In [301…

```
from IPython.core.display import display, HTML
display(HTML("<style>.container { width:95% !important; }</style>"))
```

In [291…

```
# Importing The Library
```

In [222…

```
import numpy as np
import scipy.stats as ss
import matplotlib.pyplot as plt
import pandas as pd
import random
import math
import seaborn as sns
```

In [ ]:

```
# Reading The Data File
```

In [245…

```
data_demand=pd.read_csv("STORE_DATA.csv")
data_demand
```

Out[245…

	Account	Store	ShipDate	SumOfShipwght	Tour	Store_region	Store_address	store_city
0	5	28	3-Jan-97	481.28	90.0	TX	3620 EMMETT F LOWRY EXPRWAY	TEXAS CITY
1	5	28	10-Jan-97	94.00	90.0	TX	3620 EMMETT F LOWRY EXPRWAY	TEXAS CITY
2	5	28	17-Jan-97	543.72	90.0	TX	3620 EMMETT F LOWRY EXPRWAY	TEXAS CITY
3	5	28	24-Jan-97	580.15	90.0	TX	3620 EMMETT F LOWRY EXPRWAY	TEXAS CITY
4	5	28	7-Feb-97	789.20	90.0	TX	3620 EMMETT F LOWRY EXPRWAY	TEXAS CITY
...	...	...	...	...	...	...	...	...
2333	50	588	17-Jul-97	850.00	NaN	TX	12871 INTERSTATE 10	SAN ANTONIO
2334	50	588	24-Jul-97	10127.40	NaN	TX	12871 INTERSTATE 10	SAN ANTONIO
2335	50	588	31-Jul-97	5564.00	NaN	TX	12871 INTERSTATE 10	SAN ANTONIO
2336	50	588	8-Aug-97	1044.35	NaN	TX	12871 INTERSTATE 10	SAN ANTONIO

	Account	Store	ShipDate	SumOfShipwght	Tour	Store_region	Store_address	store_city
2337	50	588	14-Aug-97	7552.40	NaN	TX	12871 INTERSTATE 10	SAN ANTONIO

2338 rows × 8 columns

In [224…

```
#CLEANING THE DATA
data_demand=data_demand.dropna()
data_demand=data_demand[data_demand.SumOfShipwght>0]
data_demand
```

Out[224…

	Account	Store	ShipDate	SumOfShipwght	Tour	Store_region	Store_address	store_city
0	5	28	3-Jan-97	481.28	90.0	TX	3620 EMMETT F LOWRY EXPRWAY	TEXAS CITY
1	5	28	10-Jan-97	94.00	90.0	TX	3620 EMMETT F LOWRY EXPRWAY	TEXAS CITY
2	5	28	17-Jan-97	543.72	90.0	TX	3620 EMMETT F LOWRY EXPRWAY	TEXAS CITY
3	5	28	24-Jan-97	580.15	90.0	TX	3620 EMMETT F LOWRY EXPRWAY	TEXAS CITY
4	5	28	7-Feb-97	789.20	90.0	TX	3620 EMMETT F LOWRY EXPRWAY	TEXAS CITY
...	...	...	...	...	...	...	...	...
2279	50	585	16-Jul-97	10196.09	87.0	TX	2828 SOUTH HWY 6	HOUSTON
2280	50	585	23-Jul-97	6028.69	87.0	TX	2828 SOUTH HWY 6	HOUSTON
2281	50	585	30-Jul-97	4516.90	87.0	TX	2828 SOUTH HWY 6	HOUSTON
2282	50	585	6-Aug-97	11671.88	87.0	TX	2828 SOUTH HWY 6	HOUSTON
2283	50	585	13-Aug-97	1893.70	87.0	TX	2828 SOUTH HWY 6	HOUSTON

1992 rows × 8 columns

In [225…

```
# Only consider Texas Stores
data_demand=data_demand[data_demand.Store_region=='TX']
data_demand
```

Out[225…

	Account	Store	ShipDate	SumOfShipwght	Tour	Store_region	Store_address	store_city

	Account	Store	ShipDate	SumOfShipwght	Tour	Store_region	Store_address	store_city
0	5	28	3-Jan-97	481.28	90.0	TX	3620 EMMETT F LOWRY EXPRWAY	TEXAS CITY
1	5	28	10-Jan-97	94.00	90.0	TX	3620 EMMETT F LOWRY EXPRWAY	TEXAS CITY
2	5	28	17-Jan-97	543.72	90.0	TX	3620 EMMETT F LOWRY EXPRWAY	TEXAS CITY
3	5	28	24-Jan-97	580.15	90.0	TX	3620 EMMETT F LOWRY EXPRWAY	TEXAS CITY
4	5	28	7-Feb-97	789.20	90.0	TX	3620 EMMETT F LOWRY EXPRWAY	TEXAS CITY
...	...	...	...	...	...	...	...	...
2279	50	585	16-Jul-97	10196.09	87.0	TX	2828 SOUTH HWY 6	HOUSTON
2280	50	585	23-Jul-97	6028.69	87.0	TX	2828 SOUTH HWY 6	HOUSTON
2281	50	585	30-Jul-97	4516.90	87.0	TX	2828 SOUTH HWY 6	HOUSTON
2282	50	585	6-Aug-97	11671.88	87.0	TX	2828 SOUTH HWY 6	HOUSTON
2283	50	585	13-Aug-97	1893.70	87.0	TX	2828 SOUTH HWY 6	HOUSTON

1665 rows × 8 columns

In [226…

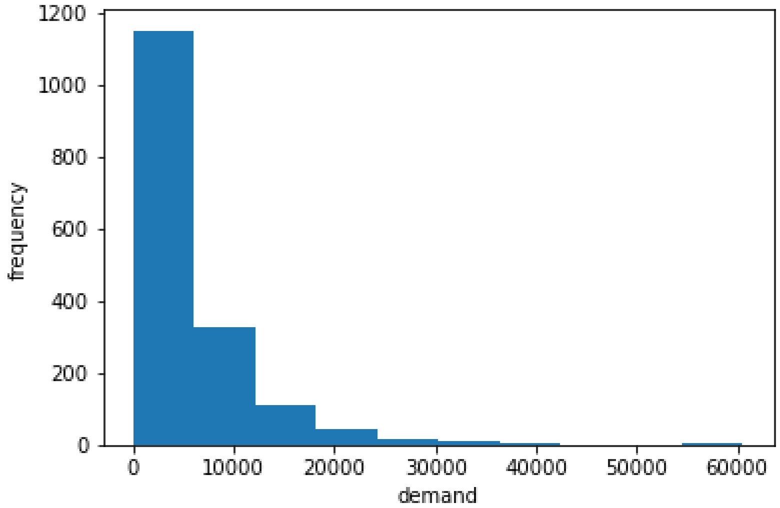
```
# Finding out number of unique stores in our model
a=data_demand.Store.unique()
len(a)
```

Out[226… 51

In [227…

```
#Plotting to visualize the demand distribution for each store order
plt.hist(data_demand['SumOfShipwght'])
plt.xlabel('demand')
plt.ylabel('frequency')
```

Out[227… Text(0, 0.5, 'frequency')

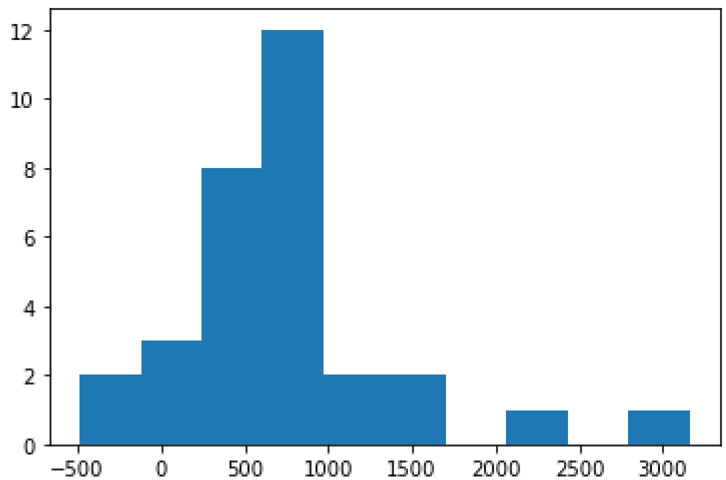


In [254…

```
# Visualizing demand variation of the store types to see if any distribution would be suitable
plt.hist(data_demand[data_demand.Store==28]['SumOfShipwght'])
```

Out[254…

```
(array([ 2.,  3.,  8., 12.,  2.,  2.,  0.,  1.,  0.,  1.]),
 array([-486.88 , -122.275,  242.33 ,  606.935,  971.54 , 1336.145,
        1700.75 , 2065.355, 2429.96 , 2794.565, 3159.17 ]),
 <BarContainer object of 10 artists>)
```



In [255…

```
Source_data=pd.DataFrame(data_demand[["ShipDate","SumOfShipwght"]])
Source_data
```

Out[255…

	ShipDate	SumOfShipwght
0	3-Jan-97	481.28
1	10-Jan-97	94.00
2	17-Jan-97	543.72
3	24-Jan-97	580.15
4	7-Feb-97	789.20
...	...	...
2333	17-Jul-97	850.00
2334	24-Jul-97	10127.40
2335	31-Jul-97	5564.00
2336	8-Aug-97	1044.35
2337	14-Aug-97	7552.40

2338 rows × 2 columns

In [257…

```
# Finding Aggregate weekly demand
```

In [232…

```
from datetime import datetime
Source_data.ShipDate=pd.to_datetime(Source_data.ShipDate)
```

In [233…

```
Source_data['weeklyship']=Source_data["ShipDate"].dt.to_period("W")
```

In [234…

```
aggregation_functions = {'SumOfShipwght': 'sum'}
Source_data_weekly = Source_data.groupby(Source_data['weeklyship']).aggregate(aggregation_functions)
Source_data_weekly
```

Out[234…

	SumOfShipwght
weeklyship	

weeklyship	SumOfShipwght
1996-12-30/1997-01-05	58578.34
1997-01-06/1997-01-12	171909.32
1997-01-13/1997-01-19	201032.87
1997-01-20/1997-01-26	167985.66
1997-01-27/1997-02-02	205277.06
1997-02-03/1997-02-09	219849.82
1997-02-10/1997-02-16	223398.20
1997-02-17/1997-02-23	274383.61
1997-02-24/1997-03-02	280742.76
1997-03-03/1997-03-09	226856.88
1997-03-10/1997-03-16	341424.71
1997-03-17/1997-03-23	291664.85
1997-03-24/1997-03-30	270953.34
1997-03-31/1997-04-06	249244.55
1997-04-07/1997-04-13	391391.20
1997-04-14/1997-04-20	264507.14
1997-04-21/1997-04-27	352017.29
1997-04-28/1997-05-04	297511.79
1997-05-05/1997-05-11	258161.34
1997-05-12/1997-05-18	255204.22
1997-05-19/1997-05-25	381020.27
1997-05-26/1997-06-01	224257.74
1997-06-02/1997-06-08	312174.34
1997-06-09/1997-06-15	281924.36

	SumOfShipwght
weeklyship	
1997-06-16/1997-06-22	417679.69
1997-06-23/1997-06-29	302541.15
1997-06-30/1997-07-06	164032.56
1997-07-07/1997-07-13	323059.71
1997-07-14/1997-07-20	367997.90
1997-07-21/1997-07-27	260738.60
1997-07-28/1997-08-03	289548.27
1997-08-04/1997-08-10	314603.61
1997-08-11/1997-08-17	293685.53

In [258…

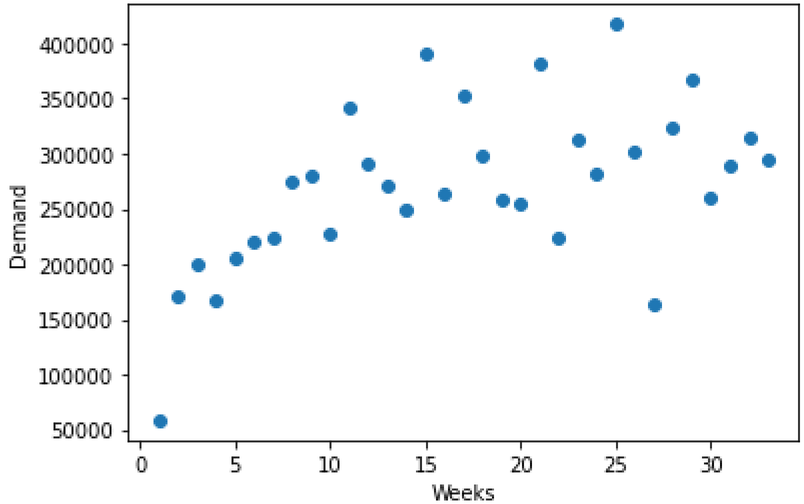
```
# Plotting demand of each week to check if there is any seasonal pattern. From the plot we can say that
# No seasonal Pattern was observed
```

In [252…

```
plt.figsize = (20,30)
a=[i+1 for i in range(len(Source_data_weekly["SumOfShipwght"]))]
plt.scatter(a,Source_data_weekly["SumOfShipwght"])
plt.xlabel("Weeks")
plt.ylabel("Demand")
```

Out[252…

```
Text(0, 0.5, 'Demand')
```

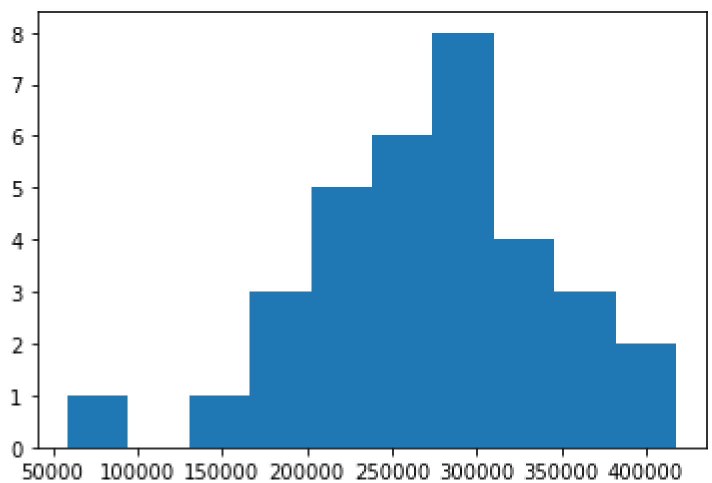


In [259…

```
# Plotting histogram of weekly demand.
# Based on this plot we can observe that the normal distribution might be a good fit
plt.hist(Source_data_weekly["SumOfShipwght"])
```

Out[259…

```
(array([1., 0., 1., 3., 5., 6., 8., 4., 3., 2.]),
 array([ 58578.34 ,  94488.475, 130398.61 , 166308.745, 202218.88 ,
        238129.015, 274039.15 , 309949.285, 345859.42 , 381769.555,
        417679.69 ]),
 <BarContainer object of 10 artists>)
```



Parameter Estimation - For 4 suitable Distribution Candidates

In [262…

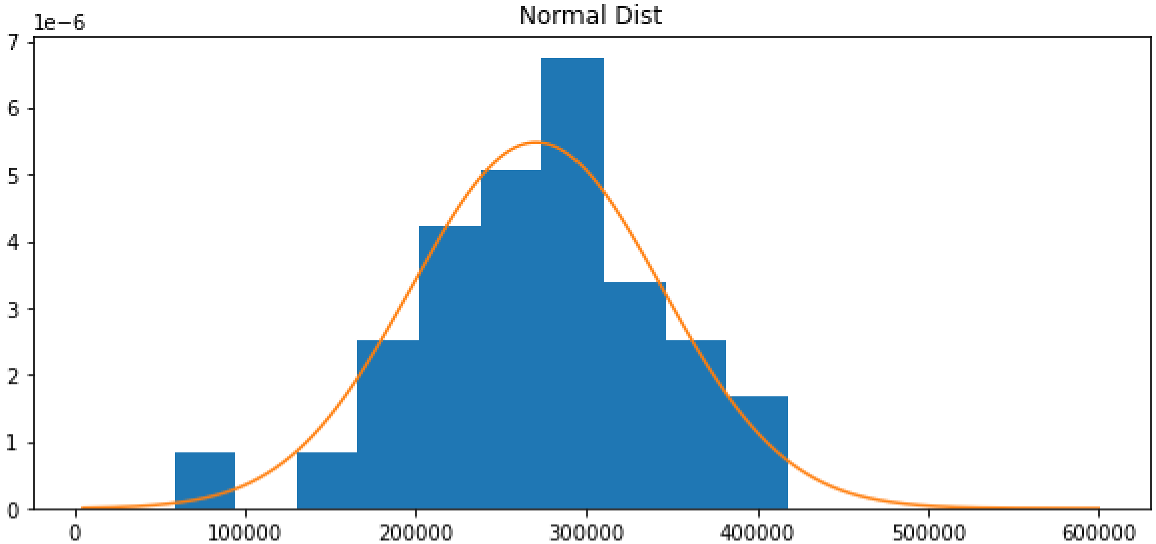
```
# Normal Distribution
u,s=ss.norm.fit(Source_data_weekly.SumOfShipwght)
print(f"The maximum Likelyhood estimators for Normal Distribution are: mean is \
{round(u,3)} and std is {round(s,3)}")

# Gamma Distribution
x,y,z=ss.gamma.fit(Source_data_weekly.SumOfShipwght)
print("The maximum Likelyhood estimators for Gamma Distribution are ",\
      {round(x,3),round(y,3),round(z,3)})

# Triangular Distribution
a,b,c=ss.triang.fit(Source_data_weekly.SumOfShipwght)
print("The maximum Likelyhood estimators for Triangular Distribution are ",\
      {round(a,3),round(b,3),round(c,3)})

# Exponrntial Distribution
m,n =ss.expon.fit(Source_data_weekly.SumOfShipwght)
print(f"The maximum Likelyhood estimators for Uniform Distribution are: mean is \
{round(m,3)} and std is {round(n,3)}")
```

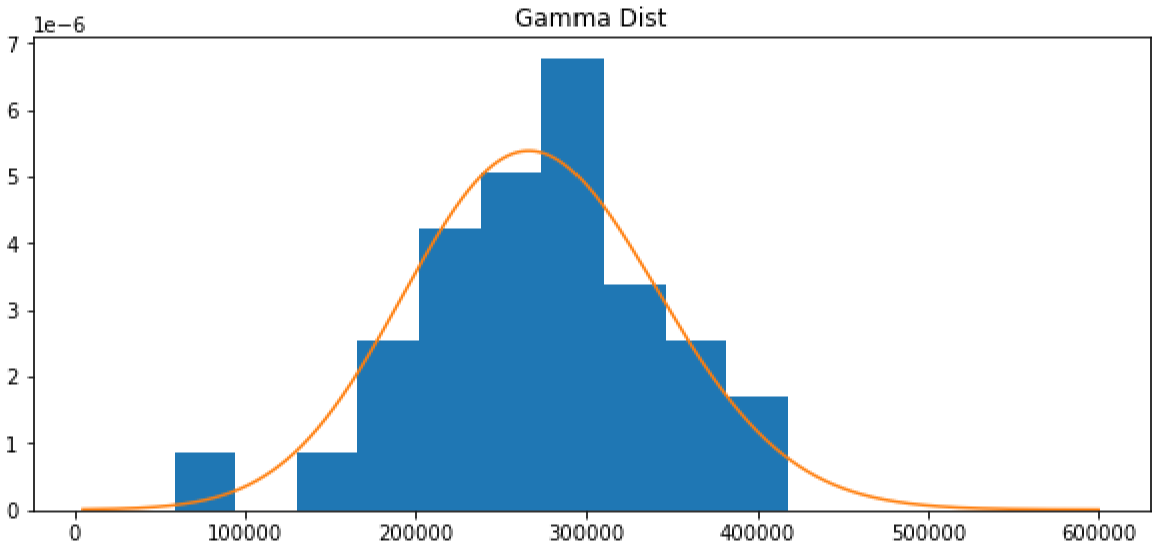
```
The maximum Likelyhood estimators for Normal Distribution are: mean is 270768.445 and std is 72784.715
The maximum Likelyhood estimators for Gamma Distribution are  {346.534, 3987.604, -1111324.743}
The maximum Likelyhood estimators for Triangular Distribution are  {0.628, 397721.925, 42054.049}
The maximum Likelyhood estimators for Uniform Distribution are: mean is 58578.34 and std is 212190.105
```

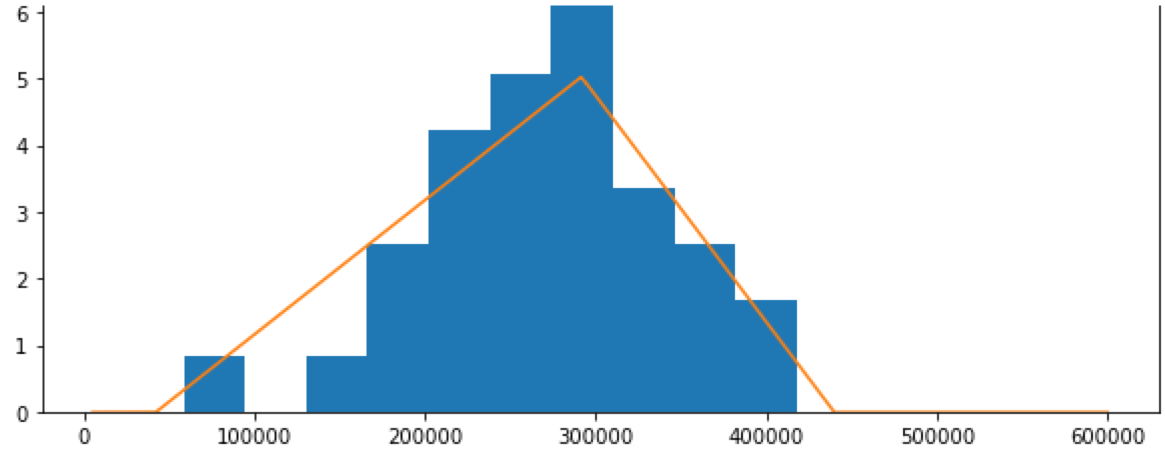
In [264…

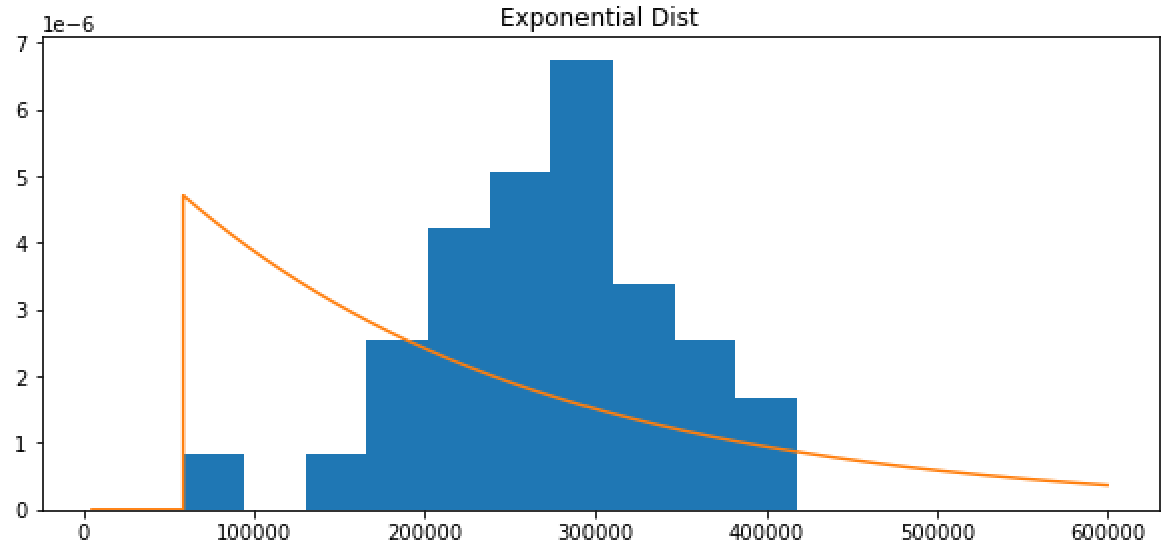
```
# Plotting Hypothesied Probable distributions over demand data
fig, (ax1,ax2,ax3,ax4) = plt.subplots(4,1, figsize = (10,20))

ax1.hist(Source_data_weekly.SumOfShipwght,density=True,bins=10)
x_seq = np.arange(5000,600000,1)
y_seq = ss.norm.pdf(x_seq,u,s)
ax1.plot(x_seq, y_seq)
ax1.title.set_text('Normal Dist')

ax2.hist(Source_data_weekly.SumOfShipwght,density=True,bins=10)
x1_seq = np.arange(5000,600000,1)
y1_seq = ss.gamma.pdf(x1_seq,x,y,z)
ax2.plot(x1_seq, y1_seq)
ax2.title.set_text('Gamma Dist')

ax3.hist(Source_data_weekly.SumOfShipwght,density=True,bins=10)
x2_seq = np.arange(5000,600000,1)
y2_seq = ss.triang.pdf(x2_seq,a,b,c)
ax3.plot(x2_seq, y2_seq)
ax3.title.set_text('Triangular Dist')
```

```
ax4.hist(Source_data_weekly.SumOfShipwght,density=True,bins=10)
x3_seq = np.arange(5000,600000,1)
y3_seq = ss.expon.pdf(x1_seq,m,n)
ax4.plot(x3_seq, y3_seq)
ax4.title.set_text('Exponential Dist')
```









In [266…

```
# We know that demand can never be negative unless its returns which we are not considering in this case.
# so another important check to find if Nomrla distribition id good fit or not is to check if 0 falls at
# least 3 standard deviastions away from
# mean

# Xi = μ ± Zσ
# so to calculate Z (no. of standard deviations from mean)
Z=(u-0)/s
Z
#So as 0 is almost 4 standard deviations away from mean so the probability of having -ve demand will be almost negligible
#Hence normal distribution can be used
```

Out[266… 3.720127856182613

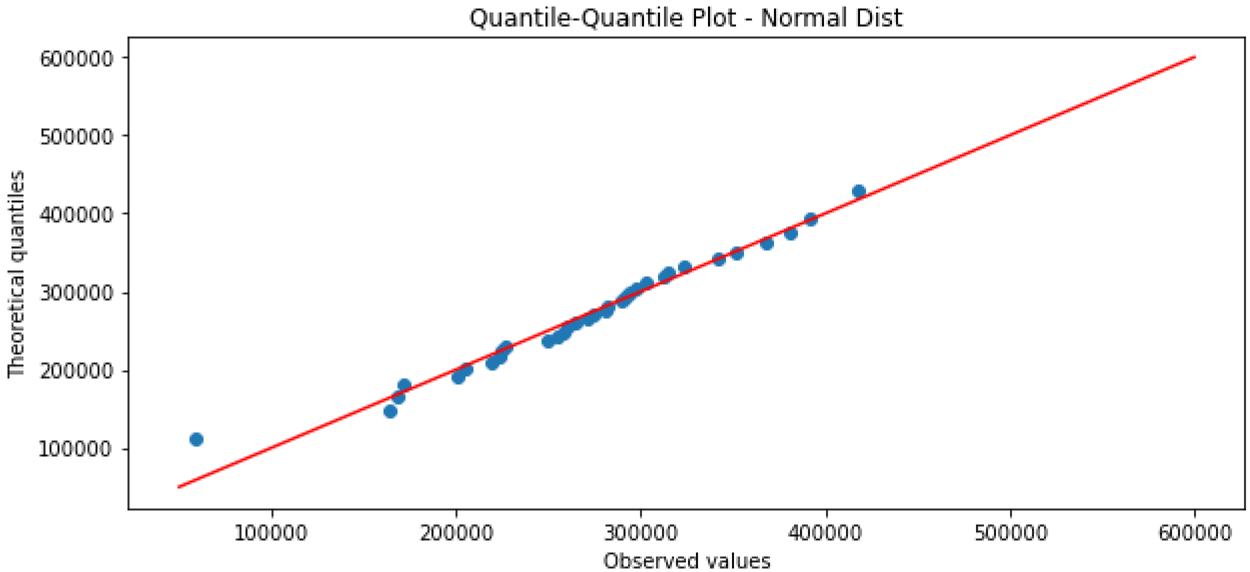
Q-Q plot

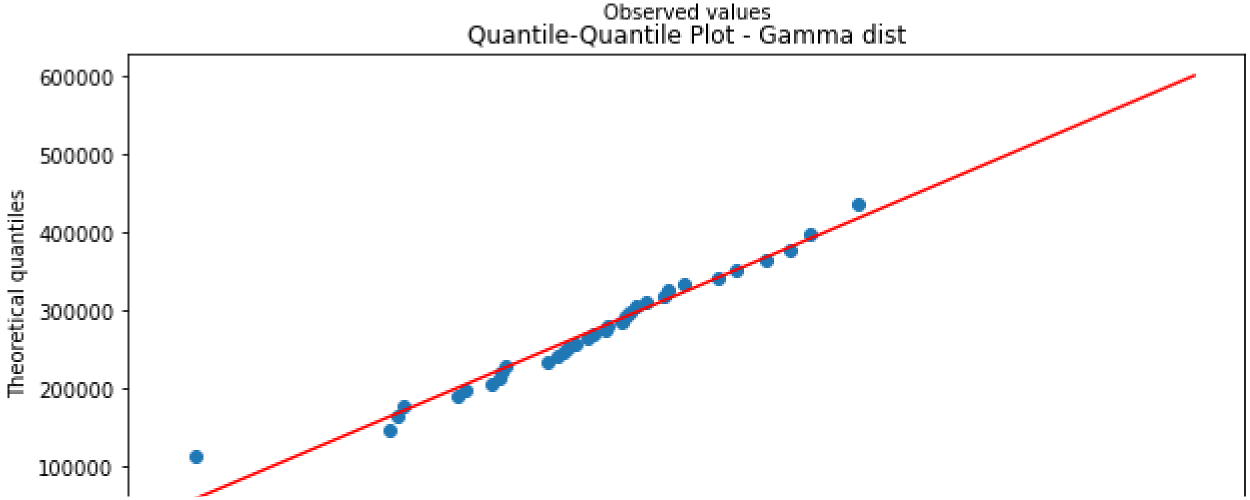
In [269…

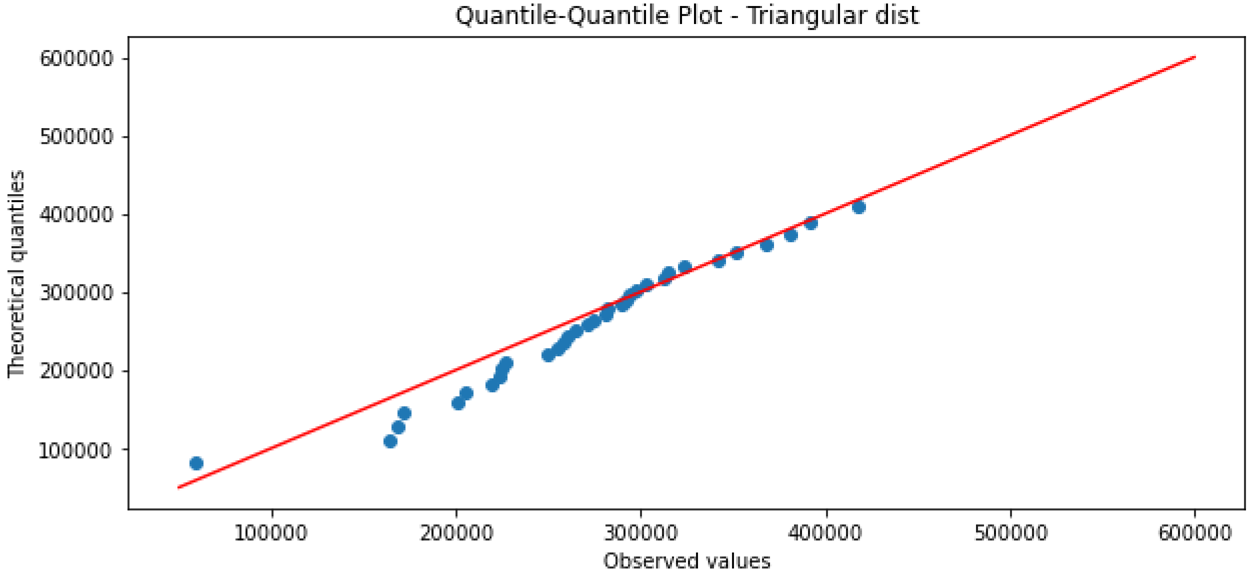
```
fig, (ax1,ax2,ax3,ax4) = plt.subplots(4,1, figsize = (10,20))

# Normal Distributions
normal_estimate = ss.norm(u, s)
# Compute the theoretical q-quantiles for q = (i-.5)/n for i = 1, ..., n
normal_quantile = normal_estimate.ppf((np.arange(1,len(Source_data_weekly.SumOfShipwght)+1)-0.5)/len(Source_data_weekly.S
# Sort the sample data in ascending order
Source_data_weekly.SumOfShipwght = np.sort(Source_data_weekly.SumOfShipwght)
ax1.scatter(Source_data_weekly.SumOfShipwght, normal_quantile)
ax1.set_xlabel('Observed values')
ax1.set_ylabel('Theoretical quantiles')
ax1.set_title('Quantile-Quantile Plot - Normal Dist')
ax1.plot([50000,600000], [50000,600000], color='red')
print("as the scatter plot for Normal Distribution almost lies on the straight line of sloper 45 degreeThus its a good fi

# Gamma Distributions
gamma_estimate = ss.gamma(x,y,z)
# Compute the theoretical q-quantiles for q = (i-.5)/n for i = 1, ..., n
gamma_quantile = gamma_estimate.ppf((np.arange(1,len(Source_data_weekly.SumOfShipwght)+1)-0.5)/len(Source_data_weekly.Sum
# Sort the sample data in ascending order
Source_data_weekly.SumOfShipwght = np.sort(Source_data_weekly.SumOfShipwght)
ax2.scatter(Source_data_weekly.SumOfShipwght, gamma_quantile)
ax2.set_xlabel('Observed values')
ax2.set_ylabel('Theoretical quantiles')
ax2.set_title('Quantile-Quantile Plot - Gamma dist')
ax2.plot([50000,600000], [50000,600000], color='red')

# Triangular Distributions
tri_estimate = ss.triang(a,b,c)
# Compute the theoretical q-quantiles for q = (i-.5)/n for i = 1, ..., n
tri_quantile = tri_estimate.ppf((np.arange(1,len(Source_data_weekly.SumOfShipwght)+1)-0.5)/len(Source_data_weekly.SumOfSh
# Sort the sample data in ascending order
Source_data_weekly.SumOfShipwght = np.sort(Source_data_weekly.SumOfShipwght)
ax3.scatter(Source_data_weekly.SumOfShipwght, tri_quantile)
ax3.set_xlabel('Observed values')
ax3.set_ylabel('Theoretical quantiles')
ax3.set_title('Quantile-Quantile Plot - Triangular dist')
ax3.plot([50000,600000], [50000,600000], color='red')

# Exponential Distributions
```

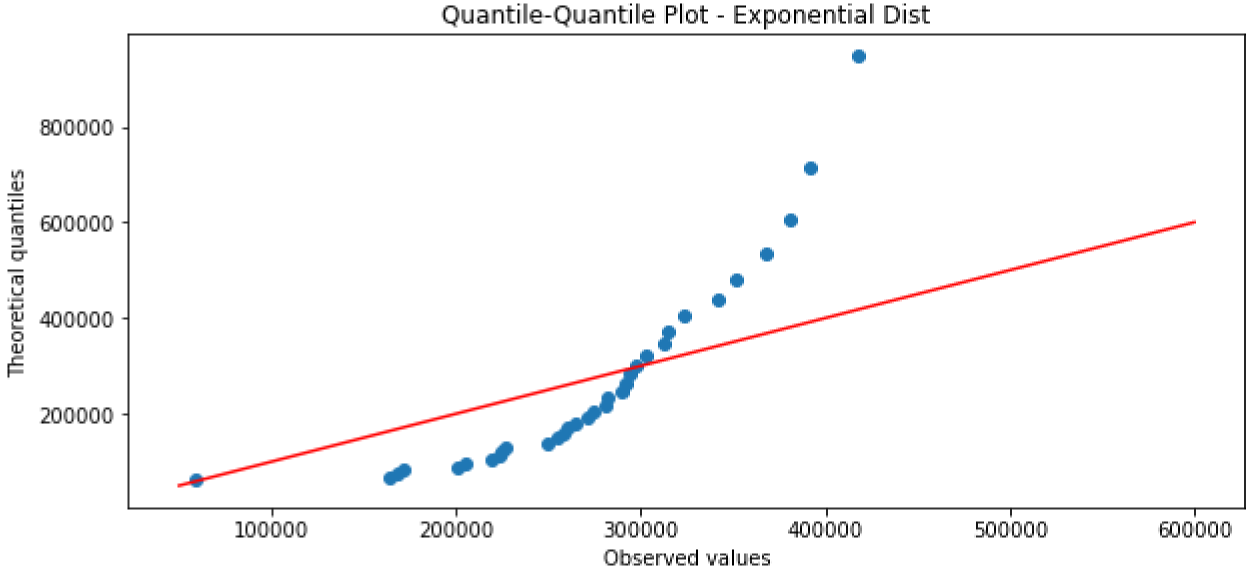
```
expon_estimate = ss.expon(m,n)
# Compute the theoretical q-quantiles for q = (i-.5)/n for i = 1, ..., n
expon_quantile = expon_estimate.ppf((np.arange(1,len(Source_data_weekly.SumOfShipwght)+1)-0.5)/len(Source_data_weekly.Sum
# Sort the sample data in ascending order
Source_data_weekly.SumOfShipwght = np.sort(Source_data_weekly.SumOfShipwght)
ax4.scatter(Source_data_weekly.SumOfShipwght, expon_quantile)
ax4.set_xlabel('Observed values')
ax4.set_ylabel('Theoretical quantiles')
ax4.set_title('Quantile-Quantile Plot - Exponential Dist')
ax4.plot([50000,600000], [50000,600000], color='red')
plt.show()
```

as the scatter plot for Normal Distribution almost lies on the straight line of sloper 45 degreeThus its a good fit









In [296…

```
#Performimng K-S test to check if its normal Distribution or not
u,s= ss.norm.fit(Source_data_weekly.SumOfShipwght)
normal_estimate = ss.norm(u,s)
statistics, pvalue = ss.kstest(rvs = Source_data_weekly.SumOfShipwght, cdf = normal_estimate.cdf)
print(f"The p-value for the K-S test for normal distribution is {round(pvalue, 4)}.")
```

```
The p-value for the K-S test for normal distribution is 0.9663.
```

In [297…

```
u,s,m= ss.gamma.fit(Source_data_weekly.SumOfShipwght)
gamma_estimate = ss.gamma(u,s,m)
statistics, pvalue = ss.kstest(rvs = Source_data_weekly.SumOfShipwght, cdf = gamma_estimate.cdf)
print(f"The p-value for the K-S test was for gamma distribution is {round(pvalue, 4)}.")
```

```
The p-value for the K-S test was for gamma distribution is 0.9206.
```

In [298…

```
u,s= ss.expon.fit(Source_data_weekly.SumOfShipwght)
expon_estimate = ss.expon(u,s)
statistics, pvalue = ss.kstest(rvs = Source_data_weekly.SumOfShipwght, cdf = expon_estimate.cdf)
print(f"The p-value for the K-S test was for Exponrential Distribution is {round(pvalue, 4)}.")
```

```
The p-value for the K-S test was for Exponrential Distribution is 0.0002.
```

In [299…

```
u,s,m= ss.triang.fit(Source_data_weekly.SumOfShipwght)
triang_estimate = ss.triang(u,s,m)
statistics, pvalue = ss.kstest(rvs = Source_data_weekly.SumOfShipwght, cdf = triang_estimate.cdf)
print(f"The p-value for the K-S test was for triangular Distribution is  {round(pvalue, 4)}.")
```

```
The p-value for the K-S test was for triangular Distribution is  0.5249.
```

As p-value from K-s Test for Normal Distribution is very high therefore Demand weekly data can be approximately assumed as Normal Distribution

STORE Demand Distribution

Trying find a good distribution fit for each of the stores present in the system

In [271…

```
data_demand
```

Out[271…

	Account	Store	ShipDate	SumOfShipwght	Tour	Store_region	Store_address	store_city
0	5	28	3-Jan-97	481.28	90.0	TX	3620 EMMETT F LOWRY EXPRWAY	TEXAS CITY
1	5	28	10-Jan-97	94.00	90.0	TX	3620 EMMETT F LOWRY EXPRWAY	TEXAS CITY
2	5	28	17-Jan-97	543.72	90.0	TX	3620 EMMETT F LOWRY EXPRWAY	TEXAS CITY
3	5	28	24-Jan-97	580.15	90.0	TX	3620 EMMETT F LOWRY EXPRWAY	TEXAS CITY
4	5	28	7-Feb-97	789.20	90.0	TX	3620 EMMETT F LOWRY EXPRWAY	TEXAS CITY

	Account	Store	ShipDate	SumOfShipwght	Tour	Store_region	Store_address	store_city
...	...	...	...	...	...	...	...	...
2333	50	588	17-Jul-97	850.00	NaN	TX	12871 INTERSTATE 10	SAN ANTONIO
2334	50	588	24-Jul-97	10127.40	NaN	TX	12871 INTERSTATE 10	SAN ANTONIO
2335	50	588	31-Jul-97	5564.00	NaN	TX	12871 INTERSTATE 10	SAN ANTONIO
2336	50	588	8-Aug-97	1044.35	NaN	TX	12871 INTERSTATE 10	SAN ANTONIO
2337	50	588	14-Aug-97	7552.40	NaN	TX	12871 INTERSTATE 10	SAN ANTONIO

2338 rows × 8 columns

Checking if Normal Distribution for all stores is good fit or not

In [283…

```
norm_dict={}
# trying to check if normal distribution will be a good fir for any of the stores or not
#for i in data_demand.Tour.unique():
for j in data_demand.Store.unique():
    #a=pd.DataFrame(data_demand.SumOfShipwght,data_demand.Store)
    d=data_demand[data_demand['Store']==j]
    a=pd.DataFrame(d.SumOfShipwght)
    u,s=ss.norm.fit(a.SumOfShipwght)
    #print(f"The MLE for Normal Distribution for store {j} is are: mean is \
    #{round(u,3)} and std is {round(s,3)}")
    norm_dict[j]=[u,s]
```

In [285…

```
for i in data_demand.Store.unique():
    u,s=(norm_dict[i][0],norm_dict[i][1])
    normal_estimate = ss.norm(u,s)
    d=data_demand[data_demand['Store']==j]
    a=pd.DataFrame(d.SumOfShipwght)
    statistics, pvalue = ss.kstest(rvs = a.SumOfShipwght, cdf = normal_estimate.cdf)
    print(f"The p-value for the store {i} was {round(pvalue, 4)}.")
```

```
The p-value for the store 28 was 0.0.
The p-value for the store 82 was 0.0232.
The p-value for the store 95 was 0.0.
The p-value for the store 97 was 0.0.
The p-value for the store 98 was 0.0.
The p-value for the store 103 was 0.0.
The p-value for the store 123 was 0.0.
The p-value for the store 137 was 0.0.
The p-value for the store 159 was 0.0.
The p-value for the store 161 was 0.0.
The p-value for the store 186 was 0.0.
The p-value for the store 232 was 0.0.
The p-value for the store 234 was 0.0.
The p-value for the store 259 was 0.0.
The p-value for the store 282 was 0.0.
The p-value for the store 365 was 0.0.
The p-value for the store 461 was 0.0.
The p-value for the store 484 was 0.0004.
The p-value for the store 501 was 0.0368.
The p-value for the store 1005 was 0.1291.
The p-value for the store 1006 was 0.0323.
The p-value for the store 1007 was 0.3825.
The p-value for the store 1036 was 0.1726.
The p-value for the store 1340 was 0.377.
The p-value for the store 1341 was 0.0806.
The p-value for the store 1344 was 0.1077.
The p-value for the store 1346 was 0.0345.
The p-value for the store 1348 was 0.0165.
The p-value for the store 1351 was 0.0001.
The p-value for the store 1373 was 0.0205.
The p-value for the store 1401 was 0.1187.
The p-value for the store 1409 was 0.5709.
The p-value for the store 1411 was 0.0083.
The p-value for the store 1421 was 0.1359.
The p-value for the store 1422 was 0.1728.
The p-value for the store 1561 was 0.5248.
The p-value for the store 1583 was 0.0194.
The p-value for the store 1584 was 0.2327.
The p-value for the store 1590 was 0.0555.
The p-value for the store 349 was 0.0029.
The p-value for the store 351 was 0.0004.
The p-value for the store 352 was 0.0.
The p-value for the store 357 was 0.0006.
The p-value for the store 358 was 0.0075.
The p-value for the store 359 was 0.0003.
The p-value for the store 360 was 0.0019.
The p-value for the store 362 was 0.0.
The p-value for the store 502 was 0.0041.
The p-value for the store 503 was 0.0058.
```

```
The p-value for the store 504 was 0.027.
The p-value for the store 520 was 0.1695.
The p-value for the store 521 was 0.0478.
The p-value for the store 526 was 0.0631.
The p-value for the store 565 was 0.0789.
The p-value for the store 566 was 0.0069.
The p-value for the store 567 was 0.0841.
The p-value for the store 568 was 0.1769.
The p-value for the store 569 was 0.0031.
The p-value for the store 571 was 0.1273.
The p-value for the store 573 was 0.0259.
The p-value for the store 574 was 0.7205.
The p-value for the store 576 was 0.0352.
The p-value for the store 577 was 0.0001.
The p-value for the store 578 was 0.3423.
The p-value for the store 579 was 0.1693.
The p-value for the store 580 was 0.0962.
The p-value for the store 581 was 0.4029.
The p-value for the store 582 was 0.3647.
The p-value for the store 584 was 0.0091.
The p-value for the store 585 was 0.1919.
The p-value for the store 586 was 0.0864.
The p-value for the store 588 was 0.194.
```

In [275…

```
#As we can observe the p-value of the stores assuming Normal Distribution is very small, so normal distribution is not a
```

Checking if Gamma Distribution for all stortes is good fit or not

In [282…

```
gamma_dict={}
# trying to check if normal distribution will be a good fir for any of the stores or not

for j in data_demand.Store.unique():
    #a=pd.DataFrame(data_demand.SumOfShipwght,data_demand.Store)
    d=data_demand[data_demand['Store']==j]
    a=pd.DataFrame(d.SumOfShipwght)
    u,s,m=ss.gamma.fit(a.SumOfShipwght)
    #print(f"The MLE for Normal Distribution for store {j} is are: mean is \
    #{round(u,3)} and std is {round(s,3)}")
    gamma_dict[j]=[u,s,m]
```

In [288…

```
for i in data_demand.Store.unique():
    u,s,m=(gamma_dict[i][0],gamma_dict[i][1],gamma_dict[i][2])
    gamma_estimate = ss.gamma(u,s,m)
    d=data_demand[data_demand['Store']==j]
```

```
    a=pd.DataFrame(d.SumOfShipwght)
    statistics, pvalue = ss.kstest(rvs = a.SumOfShipwght, cdf = gamma_estimate.cdf)
    print(f"The p-value for the store {i} was {round(pvalue, 4)}.")
```

```
The p-value for the store 28 was 0.0008.
The p-value for the store 82 was 0.0001.
The p-value for the store 95 was 0.0.
The p-value for the store 97 was 0.0.
The p-value for the store 98 was 0.0.
The p-value for the store 103 was 0.0001.
The p-value for the store 123 was 0.0.
The p-value for the store 137 was 0.0.
The p-value for the store 159 was 0.0.
The p-value for the store 161 was 0.0.
The p-value for the store 186 was 0.0.
The p-value for the store 232 was 0.0.
The p-value for the store 234 was 0.0.
The p-value for the store 259 was 0.0.
The p-value for the store 282 was 0.0.
The p-value for the store 365 was 0.0.
The p-value for the store 461 was 0.0001.
The p-value for the store 484 was 0.0004.
The p-value for the store 501 was 0.7467.
The p-value for the store 1005 was 0.0862.
The p-value for the store 1006 was 0.0334.
The p-value for the store 1007 was 0.2106.
The p-value for the store 1036 was 0.6809.
The p-value for the store 1340 was 0.2885.
The p-value for the store 1341 was 0.1657.
The p-value for the store 1344 was 0.1632.
The p-value for the store 1346 was 0.0387.
The p-value for the store 1348 was 0.1721.
The p-value for the store 1351 was 0.0.
The p-value for the store 1373 was 0.0156.
The p-value for the store 1401 was 0.2162.
The p-value for the store 1409 was 0.0.
The p-value for the store 1411 was 0.0031.
The p-value for the store 1421 was 0.0.
The p-value for the store 1422 was 0.2018.
The p-value for the store 1561 was 0.9726.
The p-value for the store 1583 was 0.011.
The p-value for the store 1584 was 0.0.
The p-value for the store 1590 was 0.0564.
The p-value for the store 349 was 0.0.
The p-value for the store 351 was 0.0007.
The p-value for the store 352 was 0.0001.
The p-value for the store 357 was 0.0007.
The p-value for the store 358 was 0.0504.
```

```
The p-value for the store 359 was 0.4229.
The p-value for the store 360 was 0.0071.
The p-value for the store 362 was 0.0495.
The p-value for the store 502 was 0.1684.
The p-value for the store 503 was 0.006.
The p-value for the store 504 was 0.0404.
The p-value for the store 520 was 0.9947.
The p-value for the store 521 was 0.4804.
The p-value for the store 526 was 0.0.
The p-value for the store 565 was 0.437.
The p-value for the store 566 was 0.9723.
The p-value for the store 567 was 0.6352.
The p-value for the store 568 was 0.781.
The p-value for the store 569 was 0.2281.
The p-value for the store 571 was 0.9603.
The p-value for the store 573 was 0.5683.
The p-value for the store 574 was 0.7214.
The p-value for the store 576 was 0.4963.
The p-value for the store 577 was 0.0296.
The p-value for the store 578 was 0.951.
The p-value for the store 579 was 0.3227.
The p-value for the store 580 was 0.8347.
The p-value for the store 581 was 0.8992.
The p-value for the store 582 was 0.2076.
The p-value for the store 584 was 0.7512.
The p-value for the store 585 was 0.5915.
The p-value for the store 586 was 0.5285.
The p-value for the store 588 was 0.991.
```

In [290…

```
#As we can observe the p-value of the stores assuming Gamma Distribution is very small, so Gamma distribution is not a go
```

For the store level, Tried a lot of distributions but no distrubution was coming as proper fit thats why we used emperical data only.

Performed Trace dfriven simulation and generated Linearly interpolated ECDF.