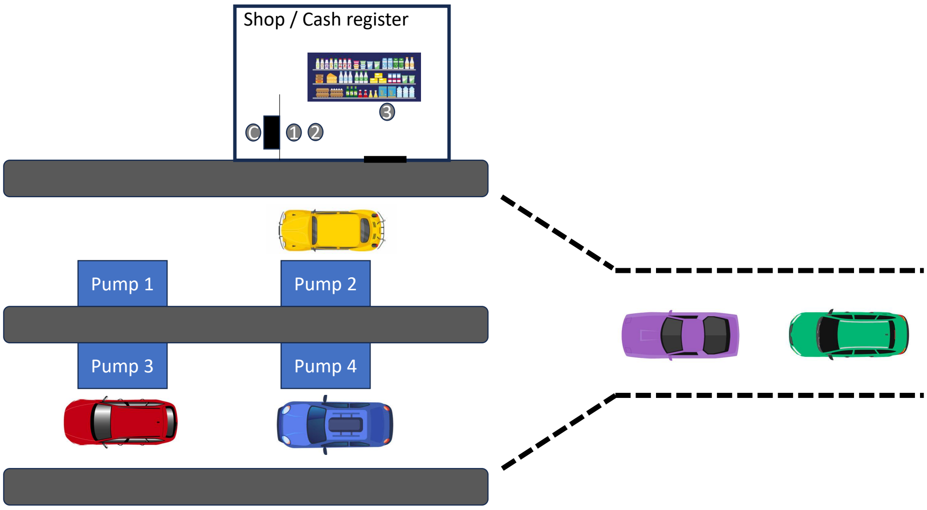
Queueing at a Gas Station

The owner of a gas station notices that the queue of cars waiting in front of the gas station becomes quite long, regularly. She would like to improve this situation, for example by adding another cash register, or by adding more gasoline pumps. To determine what the best action would be, she hires a consultancy company (that's you!) to help her analyse the data she collected and to build a mathematical model for this queueing system.

The model

The figure below gives a graphical, schematic representation of the queueing dynamics at the gas station. There are four fuel pumps, each with their own parking spot. Cars arrive at the gas station and form a single queue. As soon a free spot is available, the car at the head of the queue parks at this available pump and the driver fills up the tank with fuel (which takes a random time F). After fueling up the car, the driver goes to the shop. Inside the shop, with probability p_S the driver wants to buy something before joining the queue to pay. With probability $1 - p_S$, the driver immediately joins the queue to pay for the fuel. Each payment takes a random time P. Picking items from the shop before joining the queue takes a random time S. For simplicity, we ignore the time required to walk from the car to the shop and the time to walk back to the car and drive away.



Blocking. Since the corridors in front of the gas station are quite narrow, there can be no overtaking at all. This may result in blocking: when the parking spots next to Pump 2 and Pump 4 are occupied, the car at the head of the queue cannot move forward, even if the parking spot next to Pump 1 or Pump 3 becomes available. In the figure above, the purple car cannot drive to the free spot next to Pump 1. Instead it has to wait until the yellow car leaves its spot (or the red and blue cars leave). Similarly, if the driver of the blue car has finished paying, but the driver of the red car is still in the shop, then the blue car is also blocked.

To complicate things even further, some cars really insist on parking on a particular side (left or right), depending on the location of their fuel cap. In more detail: a fraction p_L (resp. p_R) of the arriving vehicles will only accept a free, accessible spot if it is available on left (resp. right) side. Otherwise, they will block the cars behind them, even if they would be willing to park on the other side. We assume that there is no room to overtake, anywhere in this model. So suppose that the purple car would insist on parking on their left side (i.e. pump 3 or 4), it would block the green car, even if the yellow car would leave.

The data

The opening hours of the gas station are from 6h00 until 22h00. The problem owner collected data during *one* day. During this day, for each customer, the arrival time was recorded as well as the service time at the Fuel pump, the time spent in the shop, and the time required to pay (again, this is only the service time, excluding waiting time). Customers that did not buy anything from the shop have a service time of zero registered at the shop. Finally, each driver has been asked whether they had a strict parking preference (left or right) or not. Using all these data, it should be possible to estimate the model parameters. As indicated before, we ignore the time required to walk from the car to the shop and back, and the time to drive away, so those times are not recorded. The service time at the fuel pump includes the time to drive the car to a parking spot.

Your task

Your first task is to analyse and the data and interpret the data. Each group has their own dataset. Fit distributions on the inter-arrival times and the service times of the various stations in the queueing model and report summary statistics. Find estimates for all the model parameters. Your second task is to implement a discrete-event simulation to simulate this model. For each of the three stations (Fuel, Shop, Pay), you should simulate the mean and standard deviation of the queue length (including customer(s) in service) and the mean and standard deviation of the waiting times. You should also simulate the mean and standard deviation of the total time spent in the system and give a 95% confidence interval for the mean time spent in the system. In your report, include a table like this (in the appendix) filled with the correct simulation results:

Performance measure	**Mean**	**Standard deviation**
Waiting time Fuel station Queue length Fuel station		
Queue length shop		
Waiting time Payment queue Queue length Payment queue		
Total time spent in the system		

We ask you to do this *twice*, including the above table for each of these two cases:

1. Run the simulation for one day with the *exact* input values given to you in your dataset. This means that each arrival time should be exactly at the moment specified in the dataset, and that the service times at each of the visited stations (and the parking preference) should be exactly like in the provided dataset.
2. Run the simulation with randomly sampled arrival times and service times, based on the model parameters and distributions estimated and fitted from the dataset. Do multiple simulation runs, so you can create confidence intervals for (some of) the performance measures.

We cannot stress enough that the setup of your program and the approach should be similar to the programs shown during the lectures and in the lecture notes: a continuous-time simulation with a future event set and event scheduling. If you strongly believe that a different approach would be more suitable, please consult one of the lecturers first and motivate your request to deviate from the standards we have been teaching. You are *not* allowed to use any libraries that have not been discussed during the lectures. In particular, you are *not* allowed to use SimPy.

Scenarios

Besides the standard scenario, which is the situation described above, the problem owner would like you to analyse **two out of the three** following scenarios:

1. No Blocking: by broadening the lanes, cars no longer become blocked by their predecessor. This means that empty parking spots can always be filled, unless there is no vehicle in the queue that wants to use that particular spot (for example,

because the only empty spot is on the right and all cars in the queue insist on parking in the left lane).

2. Replacing the shop by self-service payment terminals: In the new situation, the shop will be completely closed. Instead, each fuel pump has its own payment terminal. The service times at this payment terminal will remain the same as they were before (in the shop). Customers will no longer be able to buy items they used to buy in the shop.

3. Four lanes with one pump each: Instead of having two lanes (left and right) with two pumps each, there will be four lanes with one pump. From these four lanes, two will be "left lanes", suitable for cars that insist on parking on the left side of a pump, and two will be "right lanes" that can also be chosen by cars insisting on parking on the right side of a pump. Before entering one of these four lanes, cars still have to wait in a queue where no overtaking can take place. This is the main difference between this scenario and Scenario 1: In Scenario 3, a car in the queue can block a car behind it (for example, because it insists on parking on a left lane while there is no free parking spot there). In Scenario 1, this car could be overtaken.

For scenarios 1 and 3, be sure to create a table with simulation results as discussed in the previous section. If you choose scenario 2, where there is no shop or single payment terminal anymore, you only have to provide the upper two rows (waiting time and queue length fuel station) of this table.

Grading

The assignment will be 25% of the final grade of the course 2WB50. The assignment is made in groups of two or three students. Each group should hand in a well-written report and the source code of their simulation programs. **The page limit for your report is 10 pages (not including title page, table of contents, appendix).** More detailed guidelines can be found in Canvas. Upload your report in PDF format before the deadline specified in Canvas. **This is a hard deadline**! Please include your source code as an attachment (uploaded as a **separate** ZIP file).