# BM20BTECH11001-Lab16

December 2, 2021

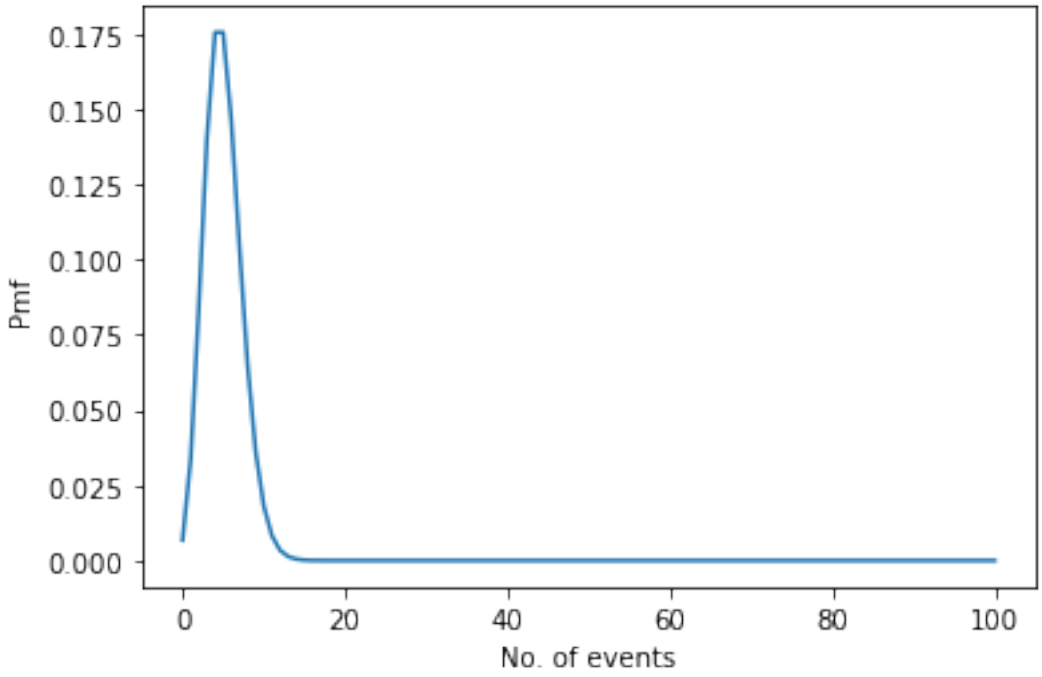
[1]:
```
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
```

## 0.1 Question-1

### 0.1.1 N1

[2]:
```
k = np.linspace(0, 100, 101)
t = 1
lam = 5
pmf_n_1 = []
for i in k:
    pmf_n_1.append((((lam*t)**i)*(np.e**(-lam*t)))/(np.math.factorial(i)))
```

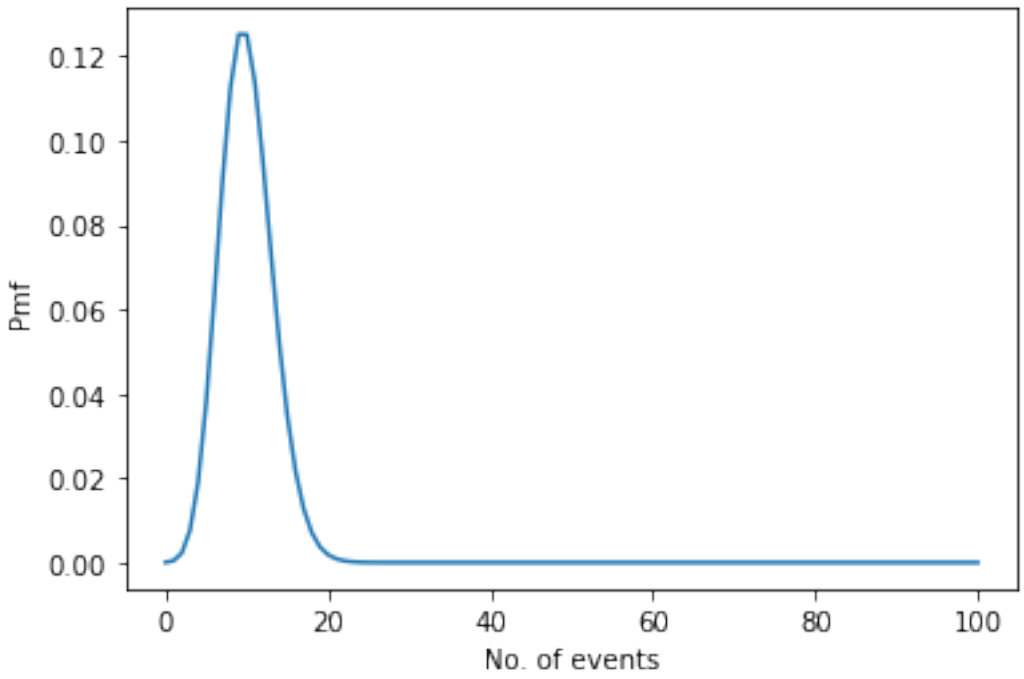
[3]:
```
fig, ax = plt.subplots()
ax.plot(k, pmf_n_1);
ax.set_xlabel('No. of events')
ax.set_ylabel('Pmf');
```




### 0.1.2 N2

[4]:
```
k = np.linspace(0, 100, 101)
t = 2
lam = 5
pmf_n_2 = []
for i in k:
    pmf_n_2.append((((lam*t)**i)*(np.e**(-lam*t)))/(np.math.factorial(i)))
```

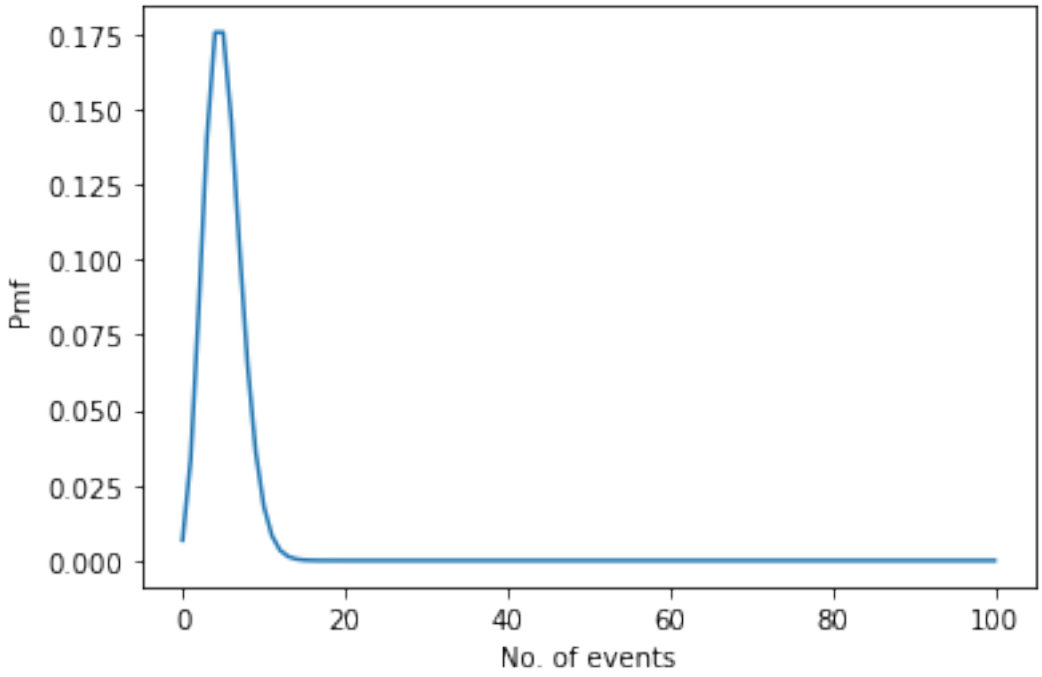
[5]:
```
fig, ax = plt.subplots()
ax.plot(k, pmf_n_2);
ax.set_xlabel('No. of events')
ax.set_ylabel('Pmf');
```




### 0.1.3 N2-N1 will be same as N1 as poisson distribution is memoryless

[6]:
```
k = np.linspace(0, 100, 101)
t = 1
lam = 5
pmf_n_2_1 = []
for i in k:
    pmf_n_2_1.append((((lam*t)**i)*(np.e**(-lam*t)))/(np.math.factorial(i)))
```

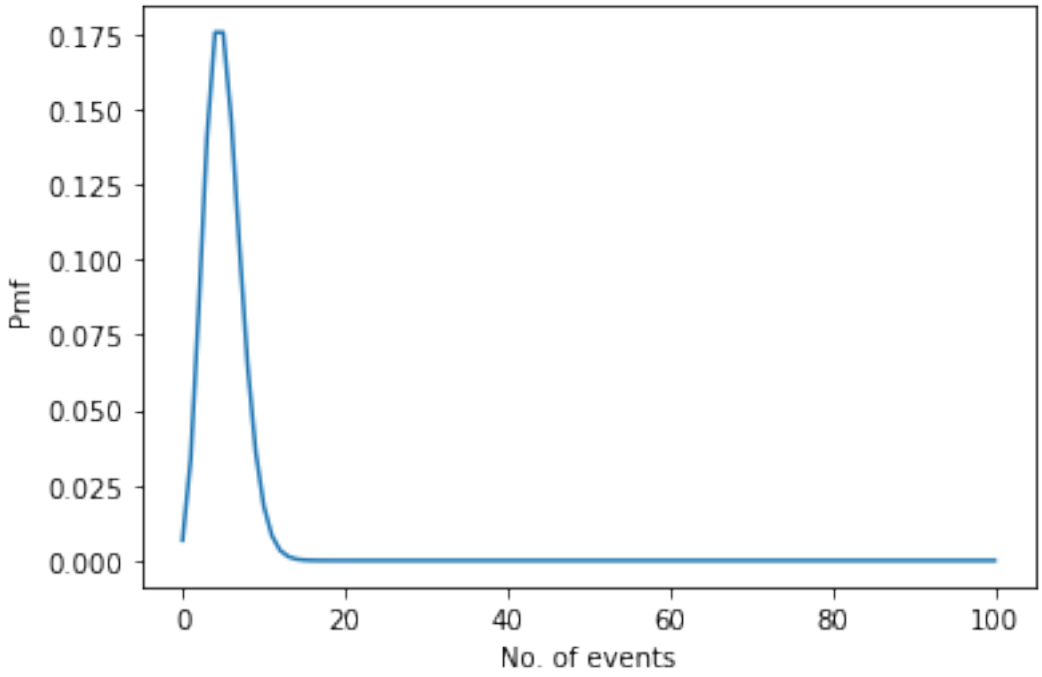
[7]:
```
fig, ax = plt.subplots()
ax.plot(k, pmf_n_2_1);
ax.set_xlabel('No. of events')
ax.set_ylabel('Pmf');
```




### 0.1.4 N3-N2 will be same as poisson distribution is memoryless

[8]:
```
k = np.linspace(0, 100, 101)
t = 1
lam = 5
pmf_n_3_2 = []
for i in k:
    pmf_n_3_2.append((((lam*t)**i)*(np.e**(-lam*t)))/(np.math.factorial(i)))
```

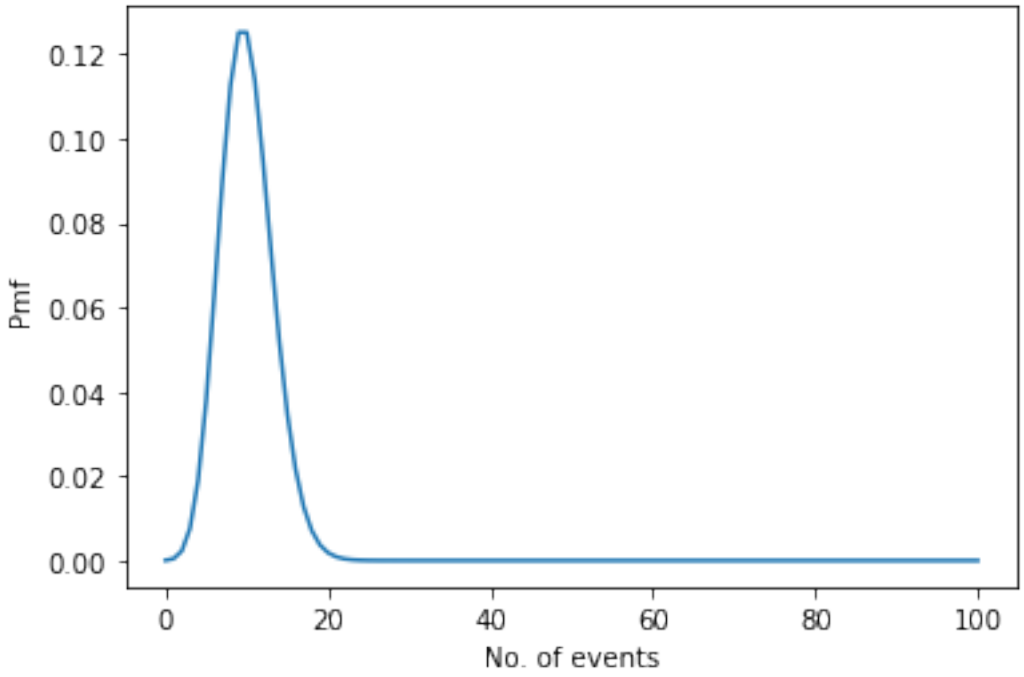
[9]:
```
fig, ax = plt.subplots()
ax.plot(k, pmf_n_3_2);
ax.set_xlabel('No. of events')
ax.set_ylabel('Pmf');
```




### 0.1.5 N3-N1 will be same as N2 as poisson distribution is memoryless

[10]:
```
k = np.linspace(0, 100, 101)
t = 2
lam = 5
pmf_n_3_1 = []
for i in k:
    pmf_n_3_1.append((((lam*t)**i)*(np.e**(-lam*t)))/(np.math.factorial(i)))
```

[11]:
```
fig, ax = plt.subplots()
ax.plot(k, pmf_n_3_1)
ax.set_xlabel('No. of events')
ax.set_ylabel('Pmf');
```




### 0.2 Question-2

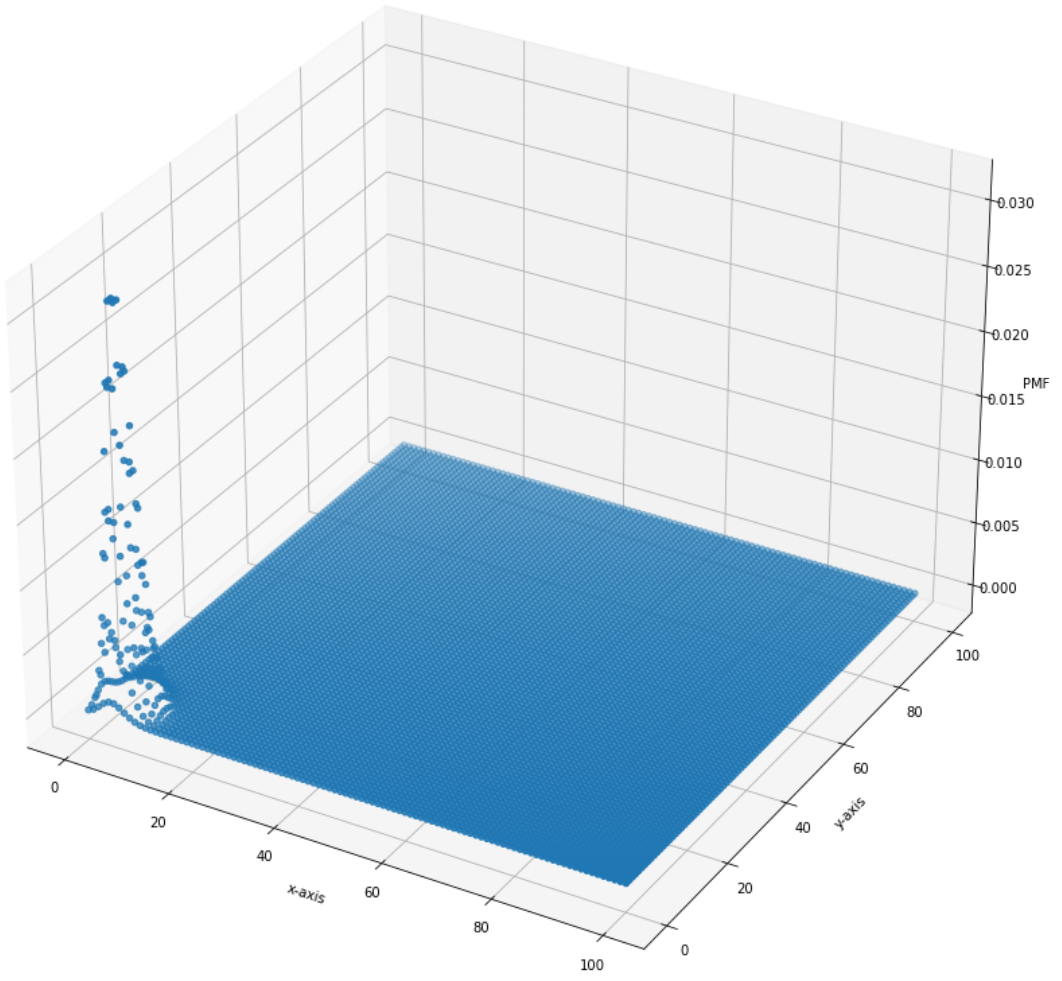
- As X=N1 and Y=N2-N1 are IID, fXY(x,y) = fX(x)*fY(y)

[12]:
```
x = np.linspace(0, 100, 101)
y = np.linspace(0, 100, 101)
z = []
x_1 = []
y_1 = []
for a in x:
    for b in y:
        x_1.append(a)
        y_1.append(b)
        t = (((lam*1)**a)*(np.e**(-lam*1)))/(np.math.factorial(a))
        s = (((lam*1)**b)*(np.e**(-lam*1)))/(np.math.factorial(b))
        z.append(t*s)
```

[13]:
```
fig = plt.figure(figsize=(15, 15))
ax = fig.add_subplot(111, projection='3d')

img = ax.scatter3D(x_1, y_1, z)

ax.set_xlabel('x-axis')
ax.set_ylabel('y-axis')
ax.set_zlabel('PMF');
```


0.030
0.025
0.020
PMF
0.015
0.010
0.005
0.000
0
20
40
60
80
100
x-axis
0
20
40
60
80
100
y-axis