Queuing Theory Exercise Series 1

Alexandros Kyriakakis (03112163)

April 2020

1 Poisson Distribution

1.1 A

At the figure 1 we notice that while the λ grows bigger so does the mean and the variance of the distribution. So from our intuition we notice "wider" "bell curve" which may reach X-Axis for $\lambda \to \inf$ and dirac while $\lambda \to 0$.

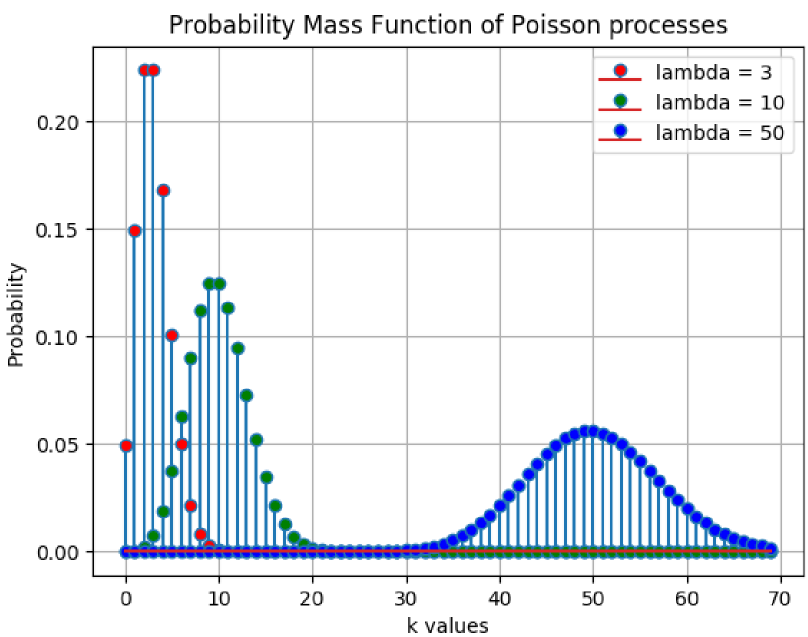


Figure 1: Poisson PMF

1.2 B

We notice that mean and variance of the Poisson Distribution with $\lambda = 30$, are equal to λ.

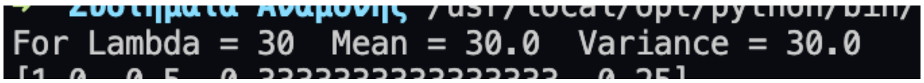


Figure 2: Mean and Variance

1.3 Γ

In figure 3 we notice that the distribution of the convolution of 2 Poisson distributions, is also a Poisson distribution, a fact we knew from theory. Also we know that the result Poisson distribution has $\lambda_{conv} = 10 + 50$. This happens if and only if the two distributions are Poisson distributions.

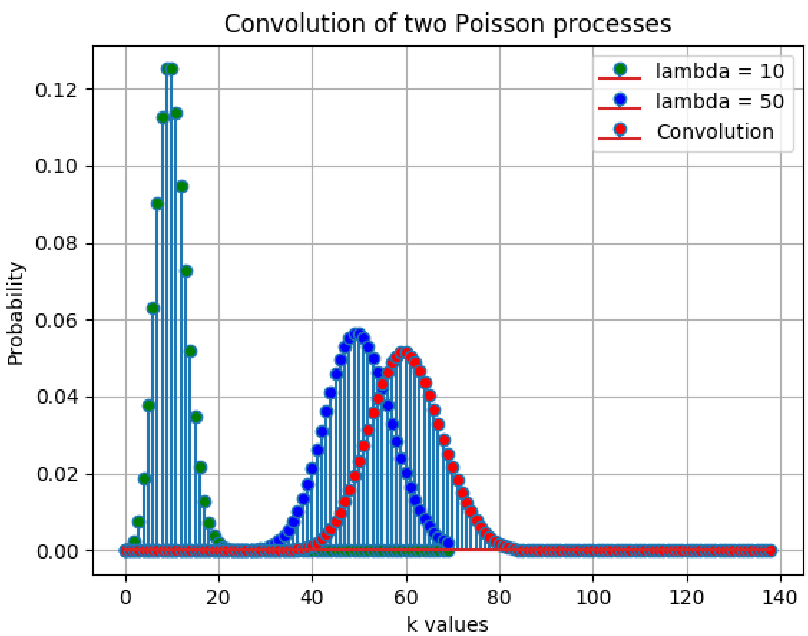


Figure 3: Poisson Convolution

1.4 Δ

In this section we are going to prove that the Poisson Distribution can be derived from the Binomial distribution. It turns out that the Poisson Distribution is just a special case of the Binomial Distribution where the number of trials is very big and the probability of success small enough. With some formalism, we have:

$$Poisson = P[X \leq k] = \frac{(\lambda t)^k}{k!} e^{-\lambda t} = \lim_{n \to \infty} \binom{n}{k} p^k (1-p)^k$$

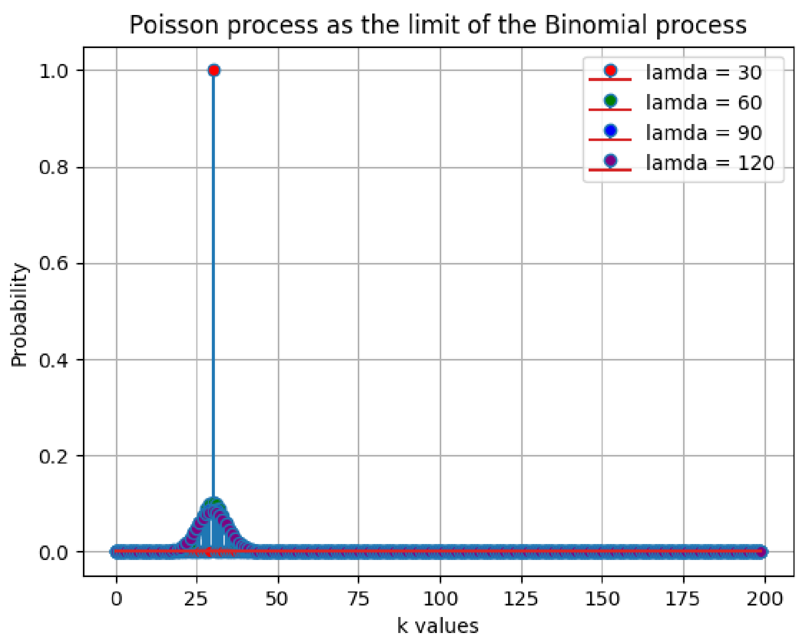


Figure 4: Poisson Convolution

2 Exponential Distribution

2.1 A

In figure 5 we plot the Exponential Probability Density Function (PDF) for $\frac{1}{\lambda} = \{0.5, 1, 3\}$

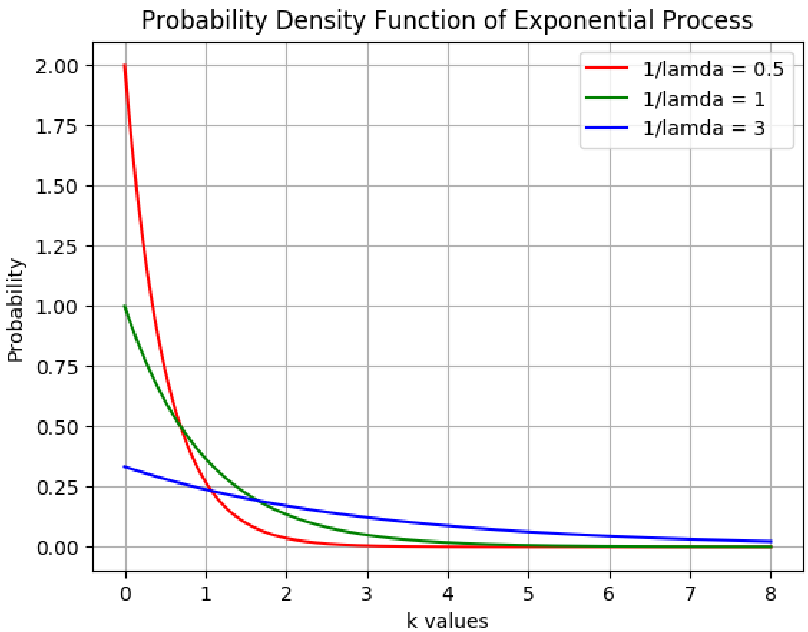


Figure 5: Exponential PDF

2.2 B

In figure 6 we plot the Exponential Cumulative Distribution Function (CDF) for $\frac{1}{\lambda} = \{0.5, 1, 3\}$

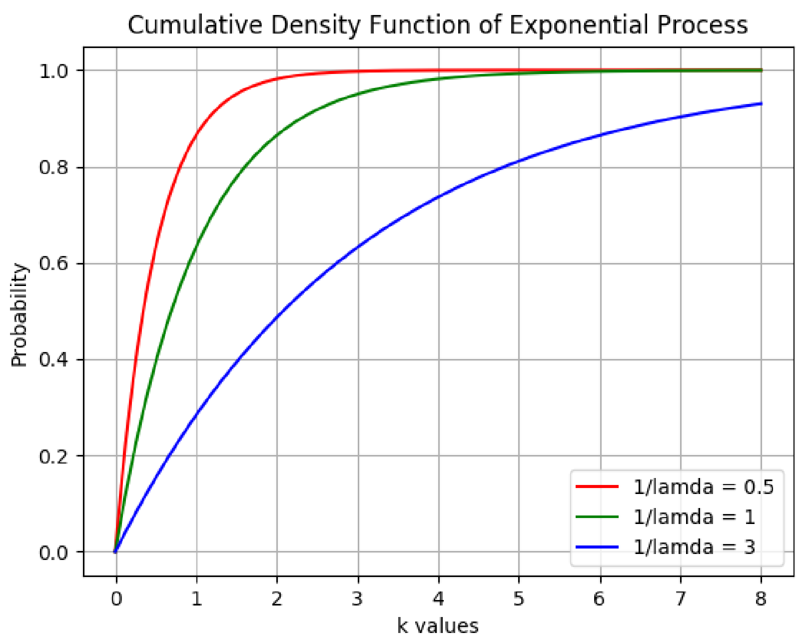


Figure 6: Exponential CDF

2.3 Γ

From the figure 7 we notice that $Pr[X > 30000] = Pr(X > 50000|X > 20000)$. This is a result of the memory loss property of exponential distribution. Again formaly we know:

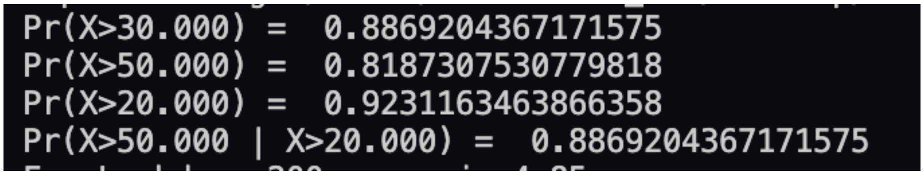
$$Pr(X > t+s|X > s) = \frac{Pr((X > t+s)\bigcap(X > s))}{Pr(X > s)} = \frac{Pr(X > t+s)}{Pr(X > s)} = Pr(X > t)$$

$$Pr(X > 50000|X > 20000) = Pr(X > 30000+20000|X > 20000) = Pr(X > 30000)$$

Where,

$$Pr(X > a) = 1 - Pr(X \leq a) = 1 - CDF(a)$$

$$\Rightarrow Pr(X > 30000) = 1 - CDF(30000)$$



```
Pr(X>30.000) =  0.8869204367171575
Pr(X>50.000) =  0.8187307530779818
Pr(X>20.000) =  0.9231163463866358
Pr(X>50.000 | X>20.000) =  0.8869204367171575
```

Figure 7: Memorylessness

3 Poisson Processes

3.1 A

The measuring time between two successively Poisson events follows Exponential Distribution, as we notice in figure 8.

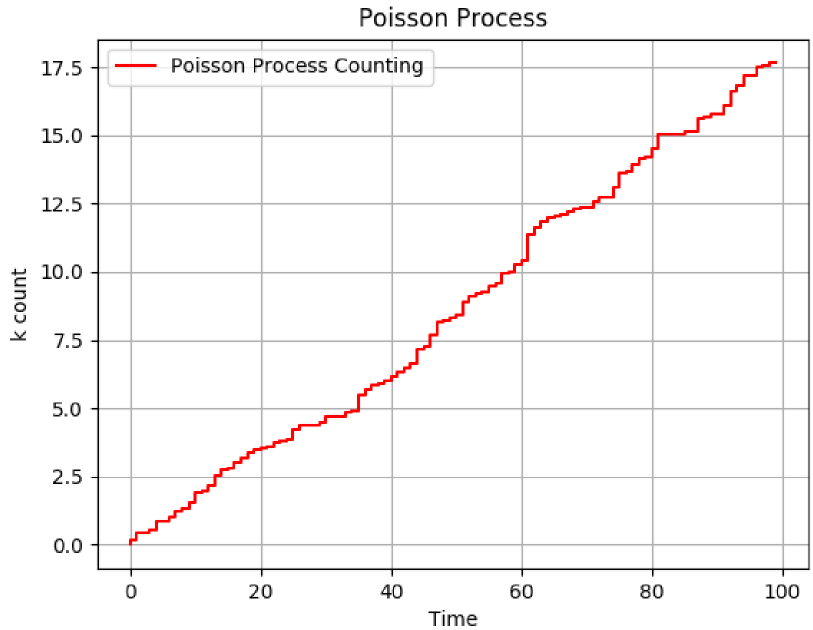


Figure 8: Exponential Stairs

3.2 B

On the other hand, in a window $dT = t_1 - t_2$ the number of events follows Poisson Distribution with mean λt.

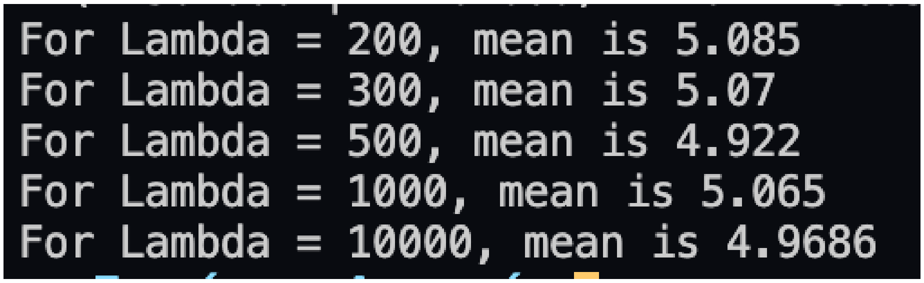


Figure 9: Mean

4 Code

```
import numpy as np
#import scipy as sp
from scipy.stats import binom
from scipy.stats import poisson
import matplotlib.pyplot as plt
countFig = 0
figureTitles = ["Probability Mass Function of Poisson processes","Convolution of two Poisson processes","Poisson process as the limit of the Binomial process","Probability Density Function of Exponential Process","Cumulative Density Function of Exponential Process","Poisson Process"]
labelTuple = [("k values","Probability"),("k values","Probability"),("k values","Probability"),("k values","Probability"),("k values","Probability"),("Time","k count")]
def show_plot():
    global countFig, labelTuple
    ax.legend()
    ax.grid()
    #plt.show() #FIXME Disabled
    plt.title(figureTitles[countFig])
    xlabel, ylabel = labelTuple[countFig]
    plt.xlabel(xlabel)
    plt.ylabel(ylabel)

    plt.savefig("/Users/alexanders_mac/Desktop/Profiteus/NTUA//Exercises/Lab1/figures/figure{}".format(countFig))
    countFig += 1
def rm_plot():
    plt.clf()
    plt.cla()
    plt.close()

#Katanomi Poisson

#Alpha
tau = np.arange(0,70,1)

fig, ax = plt.subplots()
```

```python

lamda = [3,10,50]
#print (enumerate(lamda))
for lamda_i,color_i,label_i in zip(lamda,["red","green","blue"],["lambda = 3","lambda = 10","lambda = 50"]):
    mu = lamda_i
    #mean, var, skew, kurt = poisson.stats(mu, moments='mvsk')
    markerline, stemlines, baseline = ax.stem(tau, poisson.pmf(tau, mu), label = label_i,use_line_collection=True) #'bo', ms=8, label='poisson pmf')
    #ax.vlines(tau, 0, poisson.pmf(tau, mu), colors=color_i)
    markerline.set_markerfacecolor(color_i)
show_plot()
rm_plot()
#print (mu)

#Beta
lamda = 30
mean, var, skew, kurt = poisson.stats(lamda, moments='mvsk')
print ("For Lambda =",lamda," Mean =", mean, " Variance =", var)

#Gamma
lamda = [10,50]
fig, ax = plt.subplots()
conv = np.convolve(poisson.pmf(tau, lamda[0]),poisson.pmf(tau, lamda[1]))
markerline, stemlines, baseline = ax.stem(tau,poisson.pmf(tau, lamda[0]), label = "lambda = 10",use_line_collection=True)
markerline.set_markerfacecolor("green")
markerline, stemlines, baseline = ax.stem(tau,poisson.pmf(tau, lamda[1]), label = "lambda = 50",use_line_collection=True)
markerline.set_markerfacecolor("blue")
markerline, stemlines, baseline = ax.stem(conv, label = "Convolution",use_line_collection=True)
markerline.set_markerfacecolor("red")
show_plot()
rm_plot()
#print (len(poisson.pmf(tau, lamda[0])),len(conv))
#ax.stem
#delta
fig, ax = plt.subplots()
lamda = [30,60,90,120]
#Binom
k = np.arange(0,200,1)
n = lamda
p = [30/x for x in n]
print (p)
for n_i,p_i,color_i in zip(n,p,["red","green","blue","purple"]):
    markerline, stemlines, baseline = ax.stem(k, binom.pmf(k, n_i, p_i), label=('lamda = '+ str(n_i)),use_line_collection=True)
    markerline.set_markerfacecolor(color_i)
show_plot()
rm_plot()

#Ekthetikh katanomh
from scipy.stats import expon
#Alpha
```

```
def add_plot(x,y,label,color):
    ax.plot(x, y, color = color, label=label)
inv_lamda = [0.5,1,3]
colors = ["red","green","blue"]
fig, ax = plt.subplots()
tau = np.arange(0,8,0.00001)
for lamda_i,color_i in zip(inv_lamda,colors):
    add_plot(tau, expon.pdf(tau,0,lamda_i),"1/lamda = "+str(lamda_i),color_i)
show_plot()
rm_plot()

#Beta
fig, ax = plt.subplots()
for lamda_i,color_i in zip(inv_lamda,colors):
    add_plot(tau, expon.cdf(tau,0,lamda_i),"1/lamda = "+str(lamda_i),color_i)
show_plot()
rm_plot()

#Gamma
print ("Pr(X>30.000) = ",1-expon.cdf(tau[30000],0,2.5))
print ("Pr(X>50.000) = ",1-expon.cdf(tau[50000],0,2.5))
print ("Pr(X>20.000) = ",1-expon.cdf(tau[20000],0,2.5))
print ("Pr(X>50.000 | X>20.000) = ",(1-expon.cdf(tau[50000],0,2.5))/(1-expon.cdf(tau[20000],0,2.5)))

#Poisson Process
#Alpha
lamda = 5

grid = np.random.exponential(1/lamda,100)
#print ("This is MEAN",np.mean(grid))
grid = [sum(grid[0:i]) for i in range(100)]

fig, ax = plt.subplots()
ax.step(range(100), grid,label = "Poisson Process Counting", color = "red")
show_plot()
rm_plot()

for i in [2,3,5,10,100]:
    grid = np.random.poisson(lamda,i*100)
    print ("For Lambda = {}, mean is {}".format(i*100,np.mean(grid)))

#print (grid)
```