**COL334 Assignment 3**

**Packet Trace Analysis**

Vidit Jain 2017CS10389

Rahul Yadav 2017CS50602

< Different FTP commands>

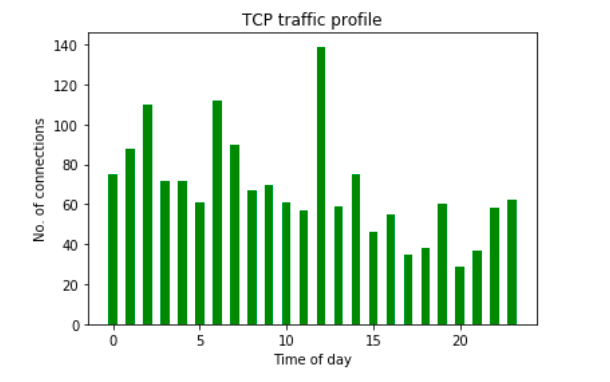
1. All TCP data was considered. All rows having [SYN] in their Info column

|  |  |  |
| --- | --- | --- |
| **Day** | **Unique server IPs** | **Unique Client IPs** |
| 1 | 45 | 522 |
| 2 | 50 | 939 |
| 3 | 89 | 510 |

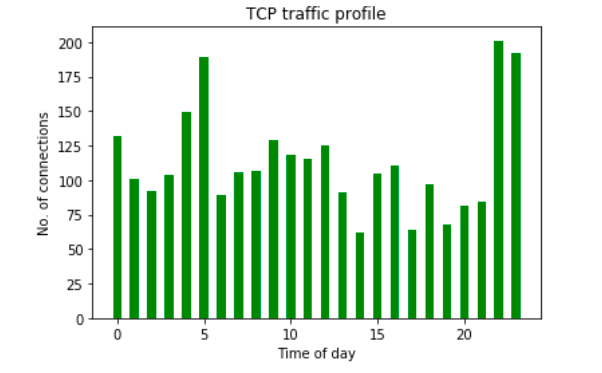
|  |  |
| --- | --- |
| **Day** | **Unique Flows** |
| 1 | 3256 |
| 2 | 5422 |
| 3 | 3280 |

1. All unique 4-tuples of the form <source IP, destination IP, source port, destination port> were considered for calculations, ie, for each exchange between a client and server, there are 2 different TCP flows.
2. Traffic flows for each day

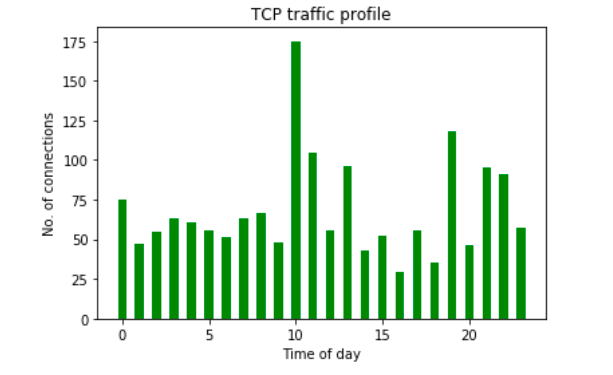
Day 1



Day 2



Day 3



To check whether a server is under a DOS attack, we can maintain a **weighted history of the traffic** expected at any given point of time, in form the TCP connections opened. If the difference in the TCP connections opened to the server at to the TCP connections expected **is more than a certain threshold,** we can claim that a DOS attack is ongoing.

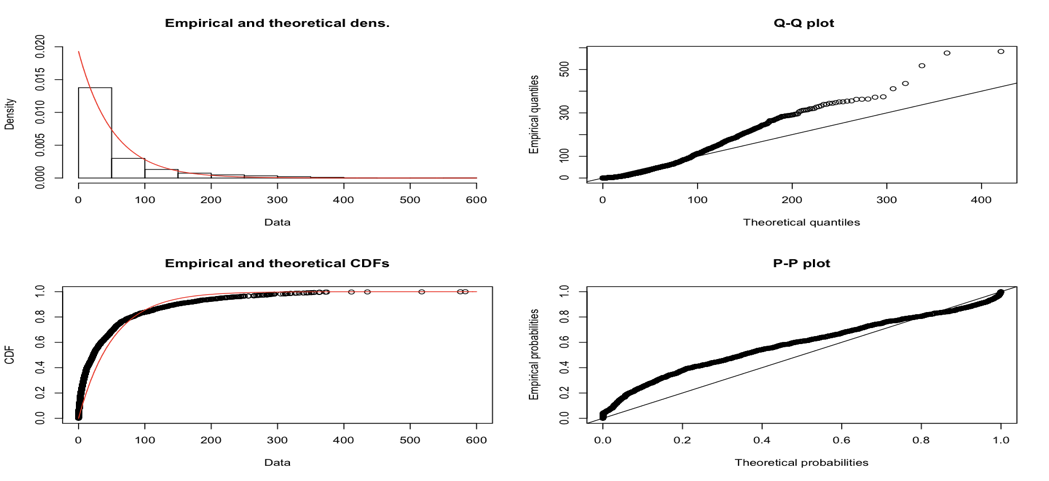
1. 3 tables were created, Table 1 with all TCP SYN packet transmissions, Table 2 with FIN and Table 3 with RST packet transmissions. For each transmission in Table 1, the earliest transmission after this one was found in Table 2 and Table 3, such that the IP addresses communicating matched with that of the transmission from Table 1. This gave us the time when this particular TCP connection ended. This was done for all packet transmissions in Table 1, thus excluding those whose start/end time wasn’t captured in the packet trace. This also took care of the different connections between same ports.

10.

**DAY1**

* **Inter-Arrival Distribution between 2 consecutive connections**

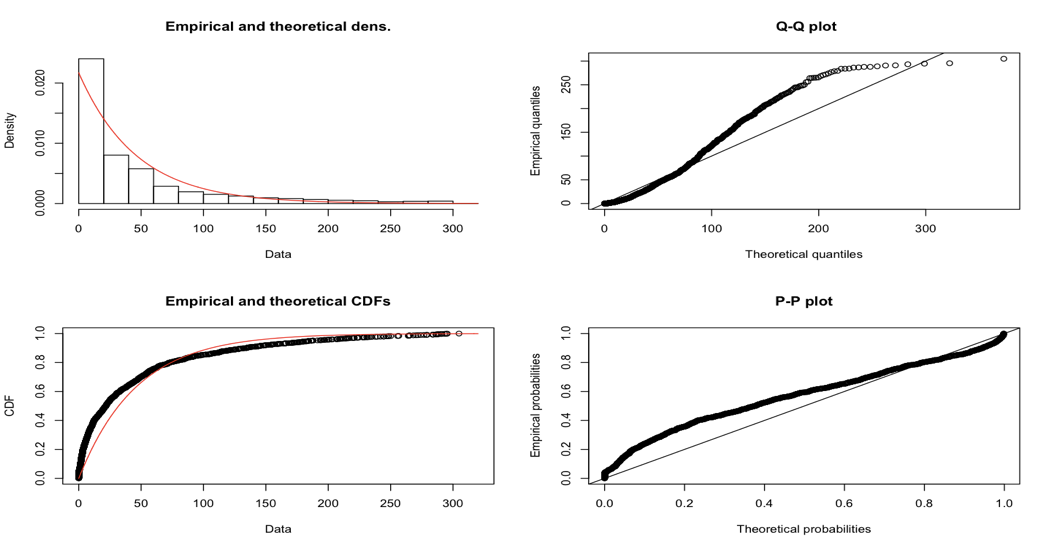
With outliers



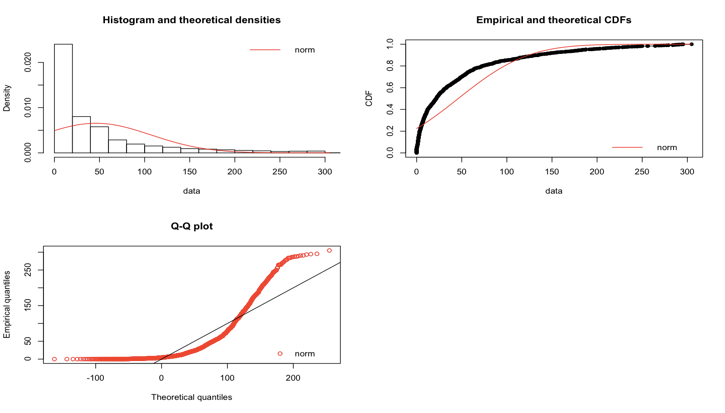
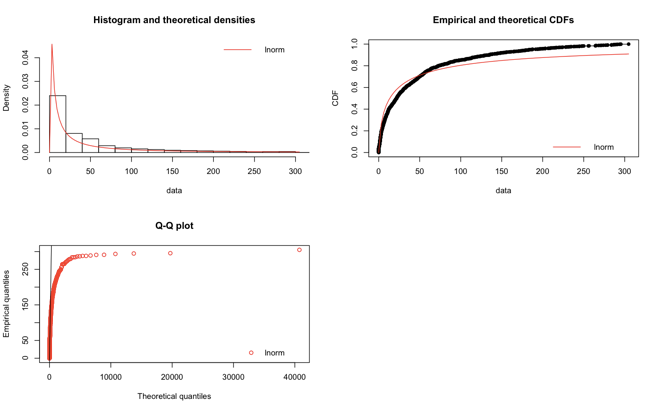
Exponential distribution Parameters:

Rate : 0.0192

After removing outliers :



Other Distributions on given dataset :

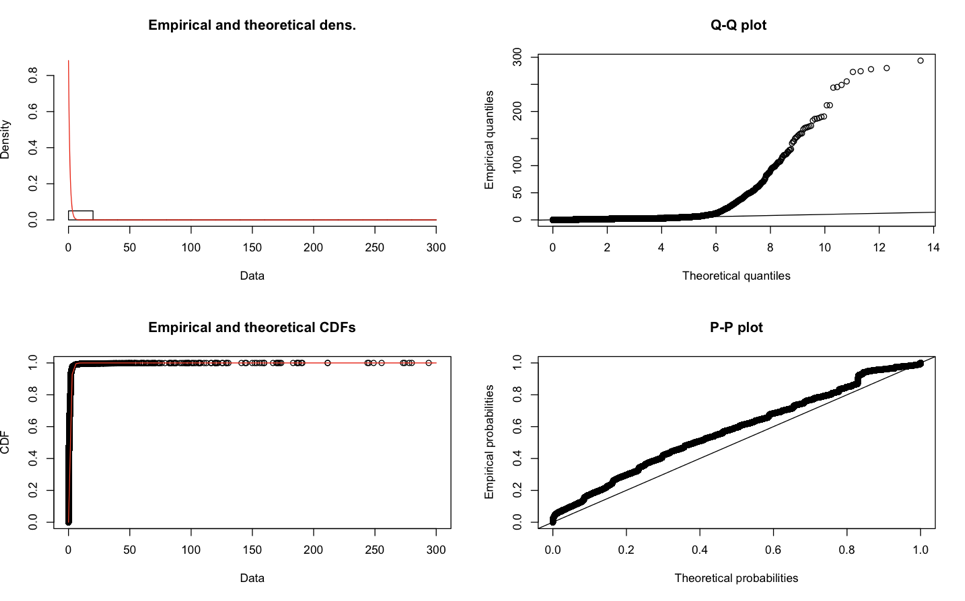
Normal distribution Parameters: Log - Normal distribution Parameters:

mean : 46.07 meanlog : 2.60

sd : 60.91 sdlog 2.33

* **Inter-Arrival Distribution between 2 consecutive incoming packets to server**

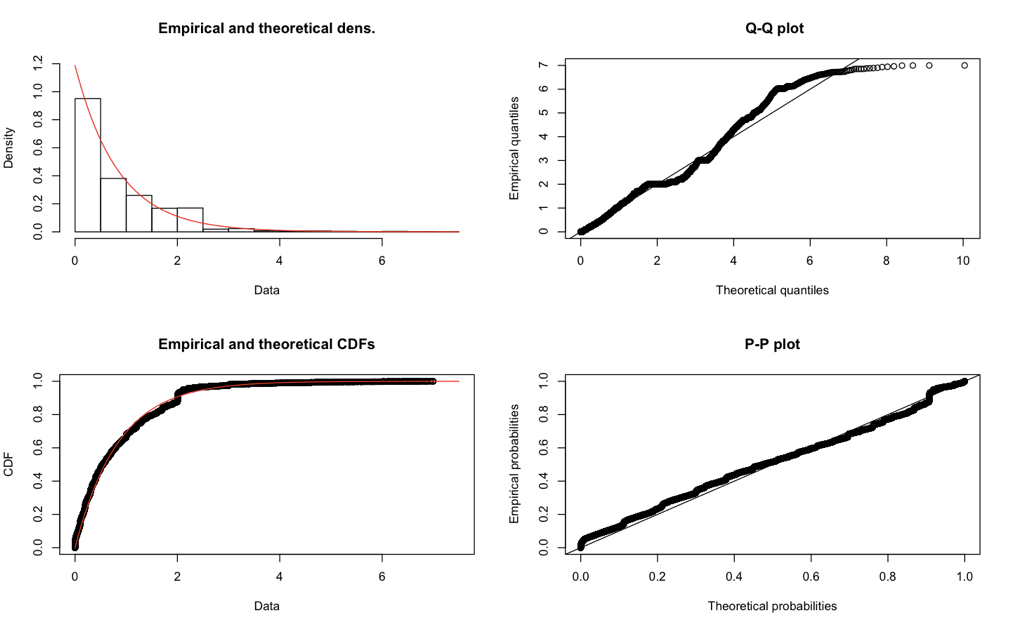
With outliers



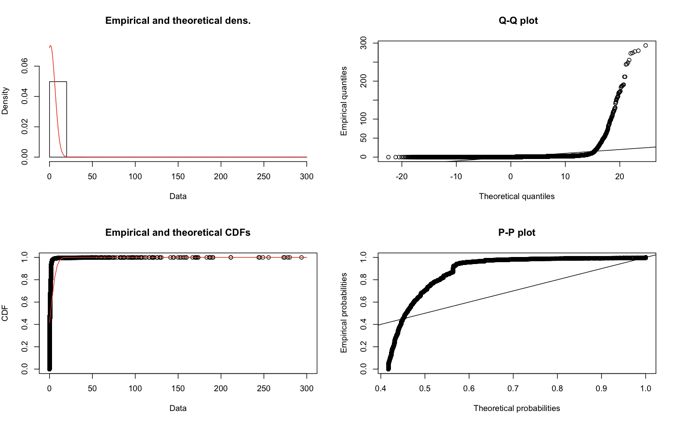
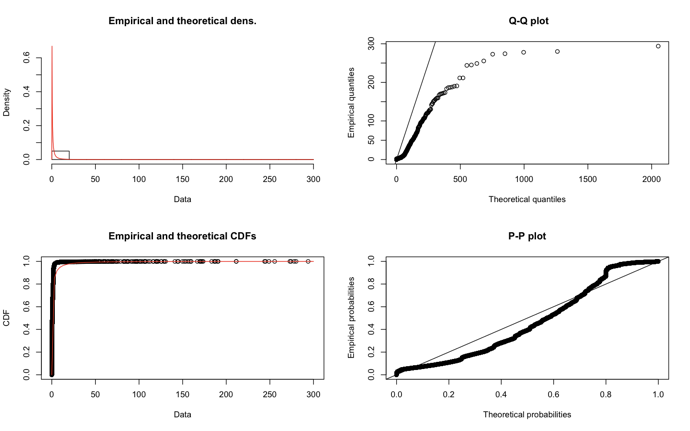
Exponential distribution Parameters:

Rate : 0.88

After removing outliers :



Other Distributions on given dataset :

Normal distribution Parameters: Log - Normal distribution Parameters:

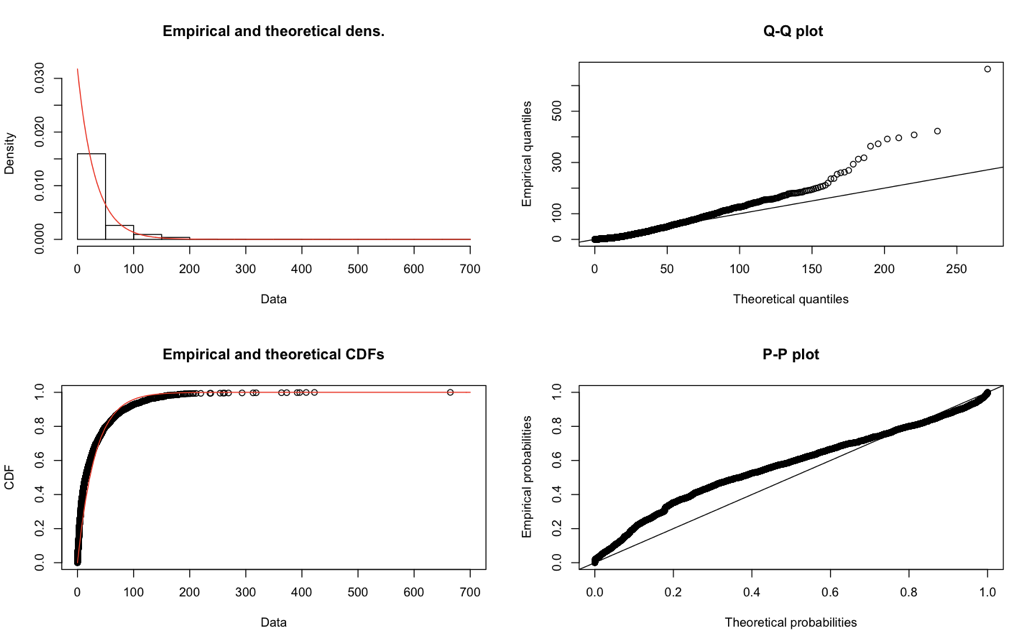
mean : 1.13 meanlog : -0.96

sd : 5.41 sdlog : 1.97

**DAY2**

* **Inter-Arrival Distribution between 2 consecutive connections**

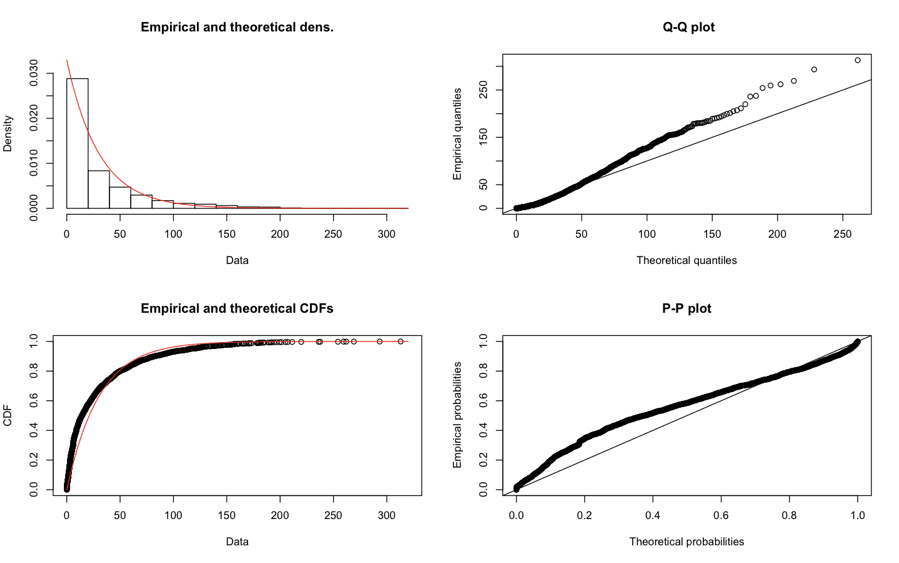
With outliers



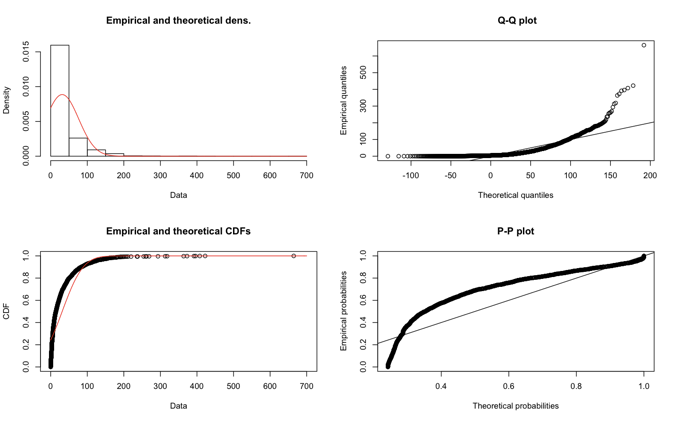
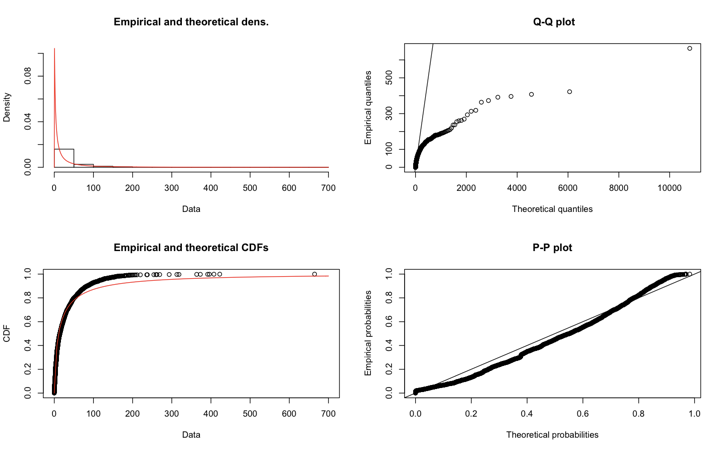
Exponential distribution Parameters:

Rate : 0.031

After removing outliers :



Other Distributions on given dataset :

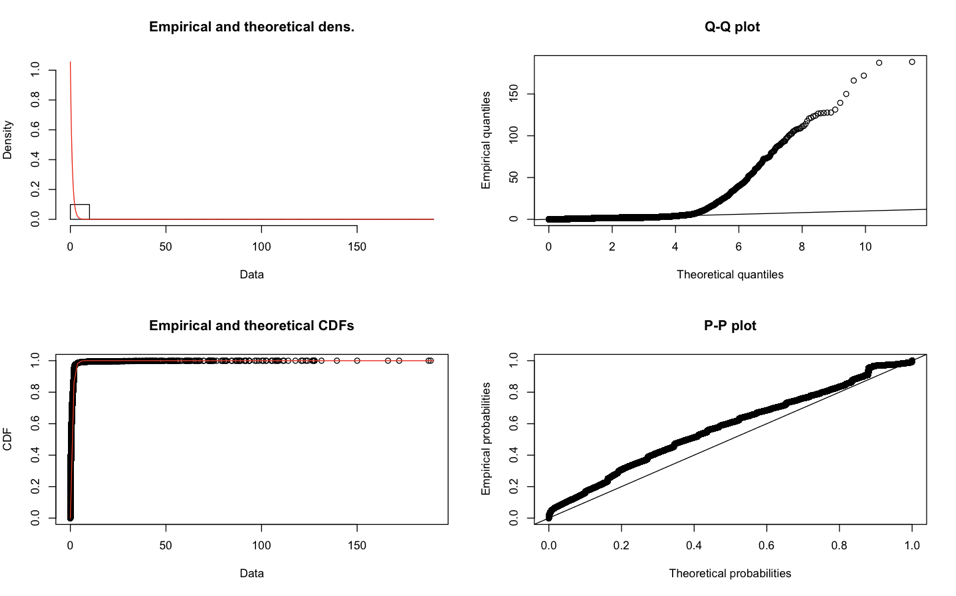
Normal distribution Parameters: Log - Normal distribution Parameters:

mean : 31.50 meanlog : 2.42

sd : 45.02 sdlog 1.92

* **Inter-Arrival Distribution between 2 consecutive incoming packets to server**

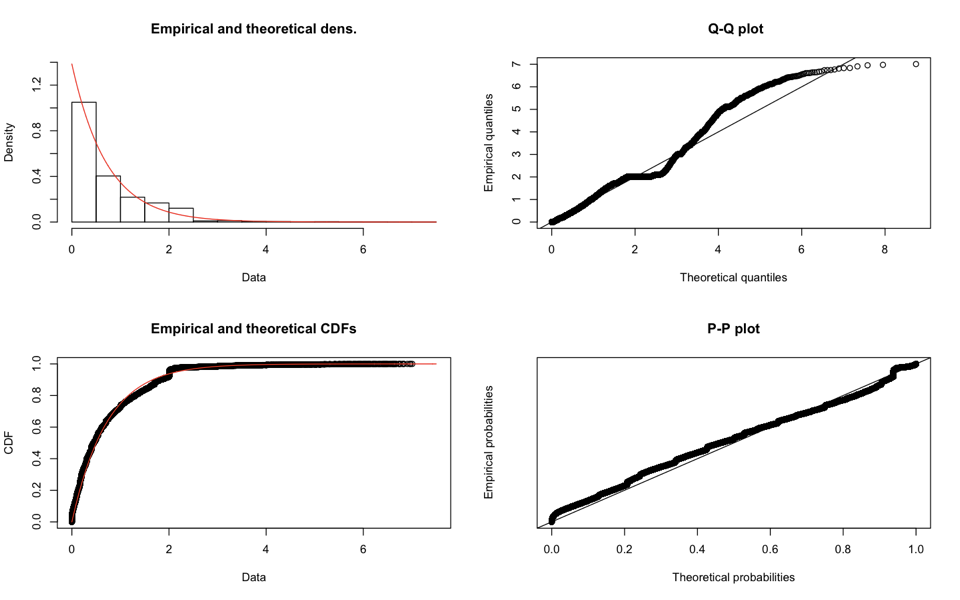
With outliers



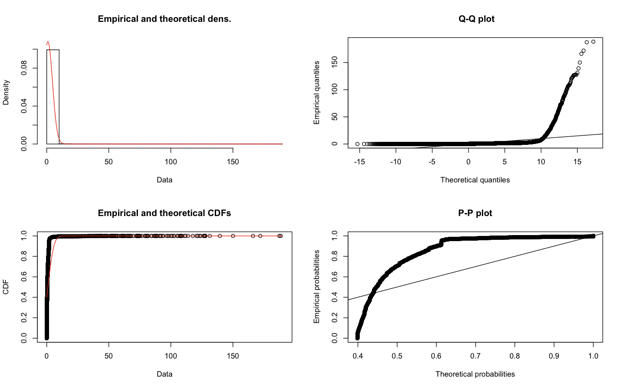
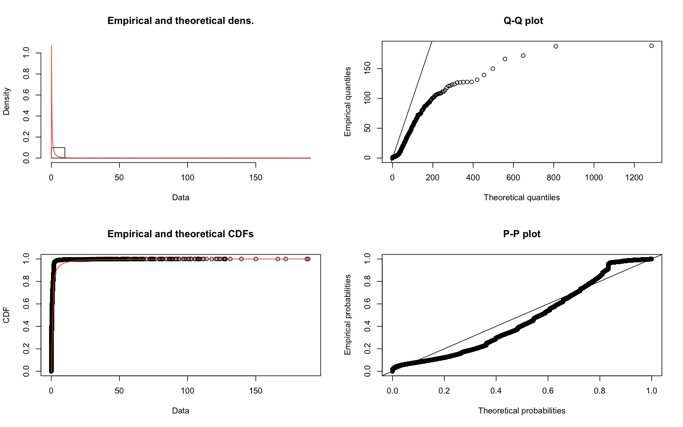
Exponential distribution Parameters:

Rate : 1.05

After removing outliers :



Other Distributions on given dataset :

Normal distribution Parameters: Log - Normal distribution Parameters:

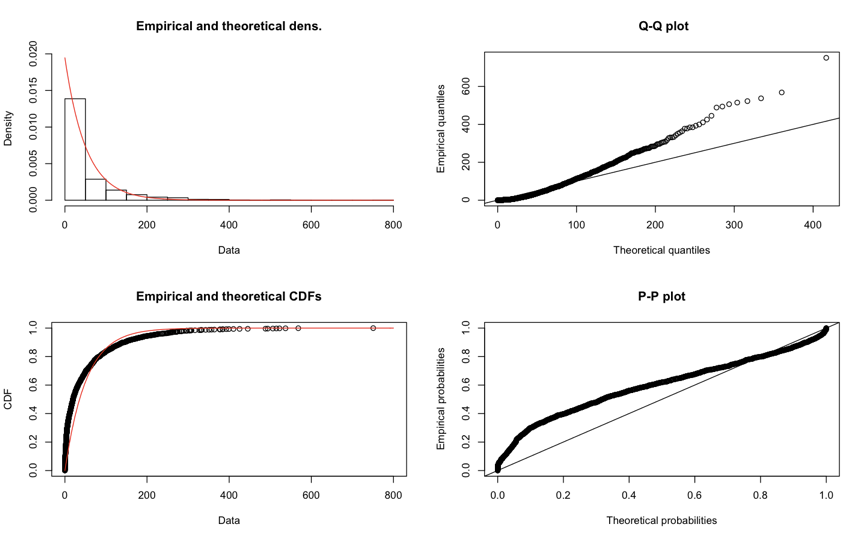
mean : 0.94 meanlog : -1.12

sd : 3.69 sdlog : 1.88

**DAY3**

* **Inter-Arrival Distribution between 2 consecutive connections**

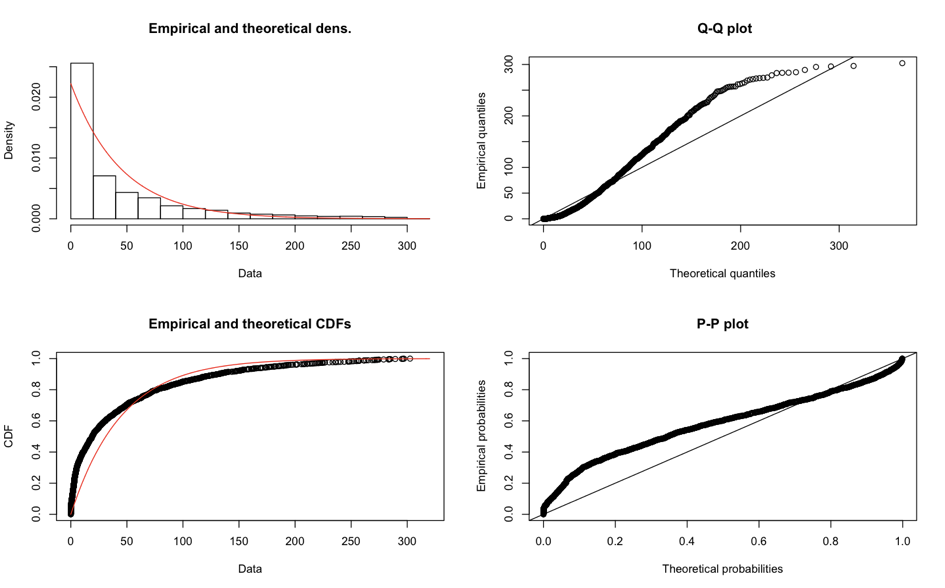
With outliers



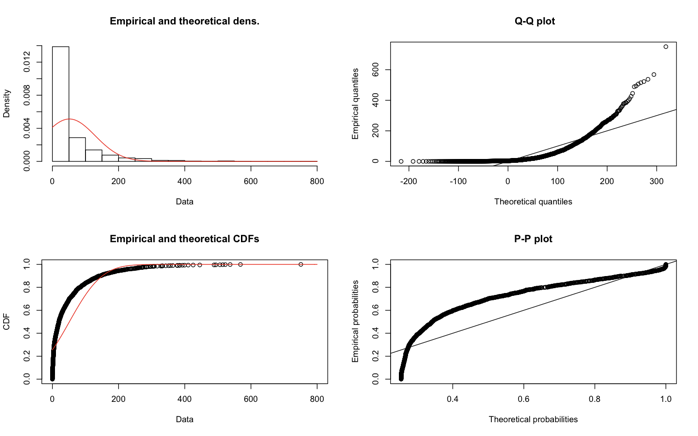
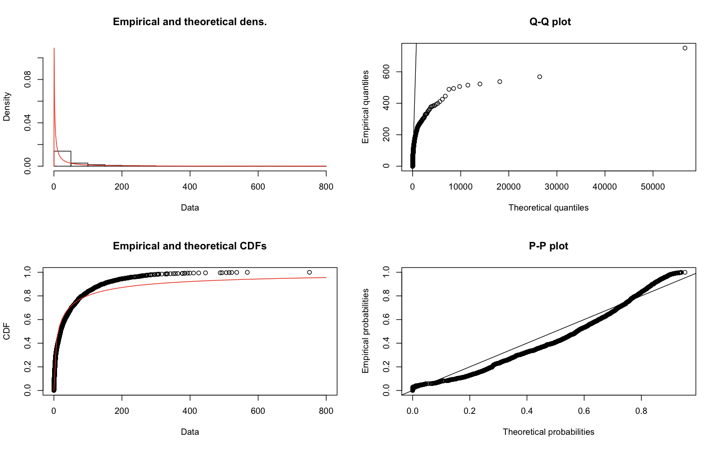
Exponential distribution Parameters:

Rate : 0.019

After removing outliers :



Other Distributions on given dataset :

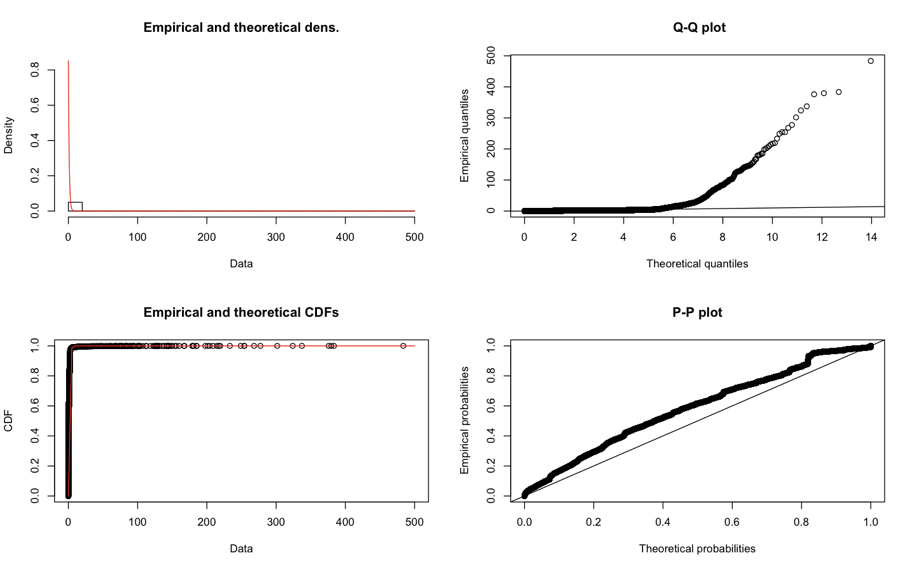
Normal distribution Parameters: Log - Normal distribution Parameters:

mean : 51.37 meanlog : 2.51

sd : 77.83 sdlog : 2.45

* **Inter-Arrival Distribution between 2 consecutive incoming packets to server**

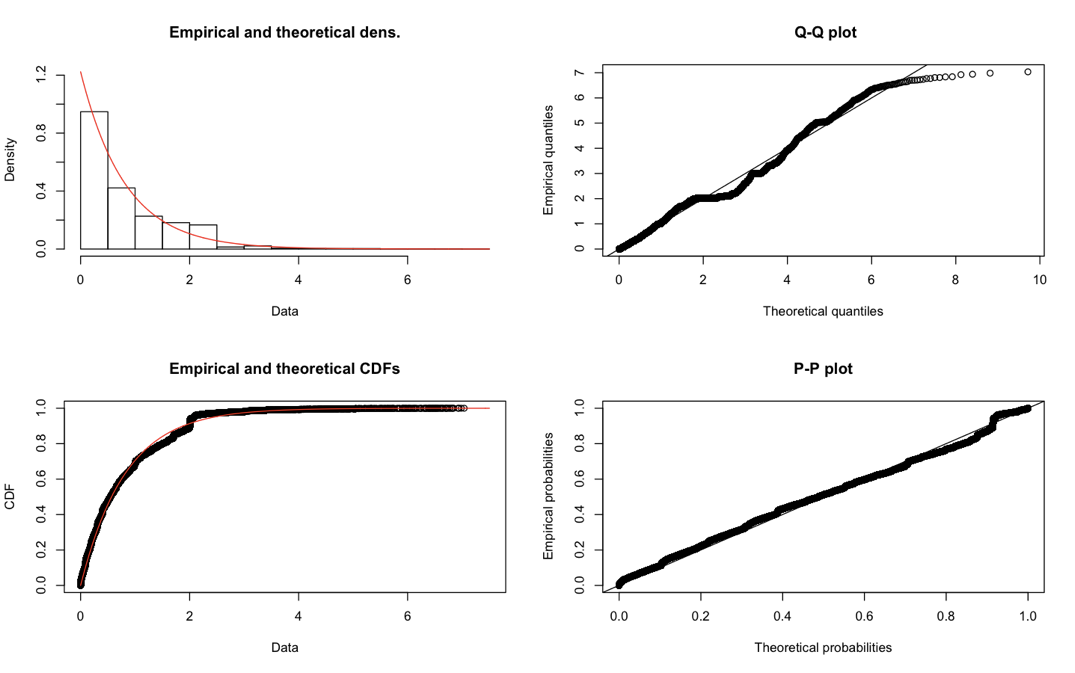
With outliers



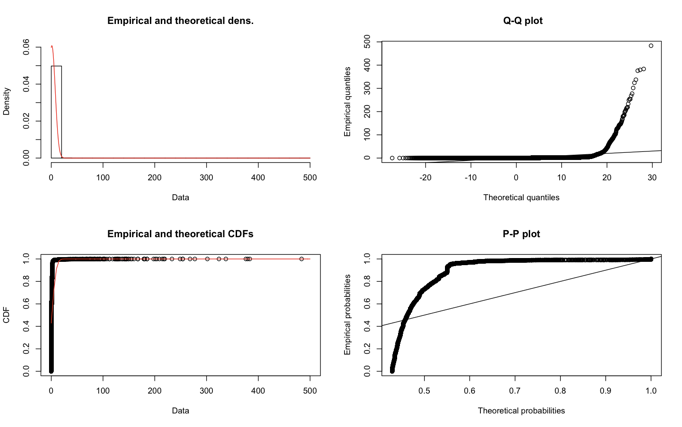
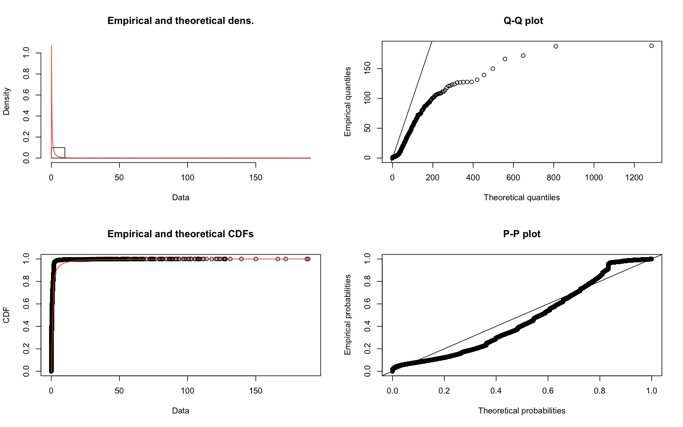
Exponential distribution Parameters:

Rate : 0.85

After removing outliers :



Other Distributions on given dataset :

Normal distribution Parameters: Log - Normal distribution Parameters:

mean : 1.17 meanlog : -0.85

sd : 6.56 sdlog : 1.61

From the above graphs, it may be seen that the Exponential distribution fits the data best, especially when compared to Normal and Log-Normal distributions. Also, removal of outliers leads to a better fitting of the exponential distribution, as evident from the various plots.

11.

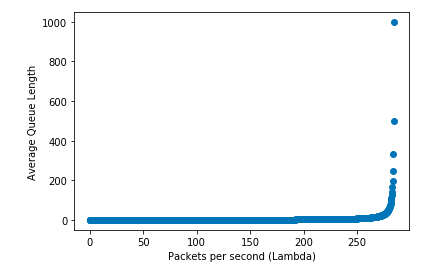
**System behavior**

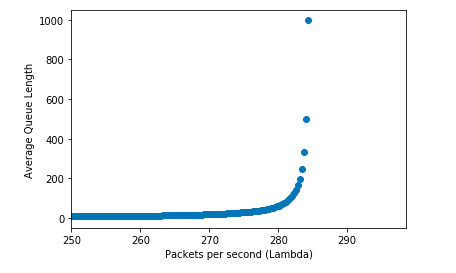
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Day** | **Lambda** | **Mean Packet Length(Bytes)** | **Mu** | **Utilization Factor, Rho** | **Average Queue Length** | **Average Waiting time(microseconds)** |
| 1 | 0.88 | 56.22 | 284.59 | 0.0031 | 0.0031 | 10.89 |
| 2 | 1.05 | 56.26 | 284.39 | 0.0036 | 0.0037 | 13.03 |
| 3 | 0.85 | 55.29 | 284.24 | 0.0029 | 0.0029 | 10.55 |

Link speed has been taken as 128Kbps.

For all the datasets, the packet length is close to 56.25, and thus Mu close to 284. The following graphs have been plot with these values, since all the plots for the datasets are almost identical.

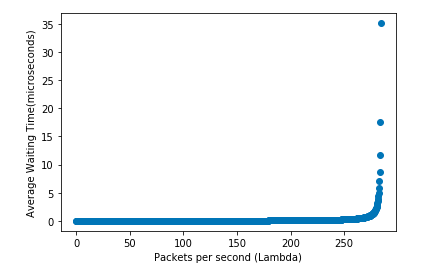
**Average Queue Length**

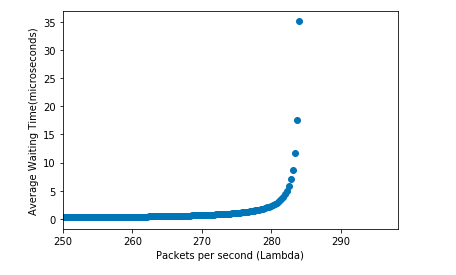




Zoomed in view

**Average Waiting Time**

****

****

Zoomed in view

It may be noted that the values of the utilization factor, average waiting time and queue length are very small, as the value of mu is much bigger than the value of lambda. The value of lambda is very less since the packet trace only contains FTP data.

From the graphs, it may be noted that the average waiting time and average queue length stay very small, and almost constant when lambda is not close to mu. These values increase rapidly when lambda nears mu, and asymptotically go to infinity at lambda = mu. Here, lambda is a measure of the incoming traffic, therefore, one can say that, till a certain limit, the network is capable of handling traffic efficiently and smoothly, however, at a certain point, the congestion rises rapidly. The increase isn’t gradual, as one might expect.