DSL1_C5_S4_Practice

In [24]:

```
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
import statistics as st
```

Task 1 . a

In [23]:

```
import numpy as np
p =0.15   # gitting filed correctly
q = 1-p    # not getting not filied correctly
n=25
k = np.arange(0,26)
from scipy.stats import binom
binomial = binom.pmf(k,n,p)
binomial
```

Out[23]:

```
array([1.71978099e-02, 7.58726905e-02, 1.60671580e-01, 2.17379196e-01,
       2.10985691e-01, 1.56377630e-01, 9.19868409e-02, 4.40609238e-02,
       1.74947786e-02, 5.83159285e-03, 1.64656739e-03, 3.96232795e-04,
       8.15773402e-05, 1.43960012e-05, 2.17754640e-06, 2.81800123e-07,
       3.10808959e-08, 2.90375152e-09, 2.27745217e-10, 1.48069646e-11,
       7.83898125e-13, 3.29368960e-14, 1.05679880e-15, 2.43252921e-17,
       3.57724884e-19, 2.52511683e-21])
```

In [3]:

```
pdist =pd.DataFrame({ 'no of order':list(k), 'probability ': list(binomial)})
pdist
```

Out[3]:

	no of order	probability
0	0	1.719781e-02
1	1	7.587269e-02
2	2	1.606716e-01
3	3	2.173792e-01
4	4	2.109857e-01
5	5	1.563776e-01
6	6	9.198684e-02
7	7	4.406092e-02
8	8	1.749478e-02
9	9	5.831593e-03
10	10	1.646567e-03
11	11	3.962328e-04
12	12	8.157734e-05
13	13	1.439600e-05
14	14	2.177546e-06
15	15	2.818001e-07
16	16	3.108090e-08
17	17	2.903752e-09
18	18	2.277452e-10
19	19	1.480696e-11
20	20	7.838981e-13
21	21	3.293690e-14
22	22	1.056799e-15
23	23	2.432529e-17
24	24	3.577249e-19
25	25	2.525117e-21

In [4]:

```
pdist[pdist['no of order']<5].sum()
```

Out[4]:

```
no of order     10.000000
probability      0.682107
dtype: float64
```

In [5]:

```
pdist[pdist['no of order']>10].sum()
```

Out[5]:

```
no of order     270.000000
probability       0.000495
dtype: float64
```

Task 1. c

In [6]:

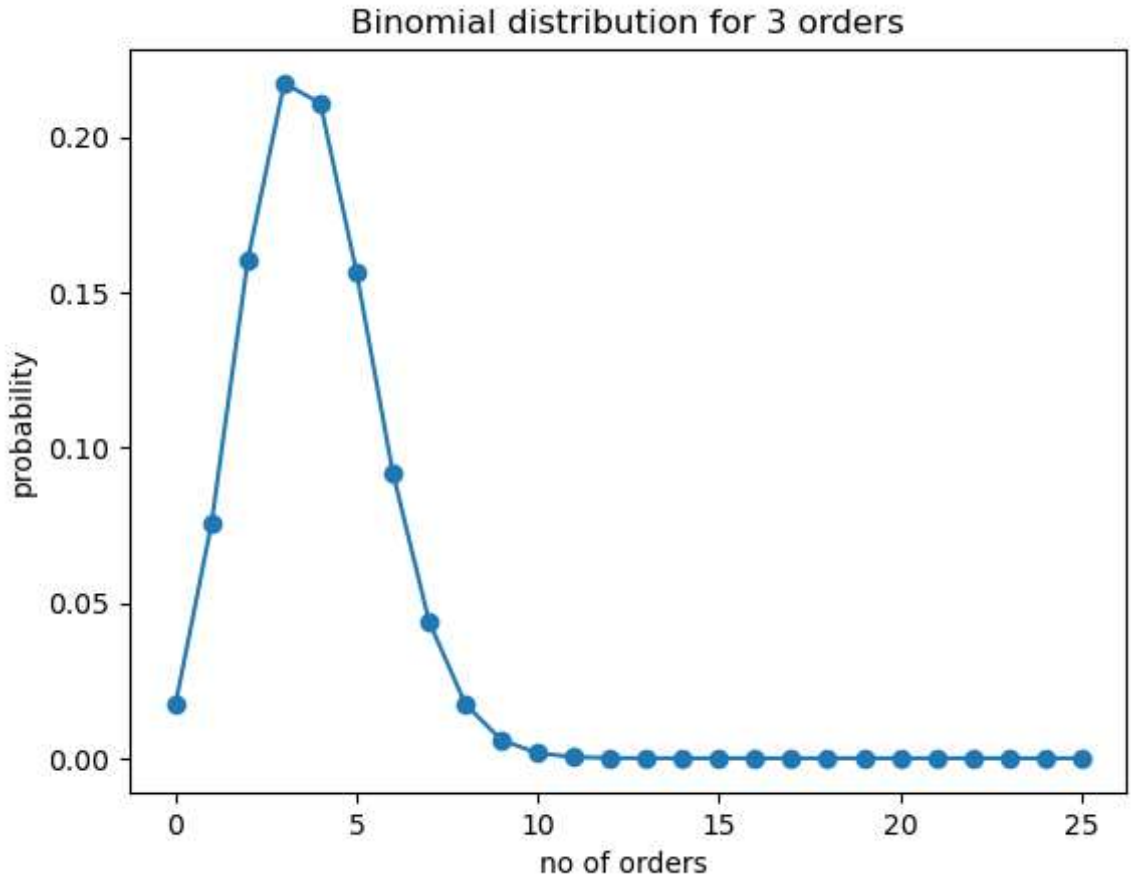
```
meanb = n*p
varianceb=n*p*q
standard_davb=varianceb**0.5
print("mean: ",meanb)
print("variance :", varianceb)
print("standard deviation:" , standard_davb)
```

```
mean:  3.75
variance : 3.1875
standard deviation: 1.7853571071357126
```

Task 1.d

In [7]:

```
plt.plot(k,binomial,'o-')
plt.title("Binomial distribution for 3 orders")
plt.xlabel("no of orders ")
plt.ylabel("probability")
plt.show()
```



Task 2

In [8]:

```
from statistics import mean
```

Task 2 a

In [9]:

```
week_all3 = [3,6,4,6,2,3,1,5,1,0,3,3,1,2,4,0,2,6,5,4,2,5,3,4,5,3,5,3,5,4,7,3,4,8,3,1]
avg = mean(week_all3)
avg
```

Out[9]:

```
3.5
```

In [10]:

```
from scipy.stats import poisson
rate = 3.5
n =np.arange(0,9) # no of customer arriving every 5 min
poisson_data = poisson.pmf(n,rate)
poisson_data
```

Out[10]:

```
array([0.03019738, 0.10569084, 0.18495897, 0.21578547, 0.18881229,
       0.1321686 , 0.07709835, 0.03854917, 0.01686526])
```

No customer in 5 min interval

In [11]:

```
print("probability of No customer in 5 min interval :",poisson_data[0] )
```

```
probability of No customer in 5 min interval : 0.0301973834223185
```

Task 2.b : Probability of 6 or more customer arrive in 5 mins

In [12]:

```
rate =3.5
n = np.arange(6,11)
poisson_data6 = poisson.pmf(n,rate)
poisson_data6
```

Out[12]:

```
array([0.07709835, 0.03854917, 0.01686526, 0.00655871, 0.00229555])
```

In [13]:

```
print(" Probability of 6 or more customer arrive in 5 mins :",np.sum(poisson_data6))
```

```
 Probability of 6 or more customer arrive in 5 mins : 0.1413670524666047
```

Task 2.c : probability of less than 4 arrive in 10 mins

In [32]:

```
rate = 3.5*2
n =np.arange(0,4) # no of customer arriving every 5 min
poisson_data10_min = poisson.pmf(n,rate)
poisson_data10_min
```

Out[32]:

```
array([0.00091188, 0.00638317, 0.02234111, 0.05212925])
```

In [33]:

```
print("probability of less than 4 customer arrive in 10 mins  :",np.sum(poisson_data10_min))
```

```
probability of less than 4 customer arrive in 10 mins  : 0.08176541624472157
```

Task 2.d : Probability that between 3 and 6 customers in 10 mins

In [16]:

```
rate = 3.5*2
n =np.arange(4,7) # no of customer arriving every 10 min
poisson_data_between10_min = poisson.pmf(n,rate)
poisson_data_between10_min
```

Out[16]:

```
array([0.05212925, 0.09122619, 0.12771667, 0.14900278])
```

In [17]:

```
print("Probability that between 3 and 6 customers in 10  mins  :",np.sum(poisson_data_betwe
```

```
Probability that between 3 and 6 customers in 10  mins  : 0.4200748919681766
6
```

Task 2.e : Probability that exactly 8 customers arrive in 15 mins

In [18]:

```
rate = 3.5*3
n =np.arange(0,9)
poisson_data1= poisson.pmf(n,rate)
poisson_data1
```

Out[18]:

```
array([2.75364493e-05, 2.89132718e-04, 1.51794677e-03, 5.31281370e-03,
       1.39461360e-02, 2.92868855e-02, 5.12520496e-02, 7.68780744e-02,
       1.00902473e-01])
```

In [19]:

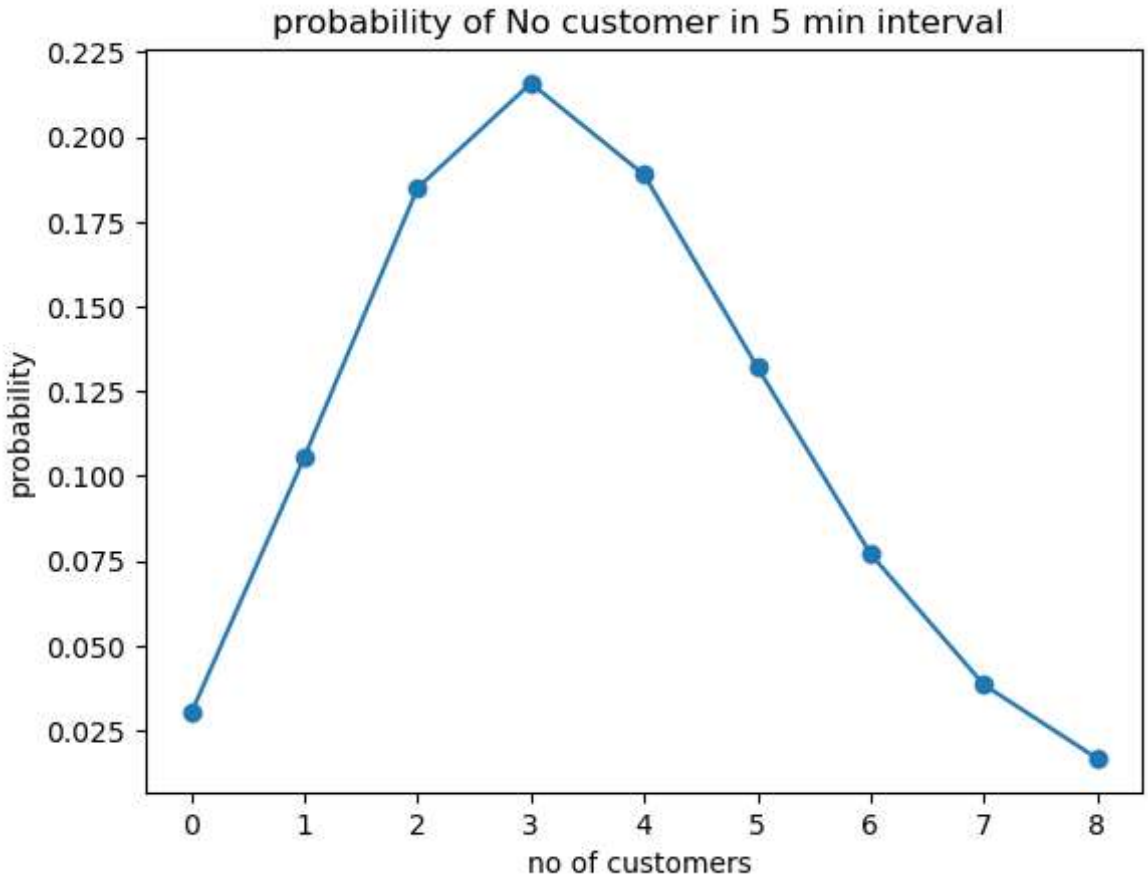
```
print("Probability that exactly 8 customers arrive in 15 mins  :", poisson_data[8])
```

```
Probability that exactly 8 customers arrive in 15 mins  : 0.0168652640352148
7
```

Task 2.f

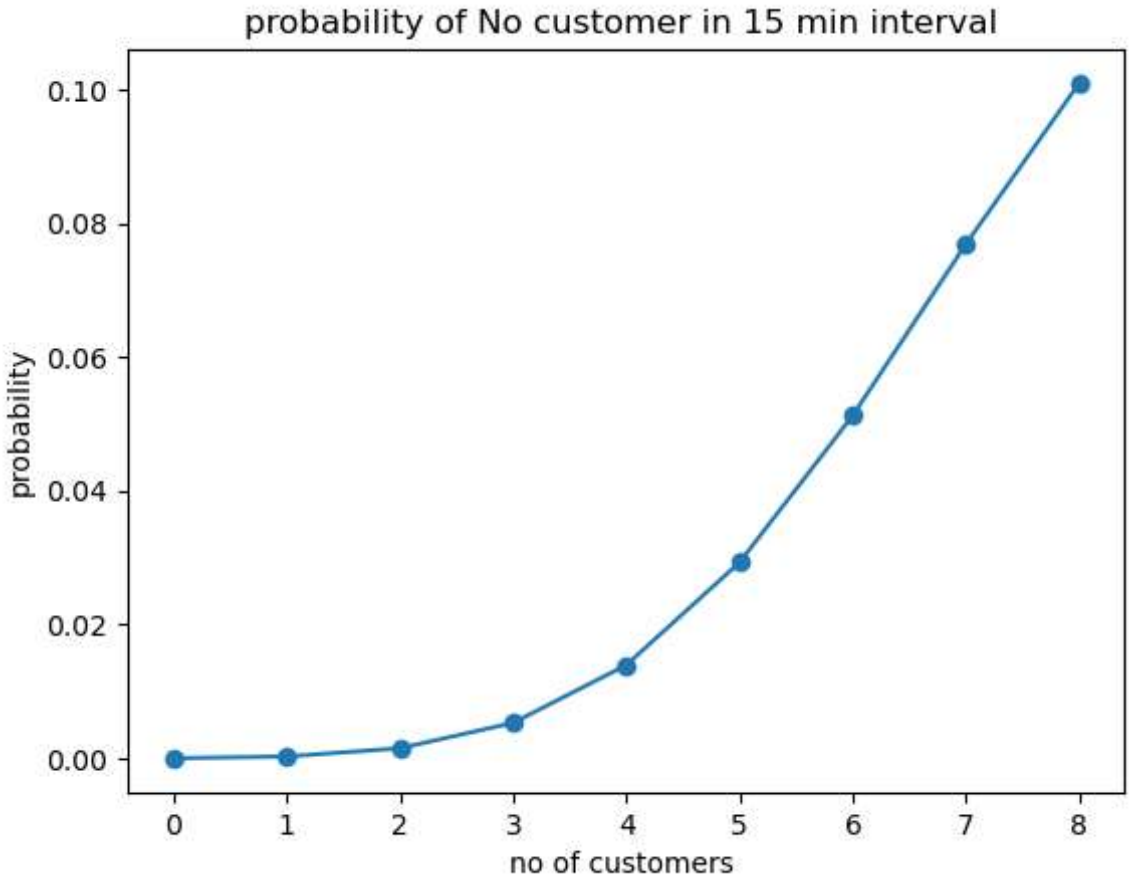
In [20]:

```
plt.plot(n,poisson_data,'o-')
plt.title("probability of No customer in 5 min interval ")
plt.xlabel("no of customers ")
plt.ylabel("probability")
plt.show()
```



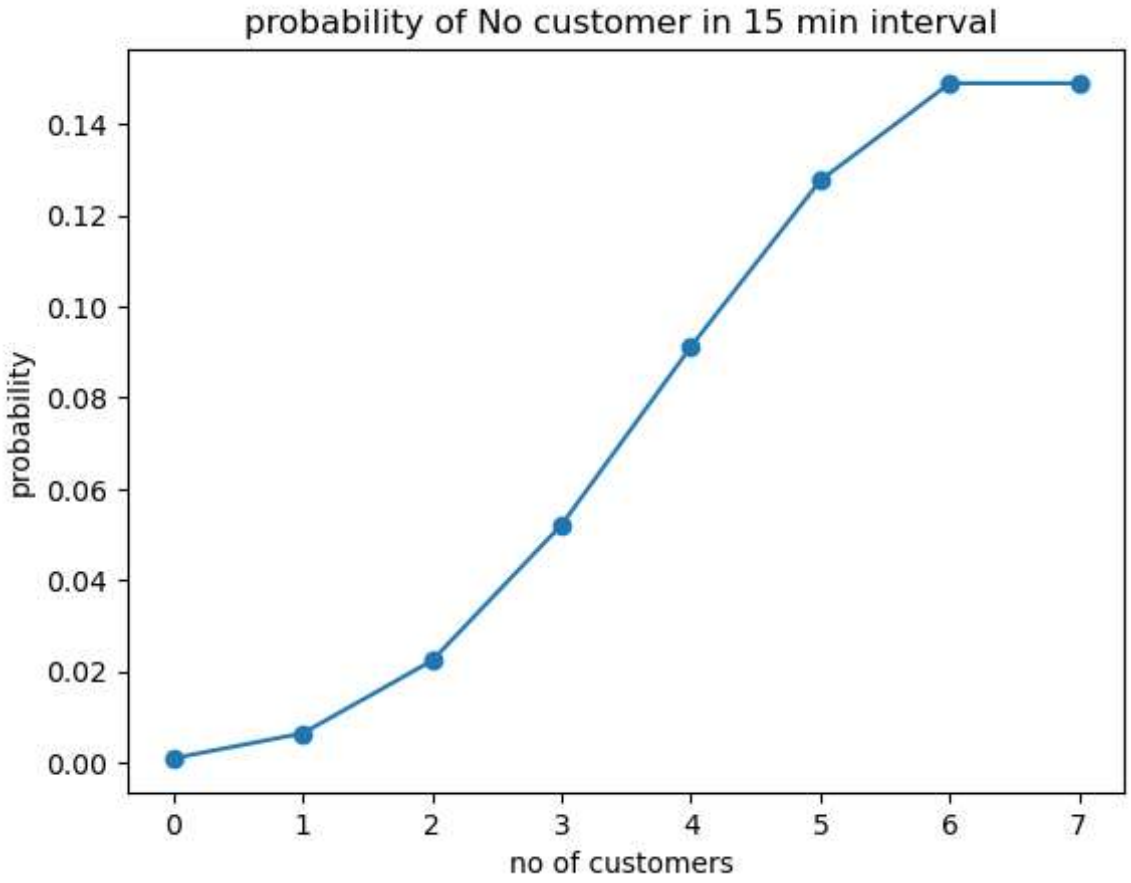
In [31]:

```
plt.plot(poisson_data1,'o-')
plt.title("probability of No customer in 15 min interval ")
plt.xlabel("no of customers ")
plt.ylabel("probability")
plt.show()
```



In [30]:

```
plt.plot(poisson_data10_min,'o-')
plt.title("probability of No customer in 15 min interval ")
plt.xlabel("no of customers ")
plt.ylabel("probability")
plt.show()
```



In [ ]: