Written Report

Group B - L02

HE Yugao, s196223

LI Minger, s196355

WU Brandon, s196225

The Hang Seng University of Hong Kong

AMS3301 - Simulation

Dr. Benson Lam

15 December 2021

Table of Contents

Part 1: Study of Analytical Solution

Introduction

The chosen datasets are dataset 20 and dataset 21, and the details are shown below.

Name	Mean	Type
Dataset 20	28.7811	Continuous
Dataset 21	31.4912	Continuous

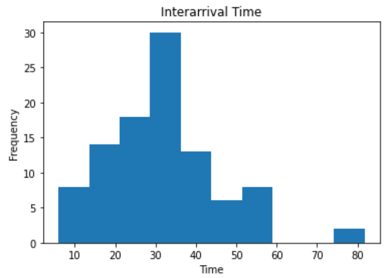
Methodology

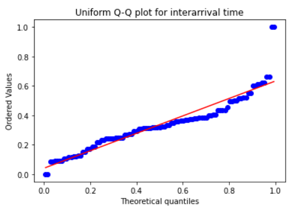
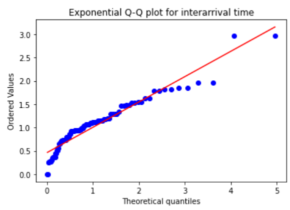
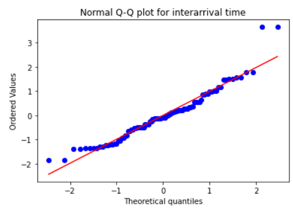
Since both datasets are continuous, we will use the Kolmogorov Smirnov Test (KS test) to perform hypothesis testing. Furthermore, because dataset 21 has a higher mean than dataset 20, we will set dataset 21 as interarrival time and set dataset 20 as service time, to avoid the system breaking down.

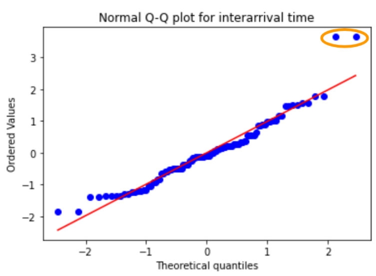
Results

Interarrival Time (dataset 21)

From the histogram, it shows that the interarrival time looks like a normal distribution. After going through the QQ plot and KS test, the data does indeed follow a normal distribution (see graph below). The reason is because the QQ plot fits well and the KS test's p-value > 0.05, hence the null hypothesis is not rejected. Thus, there is not enough evidence that the data does not follow a normal distribution



Uniform Distribution	Exponential Distribution	Normal Distribution
		
KS test Test-statistic: 0.3558 P-value: 9.9939*E-12	KS Test Test-statistic: 0.2566 P-value: 3.080*E-06	KS Test Test-statistic: 0.1026 P-value: 0.2316

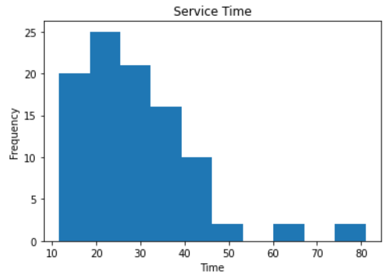


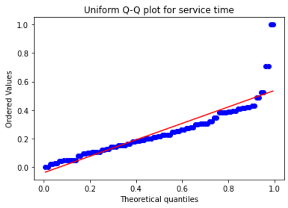
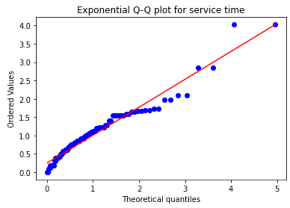
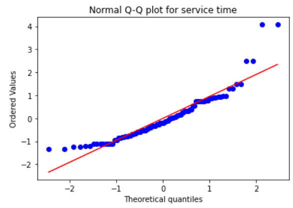
From the QQ plot, 2 outliers were discovered (circled in orange), thus we have removed them from the data to have a more precise estimation of the distribution. After doing the KS test again, the test-statistic was lower at 0.0636, and the p-value was higher at 0.8035, thus removing the outliers helped substantially in determining the distribution.

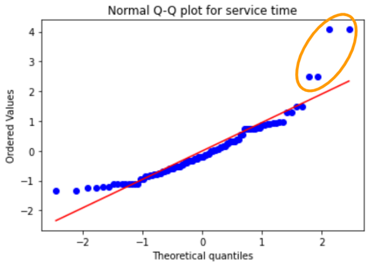
As a result, we will simulate the interval time by a normal distribution with a mean of 30.8099, and a standard deviation of 11.6043.

Service Time (dataset 20)

From the histogram, it shows that the service time may follow a normal distribution or uniform distribution. After going through the QQ plot and KS test, the data follows a normal distribution as well. Once again, the QQ plot fits well and the KS test's p-value > 0.05, hence the null hypothesis is not rejected. Therefore, there is not enough evidence that the data does not follow a normal distribution.



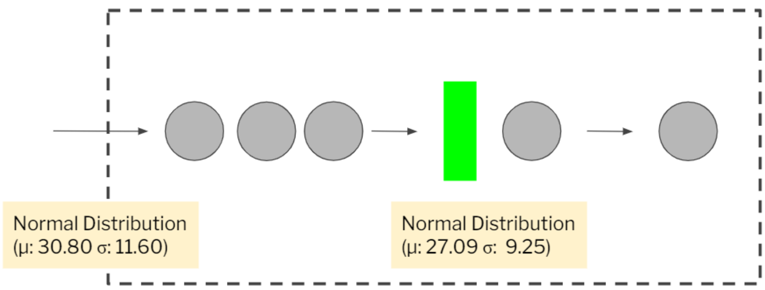
Uniform Distribution	Exponential Distribution	Normal Distribution
		
<u>KS Test</u> Test-statistic: 0.4890 P-value: 3.9134*E-22	<u>KS Test</u> Test-statistic: 0.1526 P-value: 0.0184	<u>KS Test</u> Test-statistic: 0.0961 P-value: 0.3054



From the QQ plot, 4 outliers (circled in orange) were discovered and removed from the data. After performing the KS test again, the test-statistic was lower at 0.0949, and the p-value was higher at 0.3443, hence removing the outliers improved the accuracy of the test.

Based on the test, we will simulate the service time by a normal distribution with a mean of 27.0924, and a standard deviation of 9.2516.

With this information, we can form the queueing system, which is shown below.



After plugging in our data into the Java Modelling Tools System, here are the results.

Performance Indices	Min	Mean	Max
Number of customers of the queue	1.5391	1.5688	1.5985
Queue time	20.2867	20.8315	21.3762
Response Time of the counter	47.8023	49.1303	50.4582

From the table, we can see that the average queue time is at 20.8315 seconds, which is long.

Suggestions for Improvement

In order to improve the system, we have increased the server amount to 2.

Performance Indices	Min	Mean	Max
Number of customers of the queue	0.8564	0.8820	0.9075
Queue time	0.1579	0.1626	0.1674
Response Time of the counter	26.4902	27.1962	27.9023

After the change, all the performance indices have improved. In particular, the average queue time has decreased by 20.6689 seconds.

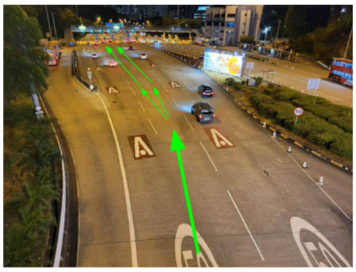
Part 2: Real World Modelling

Introduction

Tate's Cairn Tunnel is a tunnel that connects the Shatin and the Kowloon district. Every day, students and workers come to Shatin or Kowloon by taking a bus or driving a car. Our group will analyze the queuing system of the Tate's Cairn Tunnel Toll Plaza (e.g. how long would a car take to pass the Toll Plaza). The data was collected at 7 P.M. December 4th.

Methodology

The toll plaza can be defined as a two queue system. Cars come from 1 big road that is split into 2 smaller roads, and the drivers can choose a road with a shorter queue. We filmed a video to record two servers, named line 11 and line 12, with a total of 90 records. We assumed that all payment methods are the same. As the interarrival time and service time are continuous, we will use the KS test to perform hypothesis testing.

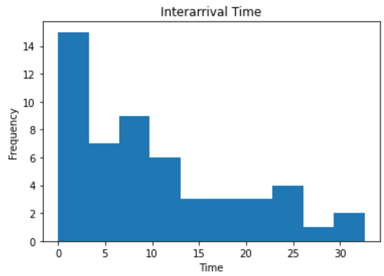


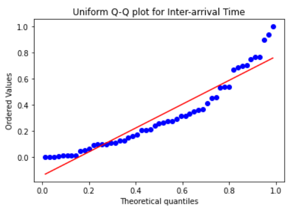
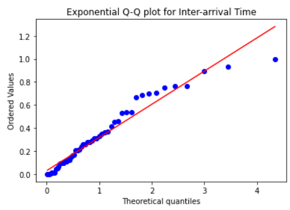
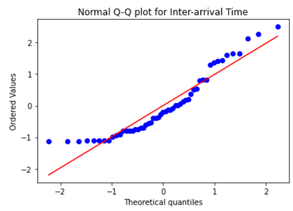
We used three steps to record the data:
Step 1: When the cars appear on the video, the arrival time will be recorded
Step 2: When the cars stop at the toll plaza, we will start recording the service time
Step 3: When the cars leave, stop recording the service time

Results

Interarrival Time

In the histogram, the interarrival time obviously looks like an exponential distribution. We used the QQ plot and the KS-test for the interarrival time. The normal distribution test shows a test-statistic of 0.1276 and a p-value of 0.3257. As the p-value is larger than 0.05, we do not reject the null hypothesis, thus there is not enough evidence that the interarrival time does not follow a normal distribution. We deleted 1 outlier based on the QQ plot to reduce the error in the tests.

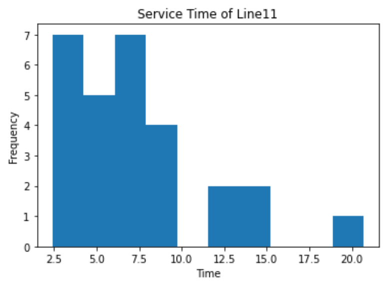


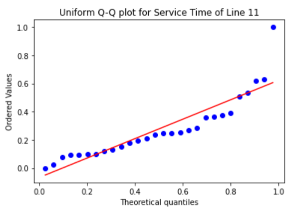
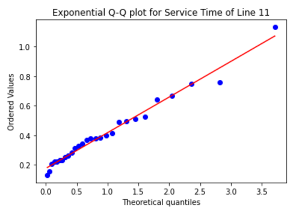
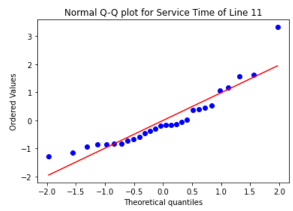
Uniform Distribution	Exponential Distribution	Normal Distribution
		
KS Test Test-statistic: 0.3300 P-value: 1.2082*E-05	KS Test Test-statistic: 0.4079 P-value: 1.7495*E-8	KS Test Test-statistic: 0.1276 P-value: 0.3257

We simulated the interarrival time by a normal distribution with a mean of 10.0056, and a standard deviation of 5.0028.

Service Time of Line 11

In the histogram, the service time of line 11 looks like an exponential distribution. The QQ plot and KS test is used in the testing. The normal distribution has the smallest test-statistic of 0.1681 and a p-value of 0.3657. The p-value is larger than 0.05, hence we do not reject the null hypothesis, thus there is not enough evidence that the service time of line 11 data does not follow a normal distribution. We deleted 1 outlier based on the QQ plot to reduce the error in the tests.

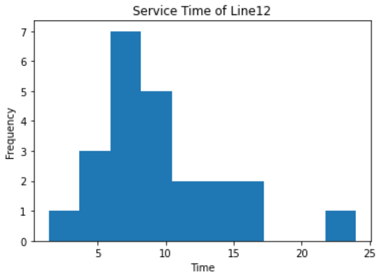


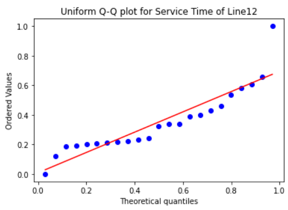
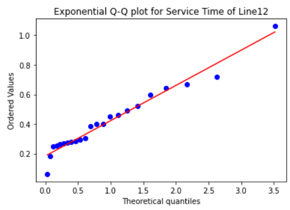
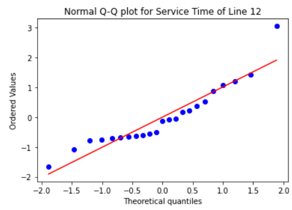
Uniform Distribution	Exponential Distribution	Normal Distribution
		
KS Test Test-statistic: 0.4280 P-value: 3.5755*E-5	KS Test Test-statistic: 0.4322 P-value: 2.8647*E-5	KS Test Test-statistic: 0.1681 P-value: 0.3657

We simulated the service time of line 11 by a normal distribution with a mean of 7.3862, and a standard deviation of 3.6931.

Service Time of Line 12

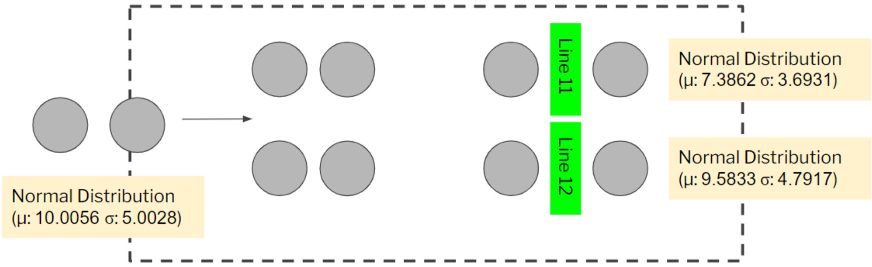
In the histogram, the service time of line 12 looks like a normal distribution. We applied the QQ plot and KS test. The normal distribution has the smallest test-statistic that equals 0.1731 and a p-value that equals 0.4462. Since the p-value is larger than 0.05, we do not reject the null hypothesis, thus there is not enough evidence that the service time of line 12 does not follow a normal distribution. We deleted 1 outlier based on the QQ plot to reduce the error in the tests.



Uniform Distribution	Exponential Distribution	Normal Distribution
		
<u>KS Test</u> Test-statistic: 0.3224 P-value: 0.0125	<u>KS Test</u> Test-statistic: 0.4448 P-value: 0.0001	<u>KS Test</u> Test-statistic: 0.1731 P-value: 0.4462

We simulated the service time of line 12 by a normal distribution with a mean of 9.5833, and a standard deviation of 4.7917.

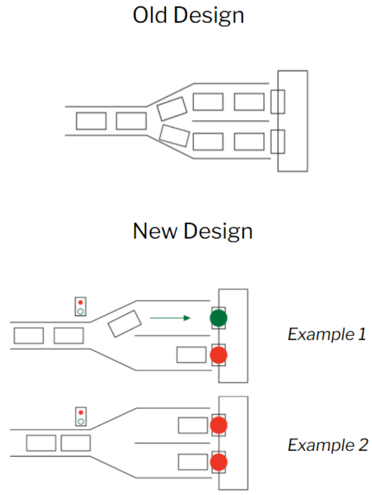
With the results, we can form the queueing system, which is shown below.



After inputting the data into the Java Modelling Tools System, the results are as follows.

Performance Indices	**Min**	**Mean**	**Max**
Number of customers of the queue	Line 11: 0.3964 Line 12: 0.4527	Line 11: 0.4056 Line 12: 0.4657	Line 11: 0.4147 Line 12: 0.4788
Queue time	Line 11: 0.1874 Line 12: 0.2663	Line 11: 0.1910 Line 12: 0.2738	Line 11: 0.1946 Line 12: 0.2813
Response Time of the counter	Line 11: 7.6761 Line 12: 9.9588	Line 11: 7.7861 Line 12: 10.1021	Line 11: 7.8960 Line 12: 10.2454
System number of customers	0.8514	0.8703	0.8891

Improving the Queuing System of the Toll Plaza

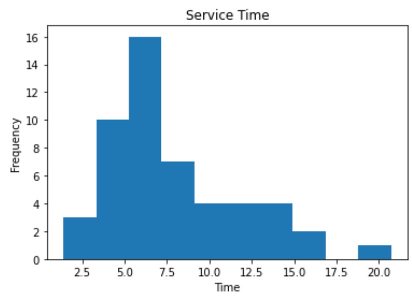


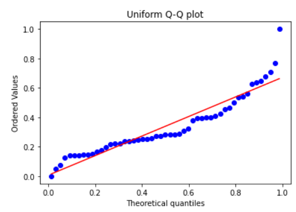
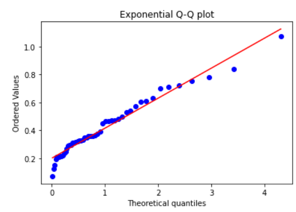
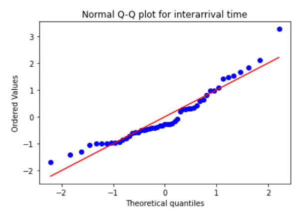
From the old design, drivers can enter the shortest queue anytime they want, thus it forms a two queue order system. In the new design, we added traffic lights. It helps maintain one queue and allows drivers to pass only when a server is empty. With traffic lights, it will change the system into a one queue two server system. In example 1, the top lane is empty, so the traffic light lets the car pass, and the toll shows a green light signalling that it is open, thus the car moves to the top lane. In example 2, since all the lanes are occupied, the traffic light is red, maintaining one queue.

The cars come from one queue and go into the combination of Line 11 and 12. Since we already know the inter-arrival time of the cars fits an exponential distribution with the lambda equals 0.0915. Then, we are going to study which distribution does the combination of Line 11 and 12 fit.

Service Time of Line 11 and Line 12

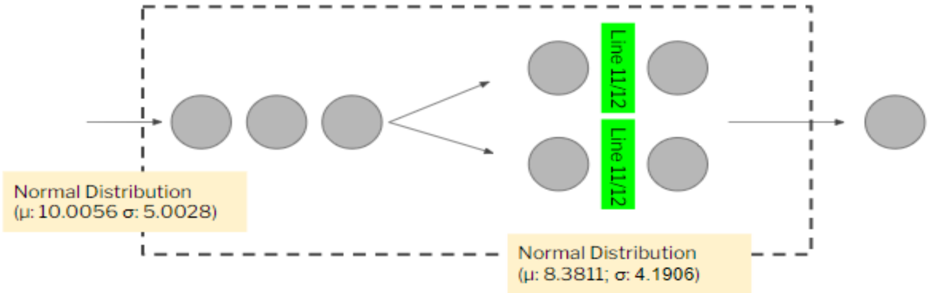
Here is a histogram showing how the service time is distributed. We think it may look like a normal distribution. We used the QQ plot and the KS Test for the above three distributions, which are Uniform, Exponential, and Normal. After comparing, the normal distribution has the lowest test-statistics among three and the p-value is 0.1139, which is greater than 0.05. Hence, the null hypothesis is not rejected and there is not enough evidence that the data do not follow a normal distribution. For higher accuracy, we removed three extreme values in the dataset based on the QQ plot.



Uniform Distribution	Exponential Distribution	Normal Distribution
		
KS Test Test-statistic: 0.3202 P-value: 3.7527e-05	KS Test Test-statistic: 0.4181 P-value: 1.3633e-08	KS Test Test-statistic: 0.1642 P-value: 0.1139

We stimulated the service time for line 11 and 12 as a normal distribution with a mean of 10.0056 and a standard deviation of 5.0027.

Based on the results above, the full model is completed here. We simulated the service time of line 11 and 12 by a normal distribution with a mean of 8.3811, and a standard deviation of 4.1906.



We used the Java Modelling Tool System to estimate this model by three parameters. The results are shown below.

Performance Indices	**Min**	**Mean**	**Max**
Number of customers of the queue	0.8423	0.8588	0.8754
Queue time	0.1337	0.1367	0.1397
Response Time of the counter	8.5106	8.7725	9.0344

Direct Comparison

<table>
<tr><th>Performance Indices</th><th colspan="2">Original Model</th><th>Improved Model</th></tr>
<tr><td rowspan="2">Number of Customers of the Queue</td><td>Line11: 0.4056</td><td>Line12: 0.4657</td><td rowspan="2">0.8588</td></tr>
<tr><td colspan="2">Total: 0.8713</td></tr>
<tr><td rowspan="2">Queue Time</td><td>Line11: 0.1910</td><td>Line12: 0.2738</td><td rowspan="2">0.1367</td></tr>
<tr><td colspan="2">Average: 0.2324</td></tr>
<tr><td rowspan="2">Response Time of the Counter</td><td>Line11: 7.7861</td><td>Line12: 10.1021</td><td rowspan="2">8.7725</td></tr>
<tr><td colspan="2">Average: 8.9941</td></tr>
</table>

For the comparison, the newly designed model has performed better in all performance indices. Compared to the original model, the improved model has relatively lower processing time among these three parameters. Especially the queue time, which is one second shorter. In handling heavy traffic, the improved model can greatly reduce the waiting time of each customer and help to operate the toll system more efficiently and smoothly.

Appendix 1: Code

```
# -*- coding: utf-8 -*-
"""
Created on Wed Dec 15 22:18:36 2021
@author: Brandon
"""
# -*- coding: utf-8 -*-
"""
Created on Mon Dec  6 15:19:56 2021
@author: Brandon
"""

#Part 1
#Interarrival time
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import scipy.stats as stats

obsit = pd.read_excel("dataset 21.xlsx")
obsit = obsit.values
plt.ylabel("Frequency")
plt.xlabel("Time")
plt.title("Interarrival Time")
plt.hist(obsit)

#Test for uniform distribution
import scipy.stats as stats
z = (obsit-np.min(obsit))/(np.max(obsit)-np.min(obsit)) #Data
normalization
z = z.reshape(-1) #change to array
plt.figure()  #plot a figure
stats.probplot(z, dist=stats.uniform, plot=plt)  #plot the QQ plot to plt
plt.title("Uniform Q-Q plot for interarrival time") #add a title to the
figure
print(stats.kstest(z,'uniform')) #Apply KS-test

#Test for exponential distribution
import scipy.stats as stats
z = (obsit-np.min(obsit))/np.mean(obsit-np.min(obsit)) #Data
normalization
z = z.reshape(-1) #change to array
plt.figure()  #plot a figure
stats.probplot(z, dist=stats.expon, plot=plt)  #plot the QQ plot to plt
plt.title("Exponential Q-Q plot for interarrival time") #add a title to
the figure
print(stats.kstest(z,'expon')) #Apply KS-test

#Test for normal distribution
import scipy.stats as stats
```

```
z = (obsit-np.mean(obsit))/np.std(obsit) #Data normalization
z = z.reshape(-1) #change to array
plt.figure()  #plot a figure
stats.probplot(z, dist=stats.norm, plot=plt)  #plot the QQ plot to plt
plt.title("Normal Q-Q plot for interarrival time") #add a title to the
figure
print(stats.kstest(z,'norm')) #Apply KS-test

#Removing the outliers
obsitol = pd.read_excel("dataset 21 (no outlier).xlsx")
obsitol = obsitol.values
plt.ylabel("Frequency")
plt.xlabel("Time")
plt.title("Interarrival Time")
plt.hist(obsitol)

#Test for normal distribution for no outliers
import scipy.stats as stats
z = (obsitol-np.mean(obsitol))/np.std(obsitol) #Data normalization
z = z.reshape(-1) #change to array
plt.figure()  #plot a figure
stats.probplot(z, dist=stats.norm, plot=plt)  #plot the QQ plot to plt
plt.title("Normal Q-Q plot for interarrival time") #add a title to the
figure
print(stats.kstest(z,'norm')) #Apply KS-test

#Plot normal distribution for no outliers
from scipy.stats import norm
_, bins, _ = plt.hist(obsitol, 7, density=1)
mu = obsitol.mean()
sigma = obsitol.std()
best_fit_line = norm.pdf(bins, mu, sigma)
plt.ylabel("Percentage")
plt.xlabel("Time")
plt.title("Interarrival Time")
plt.plot(bins, best_fit_line)

#Service time
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import scipy.stats as stats

obsst = pd.read_excel("dataset 20.xlsx")
obsst = obsst.values
plt.ylabel("Frequency")
plt.xlabel("Time")
plt.title("Service Time")
plt.hist(obsst)

#Test for uniform distribution
```

```
import scipy.stats as stats
z = (obsst-np.min(obsst))/(np.max(obsst)-np.min(obsst)) #Data
normalization
z = z.reshape(-1) #change to array
plt.figure()  #plot a figure
stats.probplot(z, dist=stats.uniform, plot=plt)  #plot the QQ plot to plt
plt.title("Uniform Q-Q plot for service time") #add a title to the figure
print(stats.kstest(z,'uniform')) #Apply KS-test

#Test for exponential distribution
import scipy.stats as stats
z = (obsst-np.min(obsst))/np.mean(obsst-np.min(obsst)) #Data
normalization
z = z.reshape(-1) #change to array
plt.figure()  #plot a figure
stats.probplot(z, dist=stats.expon, plot=plt)  #plot the QQ plot to plt
plt.title("Exponential Q-Q plot for service time") #add a title to the
figure
print(stats.kstest(z,'expon')) #Apply KS-test

#Test for normal distribution
import scipy.stats as stats
z = (obsst-np.mean(obsst))/np.std(obsst) #Data normalization
z = z.reshape(-1) #change to array
plt.figure()  #plot a figure
stats.probplot(z, dist=stats.norm, plot=plt)  #plot the QQ plot to plt
plt.title("Normal Q-Q plot for service time") #add a title to the figure
print(stats.kstest(z,'norm')) #Apply KS-test

#Removing the outliers
obsstol = pd.read_excel("dataset 20 (no outlier).xlsx")
obsstol = obsstol.values
plt.ylabel("Frequency")
plt.xlabel("Time")
plt.title("Interarrival Time")
plt.hist(obsstol)

#Test for normal distribution for no outliers
import scipy.stats as stats
z = (obsstol-np.mean(obsstol))/np.std(obsstol) #Data normalization
z = z.reshape(-1) #change to array
plt.figure()  #plot a figure
stats.probplot(z, dist=stats.norm, plot=plt)  #plot the QQ plot to plt
plt.title("Normal Q-Q plot for service time") #add a title to the figure
print(stats.kstest(z,'norm')) #Apply KS-test

#Plot normal distribution for no outliers
from scipy.stats import norm
_, bins, _ = plt.hist(obsstol, 5, density=1)
mu = obsstol.mean()
sigma = obsstol.std()
```

```
best_fit_line = norm.pdf(bins, mu, sigma)
plt.ylabel("Percentage")
plt.xlabel("Time")
plt.title("Service Time")
plt.plot(bins, best_fit_line)

#Part 2
#Inter-Arrival Time
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import scipy.stats as stats

obsit2 = pd.read_excel('InterArrivalTime.xlsx')
obsit2 = obsit2.values
plt.ylabel("Frequency")
plt.xlabel("Time")
plt.title("Interarrival Time")
plt.hist(obsit2); #Visualize data using histogram

#Test for uniform distribution
import scipy.stats as stats
z = (obsit2-np.min(obsit2))/(np.max(obsit2)-np.min(obsit2)) #Data
normalization
z = z.reshape(-1) #change to array
plt.figure() #plot a figure
stats.probplot(z, dist=stats.uniform, plot=plt) #plot the QQ plot to plt
plt.title("Uniform Q-Q plot for Interarrival Time") #add a title to the
figure
print(stats.kstest(z,'uniform')) #Apply KS-test

#Test for expon distribution
z = obsit2/(np.max(obsit2)-np.min(obsit2)) #Data normalization
z = z.reshape(-1) #change to array
plt.figure() #plot a figure
stats.probplot(z.reshape(-1), dist=stats.expon, plot=plt) #plot the QQ
plot to plt
plt.title("Exponential Q-Q plot for Interarrival Time") #add a title to
the figure
print(stats.kstest(z,'expon')) #Apply KS-test

#Test for normal distribution
import scipy.stats as stats
z = (obsit2-np.mean(obsit2))/np.std(obsit2) #Data normalization
z = z.reshape(-1) #change to array
plt.figure() #plot a figure
stats.probplot(z, dist=stats.norm, plot=plt) #plot the QQ plot to plt
plt.title("Normal Q-Q plot for Inter-arrival Time") #add a title to the
figure
print(stats.kstest(z,'norm')) #Apply KS-test
```

```
#Plot normal distribution for no outliers
from scipy.stats import norm
_, bins, _ = plt.hist(obsit2, 5, density=1)
mu = obsit2.mean()
sigma = obsit2.std()
best_fit_line = norm.pdf(bins, mu, sigma)
plt.ylabel("Percentage")
plt.xlabel("Time")
plt.title("Service Time")
plt.plot(bins, best_fit_line)

#Line 11
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import scipy.stats as stats

obs11 = pd.read_excel('Line11ServiceTime.xlsx')
obs11 = obs11.values
plt.title("Service Time of Line11")
plt.ylabel("Frequency")
plt.xlabel("Time")
plt.title("Service Time of Line11")
plt.hist(obs11);

#Test for uniform distribution
import scipy.stats as stats
z = (obs11-np.min(obs11))/(np.max(obs11)-np.min(obs11)) #Data
normalization
z = z.reshape(-1) #change to array
plt.figure() #plot a figure
stats.probplot(z, dist=stats.uniform, plot=plt) #plot the QQ plot to plt
plt.title("Uniform Q-Q plot for Service Time of Line 11") #add a title to
the figure
print(stats.kstest(z,'uniform')) #Apply KS-test

#Test for expon distribution
z = obs11/(np.max(obs11)-np.min(obs11)) #Data normalization
z = z.reshape(-1) #change to array
plt.figure() #plot a figure
stats.probplot(z.reshape(-1), dist=stats.expon, plot=plt) #plot the QQ
plot to plt
plt.title("Exponential Q-Q plot for Service Time of Line 11") #add a
title to the figure
print(stats.kstest(z,'expon')) #Apply KS-test

#Test for normal distribution
import scipy.stats as stats
z = (obs11-np.mean(obs11))/np.std(obs11) #Data normalization
z = z.reshape(-1) #change to array
plt.figure() #plot a figure
```

```
stats.probplot(z, dist=stats.norm, plot=plt) #plot the QQ plot to plt
plt.title("Normal Q-Q plot for Service Time of Line 11") #add a title to
the figure
print(stats.kstest(z,'norm')) #Apply KS-test

#Plot exponential distribution
from scipy.stats import expon
_, bins, _ = plt.hist(obs11, 10, density=1)
sigma = obs11.std()
best_fit_line = expon.pdf(bins, 0, sigma)
plt.ylabel("Percentage")
plt.xlabel("Time")
plt.title("Line11 Service Time")
plt.plot(bins, best_fit_line)

#Line12
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import scipy.stats as stats

obs12 = pd.read_excel('Line12ServiceTime.xlsx')
obs12 = obs12.values
plt.ylabel("Frequency")
plt.xlabel("Time")
plt.title("Service Time of Line12")
plt.hist(obs12); #Visualize data using histogram

#Test for uniform distribution
import scipy.stats as stats
z = (obs12-np.min(obs12))/(np.max(obs12)-np.min(obs12)) #Data
normalization
z = z.reshape(-1) #change to array
plt.figure() #plot a figure
stats.probplot(z, dist=stats.uniform, plot=plt) #plot the QQ plot to plt
plt.title("Uniform Q-Q plot for Service Time of Line12") #add a title to
the figure
print(stats.kstest(z,'uniform')) #Apply KS-test

#Test for expon distribution
z = obs12/(np.max(obs12)-np.min(obs12)) #Data normalization
z = z.reshape(-1) #change to array
plt.figure() #plot a figure
stats.probplot(z.reshape(-1), dist=stats.expon, plot=plt) #plot the QQ
plot to plt
plt.title("Exponential Q-Q plot for Service Time of Line12") #add a title
to the figure
print(stats.kstest(z,'expon')) #Apply KS-test

#Test for normal distribution
import scipy.stats as stats
```

```
z = (obs12-np.mean(obs12))/np.std(obs12) #Data normalization
z = z.reshape(-1) #change to array
plt.figure() #plot a figure
stats.probplot(z, dist=stats.norm, plot=plt) #plot the QQ plot to plt
plt.title("Normal Q-Q plot for Service Time of Line 12") #add a title to
the figure
print(stats.kstest(z,'norm')) #Apply KS-test

#Plot for normal distribution
from scipy.stats import norm
_, bins, _ = plt.hist(obs12, 5, density=1)
mu = obs12.mean()
sigma = obs12.std()
best_fit_line = norm.pdf(bins, mu, sigma)
plt.ylabel("Percentage")
plt.xlabel("Time")
plt.title("Service Time")
plt.plot(bins, best_fit_line)

#Line 11 and 12
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
obs11and12 = pd.read_excel("Line11and12Together.xlsx")
obs11and12 = obs11and12.values
plt.title("Line11/12 Service Time")
plt.ylabel("Frequency")
plt.xlabel("Time")
plt.hist(obs11and12)

#Test for uniform distribution
import scipy.stats as stats
z = (obs11and12-np.min(obs11and12))/(np.max(obs11and12)-
np.min(obs11and12)) #Data normalization
z = z.reshape(-1) #change to array
plt.figure() #plot a figure
stats.probplot(z, dist=stats.uniform, plot=plt) #plot the QQ plot to plt
plt.title("Uniform Q-Q plot") #add a title to the figure
print(stats.kstest(z,'uniform')) #Apply KS-test

#Test for exponential distribution
z = obs11and12/(np.max(obs11and12)-np.min(obs11and12)) #Data
normalization
z = z.reshape(-1) #change to array
plt.figure() #plot a figure
stats.probplot(z.reshape(-1), dist=stats.expon, plot=plt) #plot the QQ
plot to plt
plt.title("Exponential Q-Q plot") #add a title to the figure
print(stats.kstest(z,'expon')) #Apply KS-test

#Test for normal distribution
```

```
import scipy.stats as stats
z = (obs11and12-np.mean(obs11and12))/np.std(obs11and12) #Data
normalization
z = z.reshape(-1) #change to array
plt.figure() #plot a figure
stats.probplot(z, dist=stats.norm, plot=plt) #plot the QQ plot to plt
plt.title("Normal Q-Q plot for interarrival time") #add a title to the
figure
print(stats.kstest(z,'norm')) #Apply KS-test

#Plot normal distribution
from scipy.stats import norm
_, bins, _ = plt.hist(obs11and12, 10, density=1)
mu = obs11and12.mean()
sigma = obs11and12.std()
best_fit_line = norm.pdf(bins, mu, sigma)
plt.ylabel("Percentage")
plt.xlabel("Time")
plt.title("Line11/12 Service Time")
plt.plot(bins, best_fit_line)
```