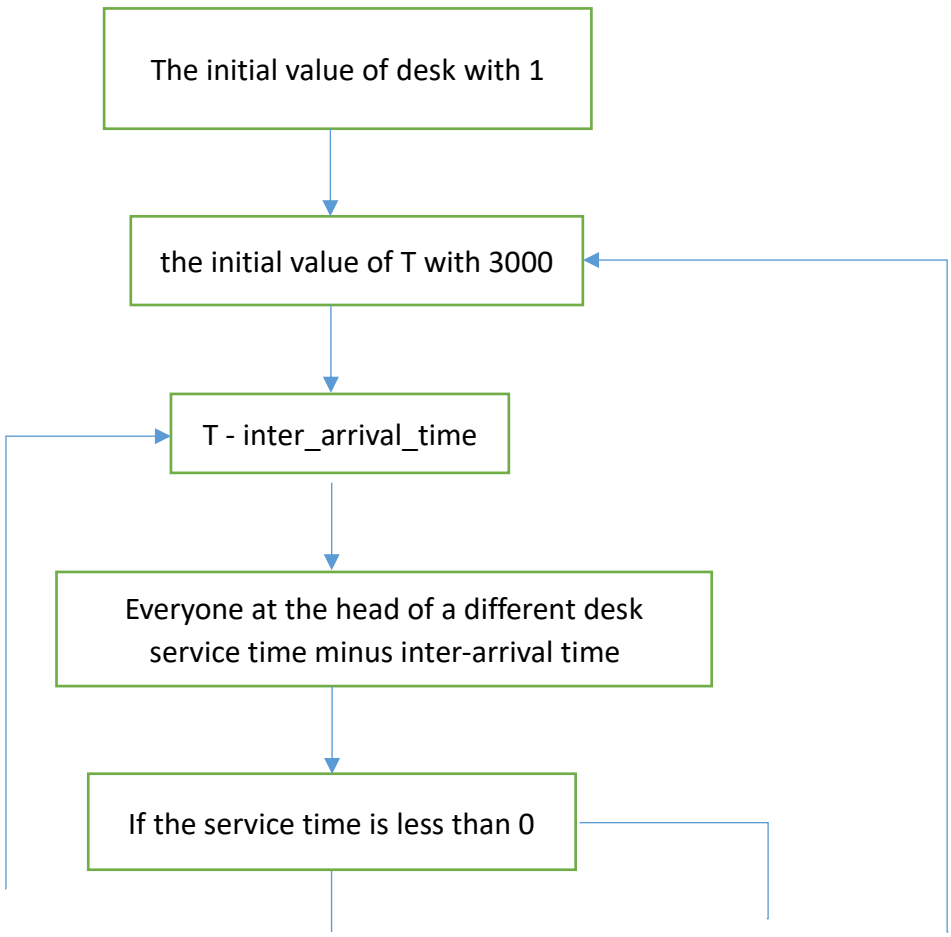
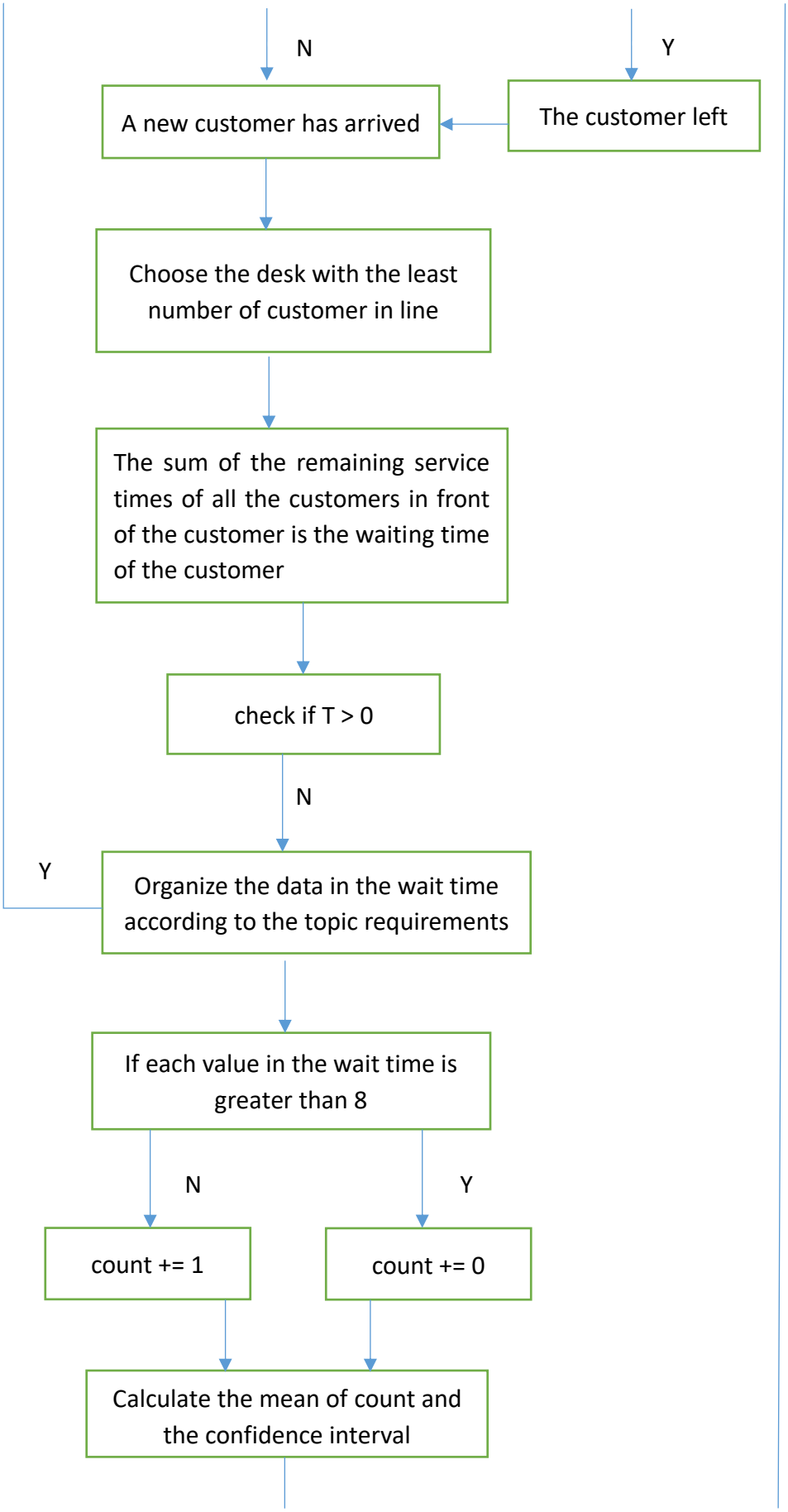
DATA7202 : Assessment 4
Name : Peng Yu
Student ID : 46635884

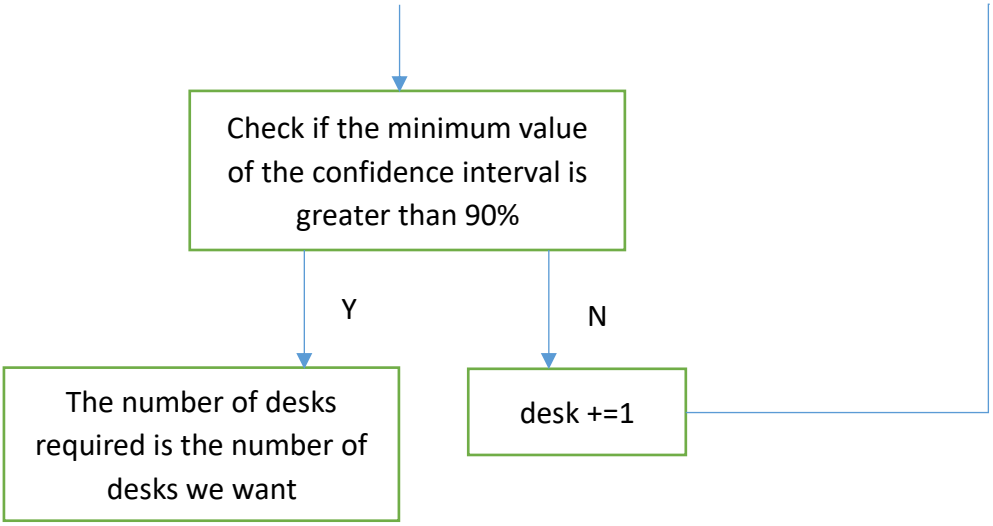
(a)

The meaning of the question is: When we were asked to figure out how many desks there were, less than 10% of us kept customers waiting for more than eight minutes. Each customer has two data, one is “inter-arrival time”, which means the time interval with the previous customer's arrival; The other is “service time”, which means the amount of time the customer spends at the desk. Customers will choose to wait at the desk with the least number of people in line. The customer's waiting time is the time required after the customer inter-arrival time and before the service time. The data in the waiting time is sorted out according to the topic requirements, and then the proportion of customer waiting time less than 8 minutes is calculated, and the confidence interval is calculated. When the minimum value of the confidence interval is greater than 90%, the number of desks required is the number of desks we want.

(b)



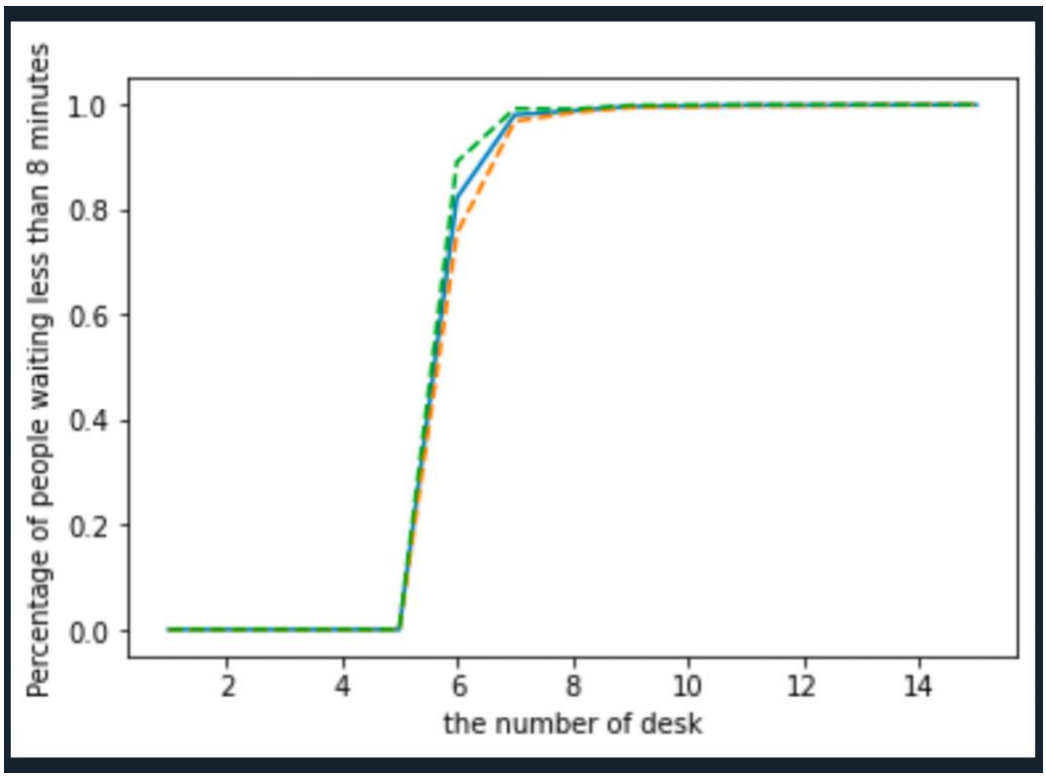

N
Y
A new customer has arrived
The customer left
Choose the desk with the least number of customer in line
The sum of the remaining service times of all the customers in front of the customer is the waiting time of the customer
check if T > 0
N
Y
Organize the data in the wait time according to the topic requirements
If each value in the wait time is greater than 8
N
Y
count += 1
count += 0
Calculate the mean of count and the confidence interval



Below is the parameter I need and what it means.

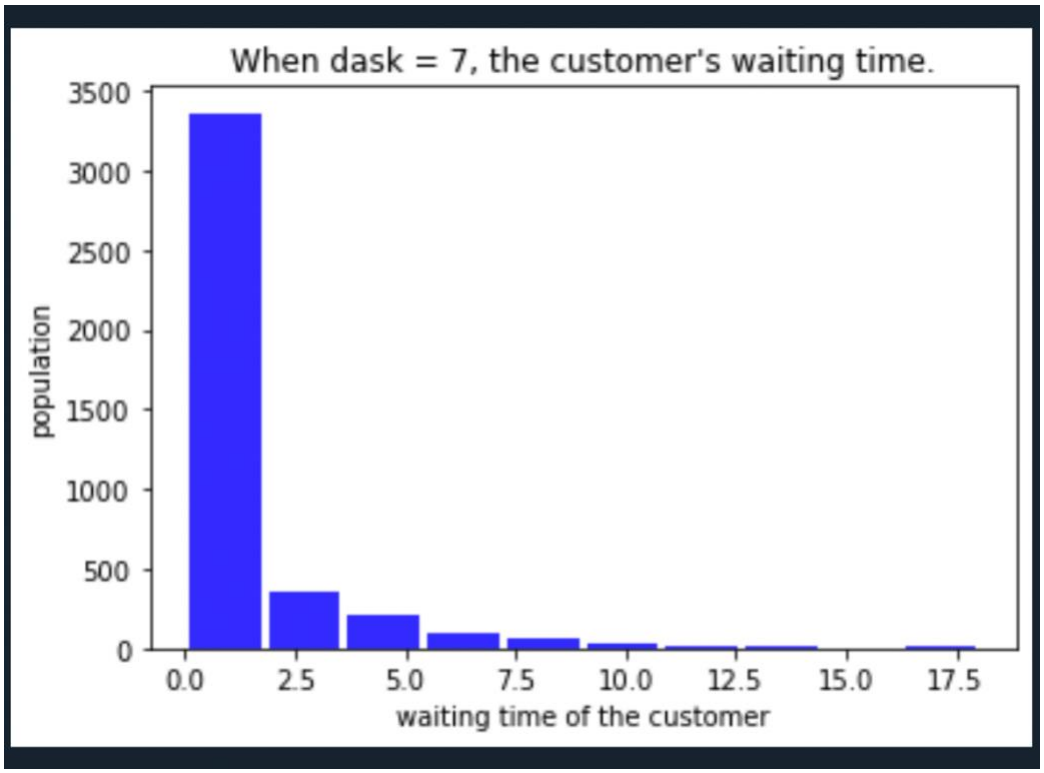
Parameter	Type	Mean
N	int	In the known that using N = 50 batches to estimate the probability
min_desk	int	The minimum number of desks that satisfy the topic condition, and the initial value with -1
wait	list	As many desks as there are, store as many sub-lists in this list. Each sub-list stores the waiting time for each customer with a corresponding number of desks.
result	list	There are three sub-lists in this list. The first sub-list stores the percentage of the number of customers meeting the conditions of the question under a different number of desks. The second sub-list stores the minimum value of the confidence interval for this percentage for a different number of desks; The third sub-list stores the maximum value of the confidence interval of this percentage for the number of desks.
desks	list	There are as many sub-lists as there are desks, and each sub-list stores the service time of the customer of the corresponding desk.
count	list	Under the current number of tables, "0" is stored if the customer has waited more than 8 minutes, and "1" is stored if the customer has waited less than 8 minutes.
count_N	list	Estimated probability of 'count' based on using N = 50 batches.
T	int	In the known that T = 3000 units of time.

(c)



Desk	Probability	95%CI
1	0.0%	(0.0%, 0.0%)
2	0.0%	(0.0%, 0.0%)
3	0.0%	(0.0%, 0.0%)
4	0.0%	(0.0%, 0.0%)
5	0.0%	(0.0%, 0.0%)
6	82.314%	(75.482%, 89.147%)
7	98.075%	(96.867%, 99.284%)
8	98.860%	(98.535%, 99.185%)
9	99.679%	(99.470%, 99.887)
10	99.795%	(99.632%, 100.0%)
11	99.926%	(99.845%, 100.0%)
12	99.923%	(99.838%, 100.0%)
13	99.967%	(99.905%, 100.0%)
14	100.0%	(100.0%, 100.0%)
15	100.0%	(100.0%, 100.0%)

As can be seen from the graph, when the number of desks is no more than 5, the waiting time of customers is more than 8 minutes. But when the number of desks exceeded 5, the proportion of customers waiting less than 8 minutes increased significantly. When the number of desks is more than 8, most customers wait less than 8 minutes.



Waiting time	(0.0~1.8)	(1.8~3.6)	(3.6~5.4)	(5.4~7.2)	(7.2~9.0)	(9.0~10.8)	(10.8~12.6)	(12.6~14.4)	(14.4~16.2)	(16.2~)
Proportion of the population	81.165%	8.656%	4.860%	2.442%	1.499%	0.629%	0.266%	0.290%	0.048%	0.145%

As can be seen from the above figure, when the number of desks is 7, the waiting time of most customers is less than 3.6 minutes, and few customers who wait more than 8 minutes are left behind.

```
The minimum number of desks is 7.
The mean of the percentage that meets the requirement =  0.9807592988264636
The CI of the percentage that meets the requirement = ( 0.9686747876163044 , 0.9928438100366228 )
```

(d)

From the above analysis, we calculate that when desk=7, is the minimum number of desks that meet the topic condition. We calculate that when desk is 7, the mean of customer waiting time is 0.9898m, and its confidence interval is (0.9241m, 1.0554m).

```
the mean of customer waiting time =  0.9898329074899873
the CI of customer waiting time = ( 0.9241743610864085 , 1.055491453893566 )
```

(e)

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

df = pd.read_csv('data.csv')

np.random.seed(len(df))

N = 50
min_desk = -1
wait = []
result = [[],[],[]]

for desk in range(1, 16):
    desks = []
    count = []
    count_N = []
    T = 3000
    wait.append([])

    for i in range(desk):
        desks.append([])

    while T >= 0:
        row = np.random.randint(len(df))
        T -= df.loc[row].values[0]

        #Check whether the remaining service time of the first customer on each desk is negative. If so, delete his time and add the negative time to the service time of the next customer.
        for i in range(desk):
            if desks[i]:
                desks[i][0] -= df.loc[row].values[0]
                while desks[i][0] <= 0:
                    if desks[i][1:]:
                        desks[i][1] += desks[i][0]
```

```
                    desks[i] = desks[i][1:]
                elif not desks[i][1:]:
                    desks[i] = desks[i][1:]
                    break

        #Find the desk with the smallest queue.
        #To be clear, I'm not looking for No. of desk with the smallest queue here,
but I'm setting No. of desk with the smallest queue to be 0.
        for i in range(len(desks)):
            least = i
            for k in range(i + 1, len(desks)):
                if len(desks[k]) < len(desks[least]):
                    least = k
            desks[least], desks[i] = desks[i], desks[least]

        #Obtain the wait time of the customer
        desks[0].append(df.loc[row].values[1])
        wait[desk - 1].append(sum(desks[0][:-1]))

    #Discard the first 30% of the samples
    index = (len(wait[desk - 1]) * 3) // 10
    wait[desk - 1] = wait[desk - 1][index:]

    #Gets the number of customers whose wait time is less than 8 minutes
    for i in range(len(wait[desk - 1])):
        if wait[desk - 1][i] > 8:
            count.append(0)
        elif wait[desk - 1][i] <= 8:
            count.append(1)

    #Using N = 50 batches to estimate the probability
    for i in range(N):

count_N.append(np.mean(count[(0+i*len(count)//50):(len(count)//50)*(i+1)]))

    #The percentage of the number of customers meeting the conditions
    result[0].append(np.mean(count_N))

    #The minimum value of the confidence interval for this percentage
    result[1].append(np.mean(count_N) - 1.96 * np.std(count_N) / len(count_N) **
0.5)
```

```
        #The maximum value of the confidence interval of this percentage
        result[2].append(np.mean(count_N) + 1.96 * np.std(count_N) / len(count_N) **
0.5)

        #Find out the minimum number of desks that meet the topic condition
        if min_desk == -1:
                if result[1][desk-1] > 0.9:
                        min_desk = desk

print('The minimum number of desks is {}.'.format(min_desk))
print('The mean of the percentage that meets the requirement = ',result[0][min_desk
- 1])
print('The CI of the percentage that meets the requirement = (',result[1][min_desk -
1], ',',result[2][min_desk - 1],')')

plt.plot(range(1,16), result[0])
plt.plot(range(1,16), result[1], linestyle='--')
plt.plot(range(1,16), result[2], linestyle='--')
plt.xlabel('the number of desk')
plt.ylabel('Percentage of people waiting less than 8 minutes')
plt.show()

fig = plt.figure()
ax = fig.add_subplot(111)
ax.hist(wait[min_desk-1],color='blue',alpha=0.8,rwidth=0.9)
plt.xlabel('waiting time of the customer')
plt.ylabel('population')
plt.title(u"When dask = 7, the customer's waiting time.")
plt.show()

print('the mean of customer waiting time = ',np.mean(wait[min_desk-1]))
print('the CI of customer waiting time = (',np.mean(wait[min_desk-1])-
1.96*np.std(wait[min_desk-1])/len(wait[min_desk-1])**0.5,
',',np.mean(wait[min_desk-1])+1.96*np.std(wait[min_desk-1])/len(wait[min_desk-
1])**0.5,')')
```