BL40A2010 Introduction to IoT-Based Systems

Assignment 5, 15.02.2023

Author: Kush Patel

(1) Create two arrays with 10000 samples generated as Binomial random variables via Numpy similar to what we have done with Poisson random variables in the tutorial considering the following paramenters of the Binomial function:

(a) $n=100$ and $p=0.3$;

(b) $n=10000$, and $p=0.6$

Make the histogram (empirical) plots related to each vector and compare them with the analytical results with the probability distribution funtion given by:

$$\mathrm{Prob}\left[X = k \right] = \binom{n}{k}p^k(1-p)^{n-k},$$ where k is the number of *successes* in n trials.

Remember to use this scipy.stats.binom in a similar way we did in the tutorial with Poisson.

In [1]:
```
import numpy as np
import matplotlib.pyplot as plt
from matplotlib import style
from scipy.stats import poisson
style.use('bmh')
#Not to show warning messages (to keep the notebook clean)
import warnings
warnings.filterwarnings('ignore')
```

In [5]:
```
mean=0.3
samples = np.random.poisson(mean, 100)
print(samples)
```

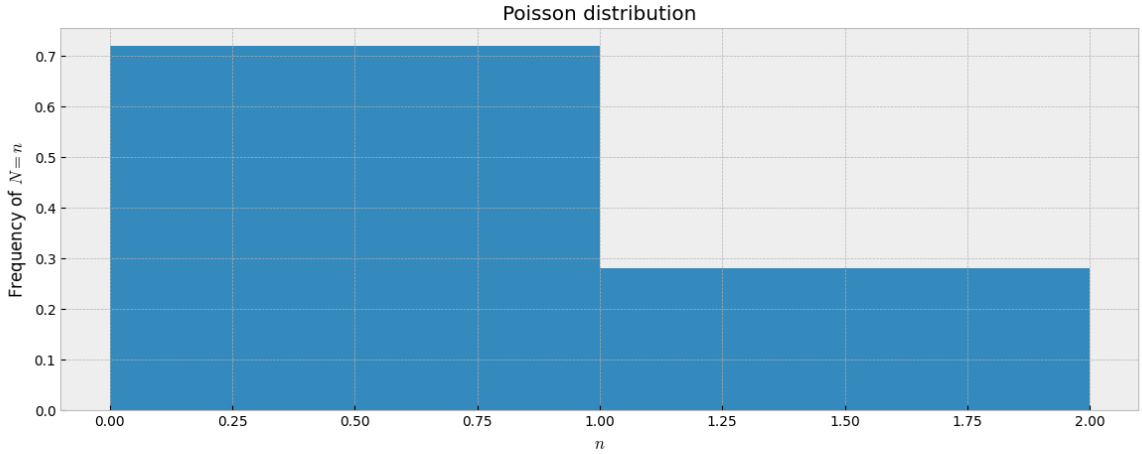
```
[1 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 1
 0 0 1 0 0 1 1 0 0 1 0 0 0 0 0 0 0 0 1 1 1 0 1 1 2 0 0 0 0 1 1 0 0 0 1 1 1
 0 1 0 0 1 0 0 0 0 0 1 0 0 1 0 1 0 0 1 2 0 0 1 0 0 0]
```

In [6]:
```
mean=0.6
samples = np.random.poisson(mean, 10000)
print(samples)
```

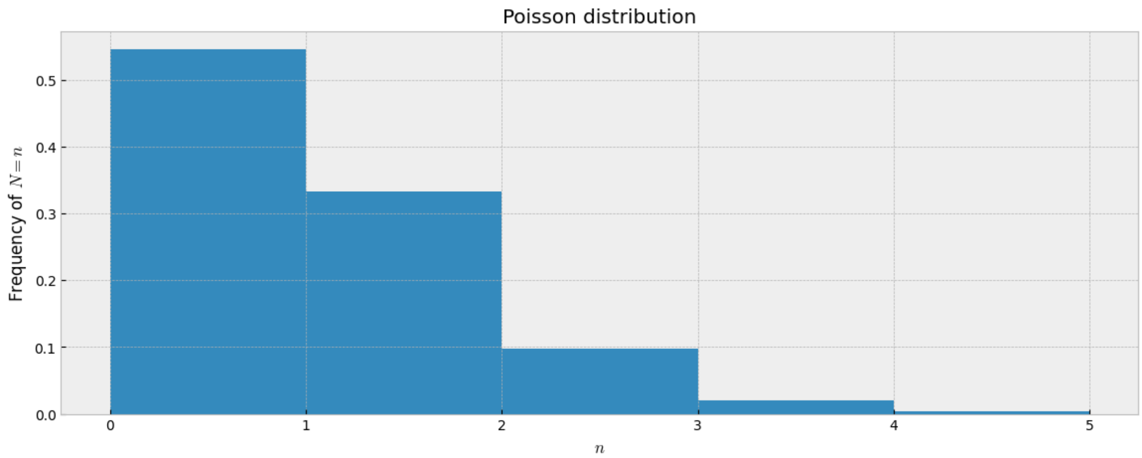
```
[2 0 1 ... 0 0 0]
```

In [8]:
```
mean=0.3
samples_mean2 = np.random.poisson(mean, 100)
n_bins2 = np.max(samples_mean2) - np.min(samples_mean2)
##
plt.figure(figsize=(14,5))
```

```
plt.hist(samples_mean2,n_bins2,density=True)
plt.xlabel('$n$')
plt.ylabel('Frequency of $N = n$')
plt.title('Poisson distribution')
plt.show()
```



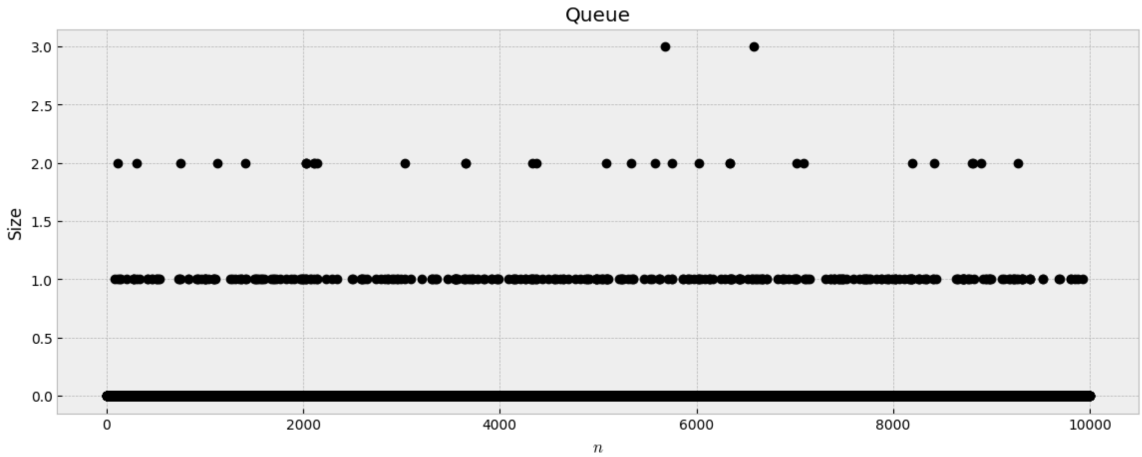
In [9]:
```
mean=0.6
samples_mean2 = np.random.poisson(mean, 10000)
n_bins2 = np.max(samples_mean2) - np.min(samples_mean2)
##
plt.figure(figsize=(14,5))
plt.hist(samples_mean2,n_bins2,density=True)
plt.xlabel('$n$')
plt.ylabel('Frequency of $N = n$')
plt.title('Poisson distribution')
plt.show()
```



(2) Read Section 2.3 and Chapter 3 from Network Science and generate three different kinds of Erdos-Renyi graphs with $N=15$ (fifteen nodes) using NetworkX. The networks shall be (a) with probability $p=0.2$, (b) with probability $p=0.5$ and (c) with probability $p=0.8$. What are the differences you see if these graphs represent communication networks.

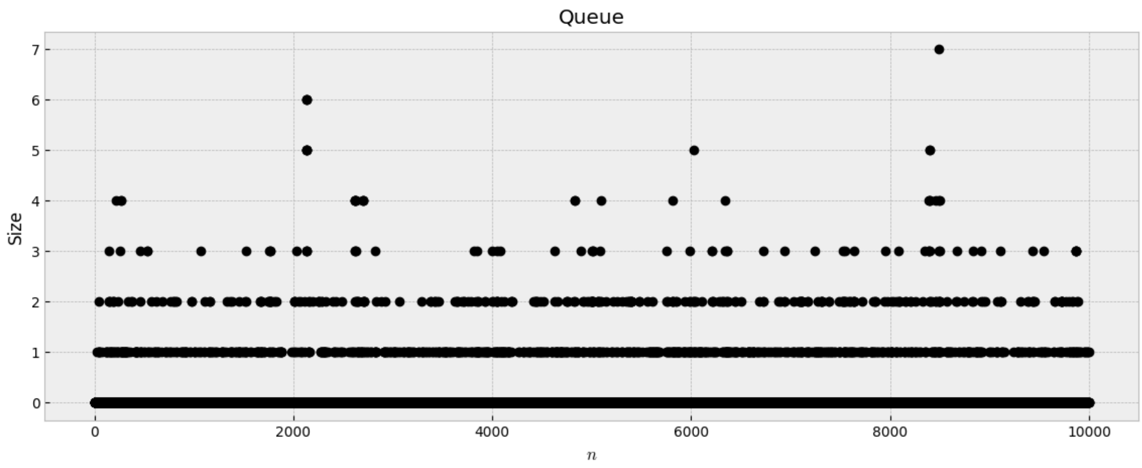
In [23]:
```
size=10000
mean=0.2
arrival = np.random.poisson(mean, size)
mean=2
service = np.random.poisson(mean, size)
queue = np.zeros(size+1)
```

```
for i in range(size):
    if ((arrival[i] - service[i]) + queue[i]) > 0:
        queue[i+1] = max(0,(arrival[i] - service[i]) + queue[i])

plt.figure(figsize=(14,5))
plt.plot( queue, 'ok')
plt.xlabel('$n$')
plt.ylabel('Size')
plt.title('Queue')
plt.grid(True)
plt.show()

print('Worst delay:', np.max(queue))
```



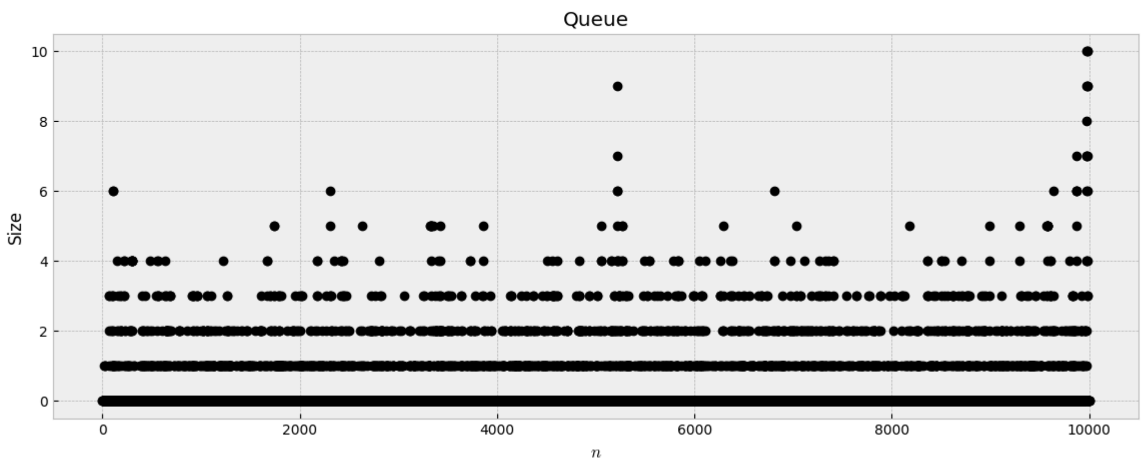

```
Worst delay: 3.0
```

In [24]:
```
size=10000
mean=0.5
arrival = np.random.poisson(mean, size)
mean=2
service = np.random.poisson(mean, size)
queue = np.zeros(size+1)

for i in range(size):
    if ((arrival[i] - service[i]) + queue[i]) > 0:
        queue[i+1] = max(0,(arrival[i] - service[i]) + queue[i])

plt.figure(figsize=(14,5))
plt.plot( queue, 'ok')
plt.xlabel('$n$')
plt.ylabel('Size')
plt.title('Queue')
#plt.ylim([0, 1])
#plt.xlim([0, 10])
plt.grid(True)
plt.show()

print('Worst delay:', np.max(queue))
```



Worst delay: 7.0

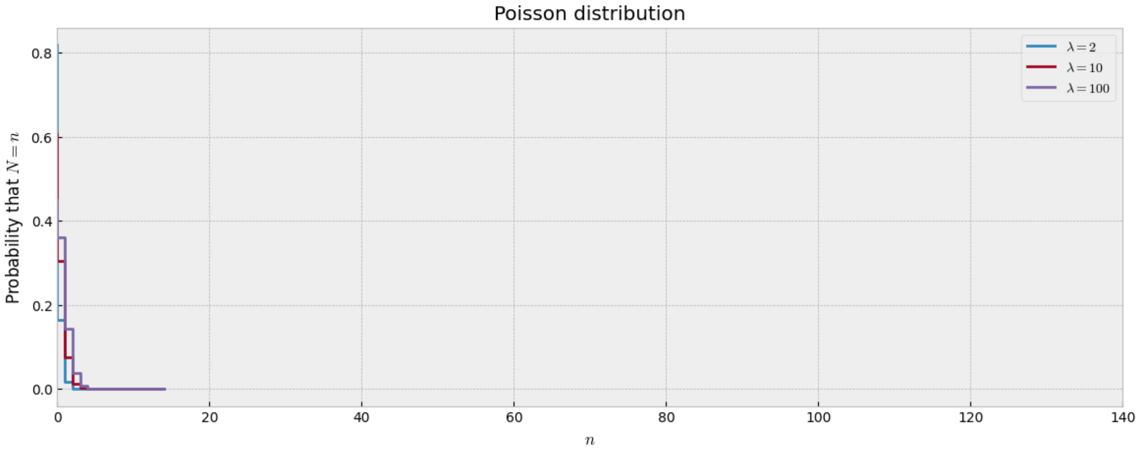
In [25]:
```python
size=10000
mean=0.8
arrival = np.random.poisson(mean, size)
mean=2
service = np.random.poisson(mean, size)
queue = np.zeros(size+1)
for i in range(size):
    if ((arrival[i] - service[i]) + queue[i]) > 0:
        queue[i+1] = max(0,(arrival[i] - service[i]) + queue[i])
plt.figure(figsize=(14,5))
plt.plot( queue, 'ok')
plt.xlabel('$n$')
plt.ylabel('Size')
plt.title('Queue')
#plt.ylim([0, 1])
#plt.xlim([0, 10])
plt.grid(True)
plt.show()

print('Worst delay:', np.max(queue))
```



Worst delay: 10.0

In [26]:
```python
n = np.arange(15)
plt.figure(figsize=(14,5))
plt.step(n, poisson.pmf(n, 0.2), where='pre', label='$\lambda=2$')
plt.step(n, poisson.pmf(n, 0.5), where='pre', label='$\lambda=10$')
plt.step(n, poisson.pmf(n, 0.8), where='pre', label='$\lambda=100$')
plt.xlabel('$n$')
plt.ylabel('Probability that $N = n$')
```

```
plt.title('Poisson distribution')
plt.xlim([0, 140])
plt.grid(True) #grid
plt.legend()
plt.show()
```



In [ ]: