Queuing System

Queue with one server

In this section we run the queue for 10000 time, and we sum up the busy server time to find the unitilazation.

The statistics seem to come out a little weird if the log interval is tooshort, apparently because the log entries are not close enough to independent. So the log interval should be long enough for several arrival and departure events happen.

for arrival rate = 0.5 and service rate = 1/1.5 we have:

```
n_samples = 1;
max_time = 1000;
NInSystem = [];
```

```
%% Run the queue simulationclc, clear all;
ArrivalRate = 0.5;
ServiceRate  = 1/1.5;
last_arrival_time = 0;
id = 0;
q1 = ServiceQueue(LogInterval=2);
q2 = ServiceQueue(LogInterval=2);
Queues = {q1, q2};
n_samples = 1000;
max_time = 1000;
busy_time= 0;
expected_T_1 = 0;
expected_n_1 = 0;

expected_T_2 = 0;
expected_n_2 = 0;
for sample_num = 1:n_samples
    n_in_queue_1 = [];
    n_in_queue_2 = [];
    q1 = ServiceQueue(LogInterval=10);
    q2 = ServiceQueue(LogInterval=10);
    q1.dest_q = q2;
    while q1.Time < max_time
      last_arrival_time = generator(ArrivalRate, last_arrival_time, id, q1);
      q1.handle_next_event();
      q2.handle_next_event();
      id = id + 1;
    end
    id = 0;
    last_arrival_time  = 0;
    n_in_queue_1 = [n_in_queue_1, q1.Log.NWaiting + q1.Log.NInService];
    n_in_queue_2 = [n_in_queue_2, q2.Log.NWaiting + q2.Log.NInService];
    busy_time = busy_time + q2.busy_time + q1.busy_time;

    r = size(n_in_queue_1);
    expected_n_1 = sum(n_in_queue_1)/r(1) + expected_n_1;
    r = size(q1.time_in_system);
    expected_T_1 = sum(q1.time_in_system)/r(2) + expected_T_1;

    r = size(n_in_queue_2);
    expected_n_2 = sum(n_in_queue_2)/r(1) + expected_n_2;
    r = size(q1.time_in_system);
    expected_T_2 = sum(q2.time_in_system)/r(2) + expected_T_2;
end

expected_T_1 = expected_T_1/n_samples
```

```
expected_T_1 = 5.7914
```

```
expected_n_1 = expected_n_1/n_samples
```

```
expected_n_1 = 2.9659
```

```
landa_1 = expected_n_1/expected_T_1
```

```
landa_1 = 0.5121
```

```


expected_T_2 = expected_T_2/n_samples
```

```
expected_T_2 = 8.0288
```

```
expected_n_2 = expected_n_2/n_samples
```

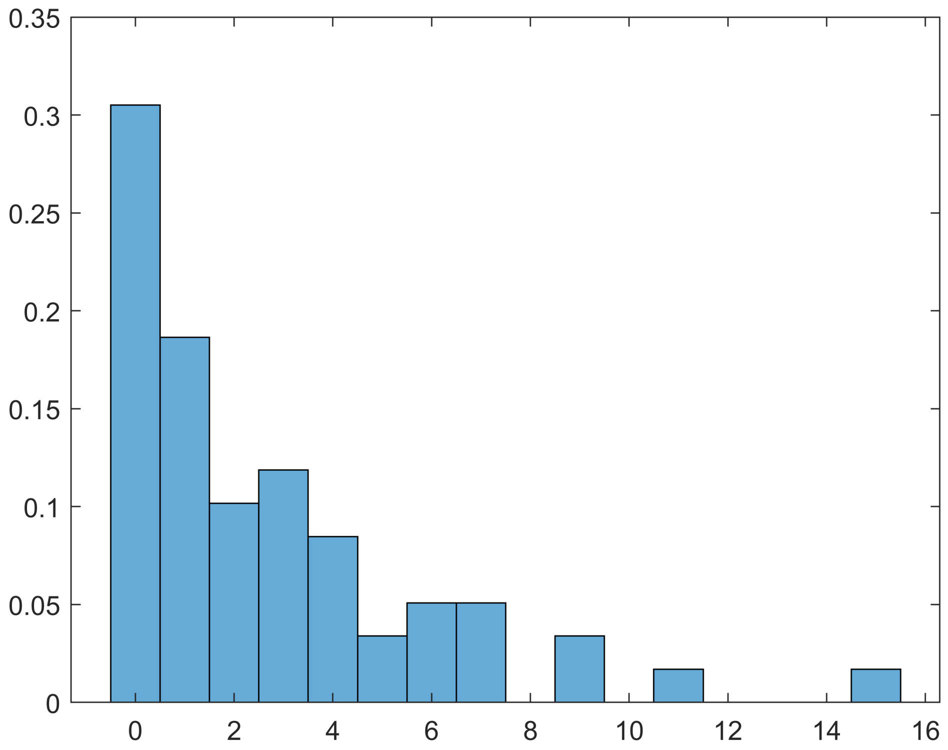
```
expected_n_2 = 3.5486
```

```
landa_2 = expected_n_2/expected_T_2
```

```
landa_2 = 0.4420
```

```
rho = busy_time/(n_samples * max_time*2);

h = histogram(n_in_queue_1, Normalization="probability", BinMethod="integers")
```

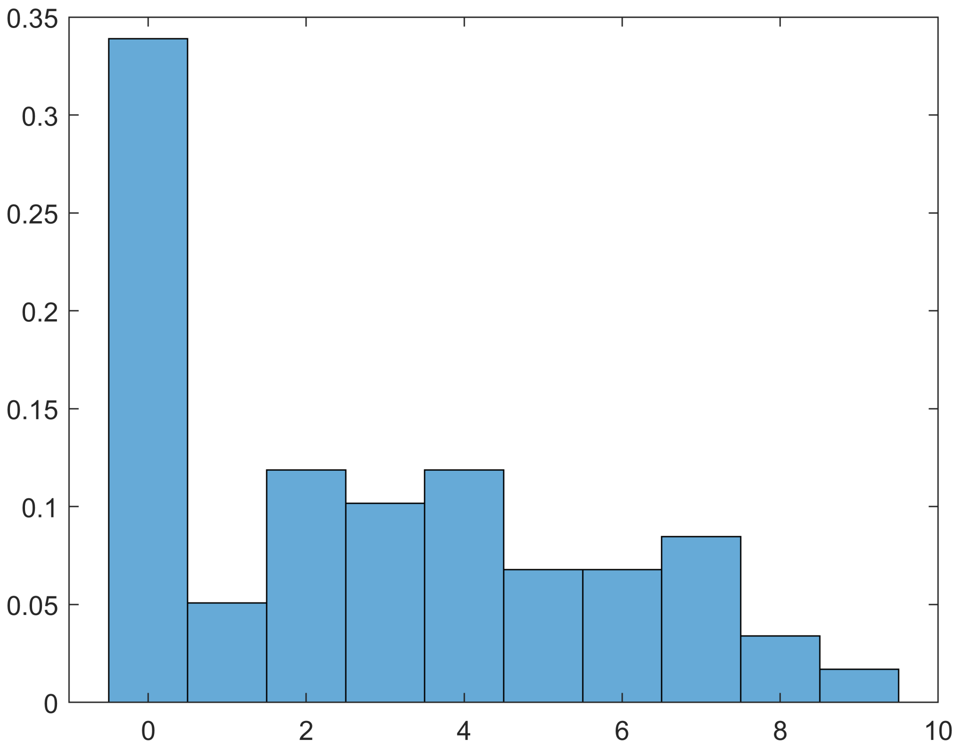


```
h = 
  Histogram with properties:

             Data: [59×1 int64]
           Values: [0.3051 0.1864 0.1017 0.1186 0.0847 0.0339 0.0508 0.0508 0 0.0339 0 0.0169 0 0 0 0.0169]
          NumBins: 16
         BinEdges: [-0.5000 0.5000 1.5000 2.5000 3.5000 4.5000 5.5000 6.5000 7.5000 8.5000 9.5000 10.5000 11.5000 12
         BinWidth: 1
        BinLimits: [-0.5000 15.5000]
    Normalization: 'probability'
        FaceColor: 'auto'
        EdgeColor: [0 0 0]

  Show all properties
```

```
h = histogram(n_in_queue_2, Normalization="probability", BinMethod="integers")
```



```
h = 
  Histogram with properties:

             Data: [59×1 int64]
           Values: [0.3390 0.0508 0.1186 0.1017 0.1186 0.0678 0.0678 0.0847 0.0339 0.0169]
          NumBins: 10
         BinEdges: [-0.5000 0.5000 1.5000 2.5000 3.5000 4.5000 5.5000 6.5000 7.5000 8.5000 9.5000]
         BinWidth: 1
        BinLimits: [-0.5000 9.5000]
    Normalization: 'probability'
        FaceColor: 'auto'
        EdgeColor: [0 0 0]

  Show all properties
```