Simulation

Assignment 2.1 - A queuing model

Ari Viitala 432568

In [124]:

```
import numpy as np
import matplotlib.pyplot as plt
```

A function that will simulate one day of running a taxi business with the given amount of taxis

In [125]:

```
def day(n_taxis):
    #parameters for the poisson process
    lambda_taxi = 11.5
    lambda_customer = 3

    #initialize the simulation clock
    t = 0

    #initializing the arrival of first customer and the arrival of first taxi which is never at this moment
    customer = np.random.exponential(lambda_customer)
    next_taxi = 1000000

    #amount of people hopping on taxis and arriving at their destination
    intake = 0
    outtake= 0

    #list of taxis in operation and the queue for customers
    taxis = []
    queue = []

    while t < 600:
        #while there is time left on the day

        if customer < next_taxi:
            #if a customer arrives before a taxi is ready
            if len(taxis) < n_taxis:
                #if there is a free taxi put them there
                intake += 1
                #create arrival time for taxi
                taxis.append(customer + np.random.exponential(lambda_taxi))
                #make sure the first taxi to be ready is first on the list
                taxis = sorted(taxis)
            elif len(queue) < 16:
                #If no taxis are free but there is room in the queue the put them there. Otherwise do nothing
                queue.append(customer)

            #simulation time is the time customer came
            t = customer
            #make a new arrival time for the next customer
            customer += np.random.exponential(lambda_customer)
            #save the arrival time of the next taxi to be ready to next_taxi variable
            next_taxi = taxis[0]

        else:
            #if taxi is ready before customer comes empty the taxi and increase delivered people
            taxis.pop(0)
            outtake += 1
            if len(queue) > 0:
                #if there is a person in que put them in taxi
                intake += 1
                #simulate the arrival time
                taxis.append(next_taxi + np.random.exponential(lambda_taxi))
                #take a person from queue
                queue.pop(0)
                #make sure the earliest taxi is the first on the list
                taxis = sorted(taxis)

            #update simulation clock to the arrival of the taxi
            t = next_taxi

            if len(taxis) == 0:
                #if there are no taxis driving make sure no taxi will be arriving
                next_taxi = 100000
            else:
                #else next taxi is the one that is ready next
                next_taxi = taxis[0]

    #return the amount of people who we have been taken into taxis before the day has ended
    #it would be bad customer service to leave the on the road
    return intake
```

A function that will simulate taxi business for five years with given amount of taxis

In [130]:

```
def five_years(n_taxis):
    profit = 0
    if n_taxis == 5:
        #if the manager buys a new taxi there will be an initial cost and discounted resell value
        profit = -15000 + 7500 / (1.1)**5

    for i in range(1, 6):
        yearly_profit = 0
        for j in range(0,365):
            #profit for driving the taxis for a day is the delivered people times 5$ on average
            yearly_profit += day(n_taxis) * 5
        if n_taxis == 5:
                #the new taxi will have the maintenance cost taken away from profit
                yearly_profit -= 15000
        #yearly profit will be discounted by the interest rate and added to the total profit
        profit += yearly_profit / (1.1)**i
    #return the discounted profit of the net present value of the cash flow
    return profit
```

Let's simulate a few scenarios of 5 years of taxibusiness and take the difference between 4 and 5 taxis

In [128]:

```
#initialize vector for profits
profits = []

#simulated ten 5 year scenarios of taxi business
for i in range(0,10):
    #print(i)
    #add the difference between four taxis and five taxis to the profits list
    profits.append(five_years(5) - five_years(4))

print("Average difference between the net present values of 5 and 4 taxis: " + str(np.mean(profits)))
```

```
Average difference between the net present values of 5 and 4 taxis: -8073.82316782
```

The company manager can't justify buying a new taxi since it will cause a net loss of several thousand dollars in company's profit in a five year time period on average.