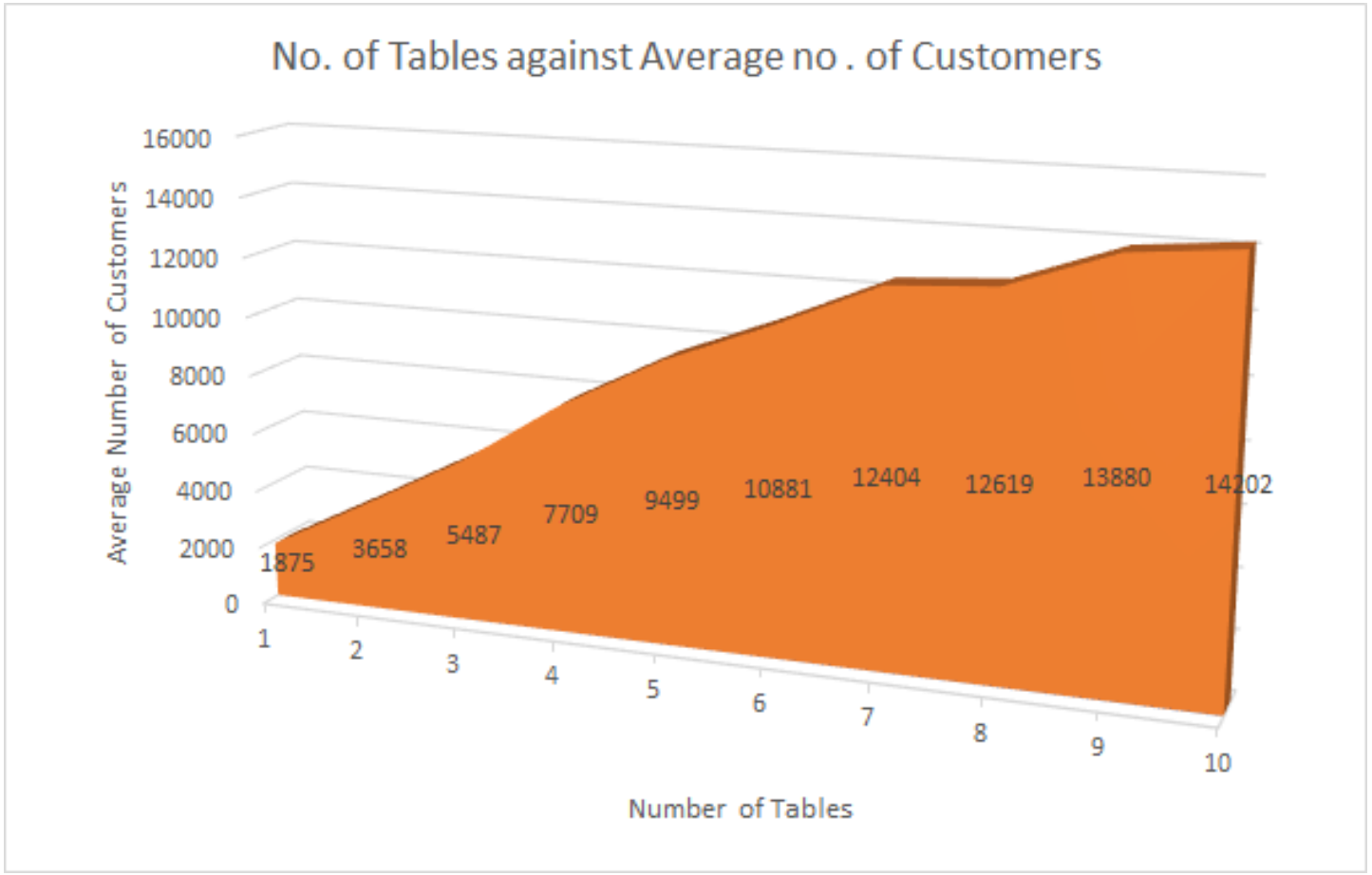



Observations

As the number of tables increases, we see that the average percentage rejected decreases.

This is expected as more tables = more capacity to cater to the inflow of customers

# Findings – Customer Count




Observations

As the number of tables increases, we see that the average number of customers also increases.

This makes sense as more tables means more capacity to cater to more customers