DSL1_C5_S4_Challenge

In [1]:

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

Task 1

In [28]:

```
X = [6,7,8,9,10,11,12,13,14]
p_x = [0.03, 0.08, 0.15, 0.20, 0.19, 0.16, 0.10, ]
```

In [29]:

```
dataf = pd.DataFrame ({'weather ': X, 'probabolity value':p_x})
dataf
```

Out[29]:

	weather	probabolity value
0	6	0.03
1	7	0.08
2	8	0.15
3	9	0.20
4	10	0.19
5	11	0.16
6	12	0.10
7	13	0.07
8	14	0.02

Task 1.a: less than 9

In [33]:

```
dataf[:4]
```

Out[33]:

	weather	probabolity value
0	6	0.03
1	7	0.08
2	8	0.15
3	9	0.20

In [34]:

```
print("The probability of getting sum of less than 9 ", dataf['probabolity value'][:4].sum(
```

```
The probability of getting sum of less than 9  0.46
```

Task 1.b: 7 to 13

In [36]:

```
dataf[1:8]
```

Out[36]:

	weather	probabolity value
1	7	0.08
2	8	0.15
3	9	0.20
4	10	0.19
5	11	0.16
6	12	0.10
7	13	0.07

In [37]:

```
print("The probability of getting sum of 7 to 9 ", dataf['probabolity value'][1:8].sum())
```

```
The probability of getting sum of 7 to 9  0.95
```

Task 1.c : lost all summer

In [55]:

```
print("the probability of lost all summer ", 1-(dataf['probabolity value'].sum()))
```

```
the probability of lost all summer  0.0
```

Task 1.d : compute mean and sd

In [38]:

```
mean =0
for x, p in zip(X,p_x):
    mean=mean+x*p
print("Expected value is :", mean)
```

```
Expected value is : 9.790000000000001
```

In [39]:

```
variance =0
for x, p in zip(X,p_x):
    variance = variance+((x - mean)**2)*p
print("variance is :", variance)
```

```
variance is : 3.4659
```

In [40]:

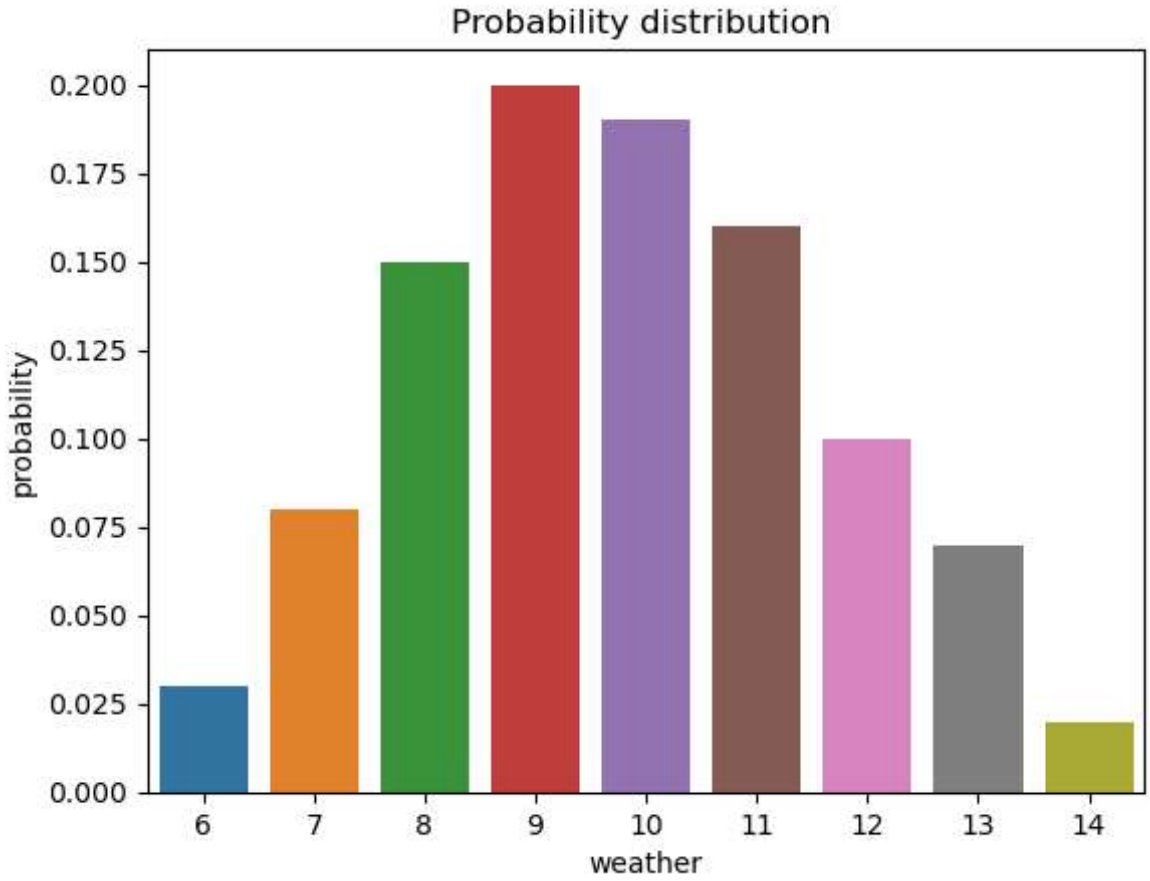
```
print("Standard deviation", variance**(0.5))
```

```
Standard deviation 1.8616927780920245
```

Task 1.e: Distribution plot

In [41]:

```
sns.barplot(x=X, y = p_x)
plt.title("Probability distribution ")
plt.xlabel("weather")
plt.ylabel("probability")
plt.show()
```



Task 2

In [21]:

```
p =0.20
q = 1-p
n=15
k = np.arange(0,16)
from scipy.stats import binom
binomial = binom.pmf(k,n,p)
binomial
```

Out[21]:

```
array([3.51843721e-02, 1.31941395e-01, 2.30897442e-01, 2.50138895e-01,
       1.87604171e-01, 1.03182294e-01, 4.29926226e-02, 1.38190573e-02,
       3.45476432e-03, 6.71759729e-04, 1.00763959e-04, 1.14504499e-05,
       9.54204160e-07, 5.50502400e-08, 1.96608000e-09, 3.27680000e-11])
```

Task 2.a : exactly 5 overseas

In [22]:

```
print("probability  that exactly 5 overseas",binomial[5])
```

```
probability  that all the value filled non correctly 0.10318229431910408
```

Task 2.b : more than 9 overseas

In [20]:

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

Out[20]:

	no of overseas	probability
0	0	3.518437e-02
1	1	1.319414e-01
2	2	2.308974e-01
3	3	2.501389e-01
4	4	1.876042e-01
5	5	1.031823e-01
6	6	4.299262e-02
7	7	1.381906e-02
8	8	3.454764e-03
9	9	6.717597e-04
10	10	1.007640e-04
11	11	1.145045e-05
12	12	9.542042e-07
13	13	5.505024e-08
14	14	1.966080e-09

In [23]:

```
pdist[pdist['no of overseas']>9].sum()
```

Out[23]:

```
no of overseas    60.000000
probability        0.000113
dtype: float64
```

Task 2.c: none of the overseas

In [24]:

```
print("probability  that none of the overseas",binomial[0])
```

```
probability  that none of the overseas 0.03518437208883203
```

Task 2.d : 4 to 7 overseas

In [25]:

```
p_4to_7 = 1-binom.cdf(4,7,p)
p_4to_7
```

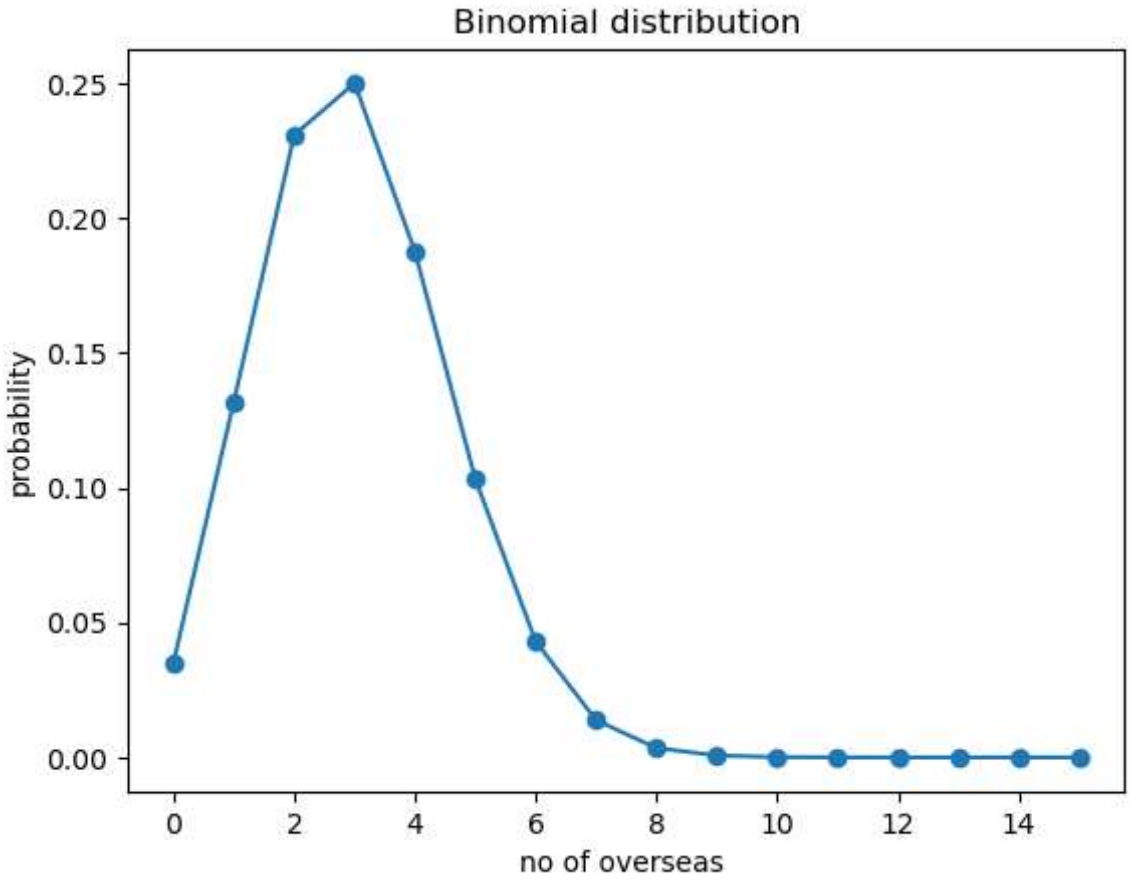
Out[25]:

```
0.0046720000000000095
```

Task 2.e : construct graph of binomial distribution

In [27]:

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



Task 3.a: Family did not make a trip

In [2]:

```
from scipy.stats import poisson
rate = 0.548
n =np.arange(0,9) # no of customer arriving every 5 min
poisson_data = poisson.pmf(n,rate)
poisson_data
```

Out[2]:

```
array([5.78104865e-01, 3.16801466e-01, 8.68036016e-02, 1.58561246e-02,
       2.17228907e-03, 2.38082882e-04, 2.17449032e-05, 1.70231528e-06,
       1.16608597e-07])
```

In [3]:

```
print("probability of Family did not make a trip :",poisson_data[0] )
```

```
probability of Family did not make a trip : 0.5781048646705196
```

Task 3.b: family took exactly one trip

In [5]:

```
print("probability of Family took exactly one trip :",poisson_data[1] )
```

```
probability of Family took exactly one trip : 0.3168014658394448
```

Task 3.c: family took 2 or more trip

In [13]:

```
rate = 0.548
n =np.arange(2,9) # no of customer arriving every 5 min
poisson_data1 = poisson.pmf(n,rate)
poisson_data1
```

Out[13]:

```
array([8.68036016e-02, 1.58561246e-02, 2.17228907e-03, 2.38082882e-04,
       2.17449032e-05, 1.70231528e-06, 1.16608597e-07])
```

In [14]:

```
print(" Probability of family took 2 or more trip :",np.sum(poisson_data1))
```

```
 Probability of family took 2 or more trip : 0.10509366198047082
```

Task 3.d: family took 3 or less than 3 trip over 3 years

In [10]:

```
rate = 0.548*3
n =np.arange(0,3) # no of customer arriving every 5 min
poisson_data2 = poisson.pmf(n,rate)
poisson_data2
```

Out[10]:

```
array([0.19320567, 0.31763012, 0.26109196])
```

In [11]:

```
print(" Probability of family took 3 or less than 3 trip over 3 years :",np.sum(poisson_dat
```

```
 Probability of family took 3 or less than 3 trip over 3 years : 0.771927758
9078369
```

Task 3.e: family took exactly 4 trips during past 6 years

In [16]:

```
rate = 0.548*6
n =np.arange(0,9) # no of customer arriving every 5 min
poisson_data3 = poisson.pmf(n,rate)
poisson_data3
```

Out[16]:

```
array([0.03732843, 0.12273588, 0.20177779, 0.22114846, 0.18178403,
       0.11954118, 0.06550857, 0.03077031, 0.0126466 ])
```

In [17]:

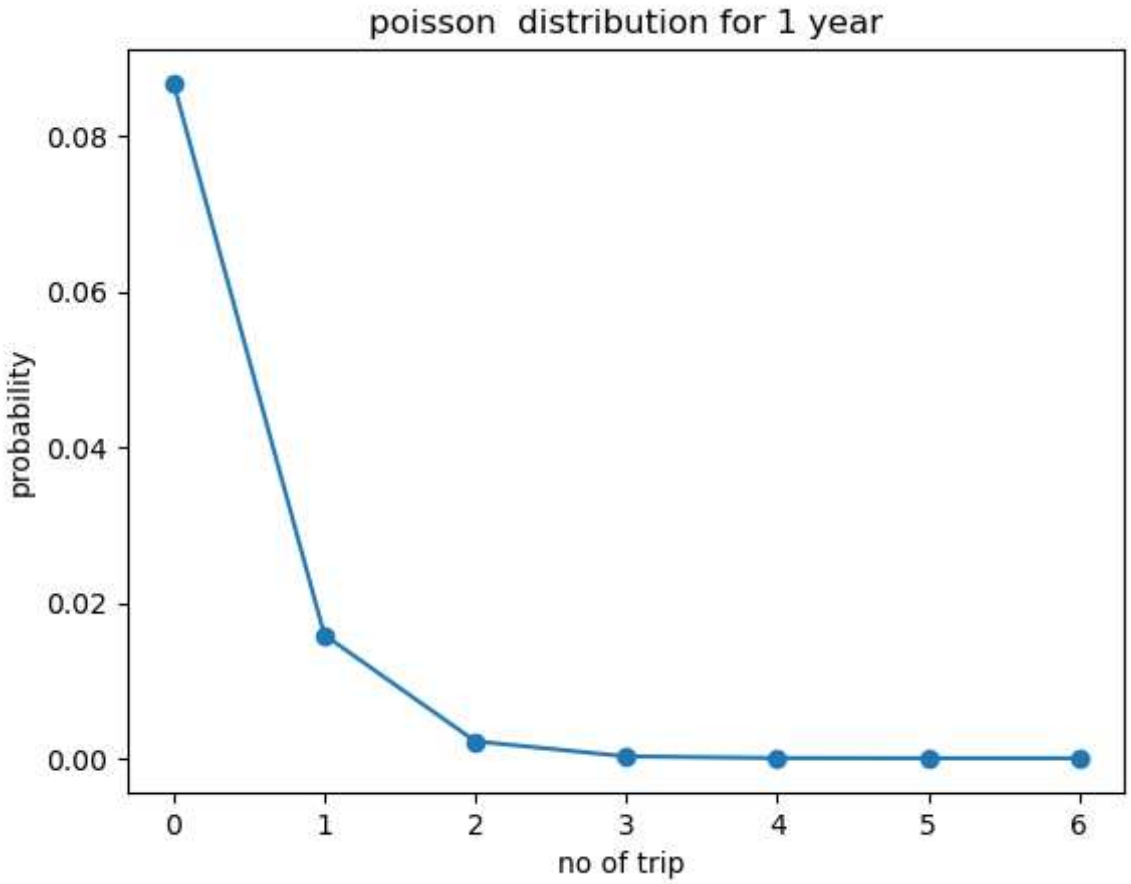
```
print(" Probability of family took exactly 4 trips during past 6 years :",poisson_data3[4])
```

```
 Probability of family took exactly 4 trips during past 6 years : 0.18178403
42372797
```

Task 3.f : Sketch the graphs

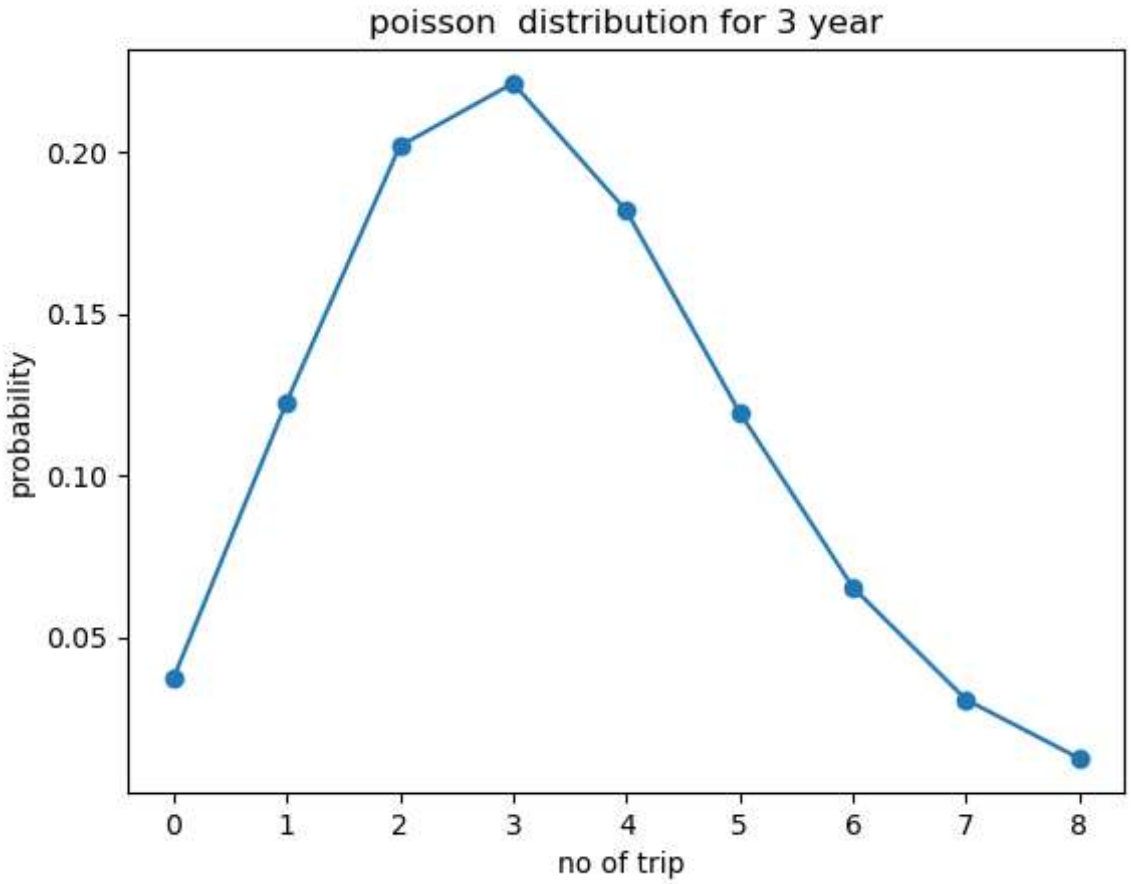
In [45]:

```
plt.plot(poisson_data,'o-')
plt.title("poisson  distribution for 1 year")
plt.xlabel("no of trip ")
plt.ylabel("probability")
plt.show()
```



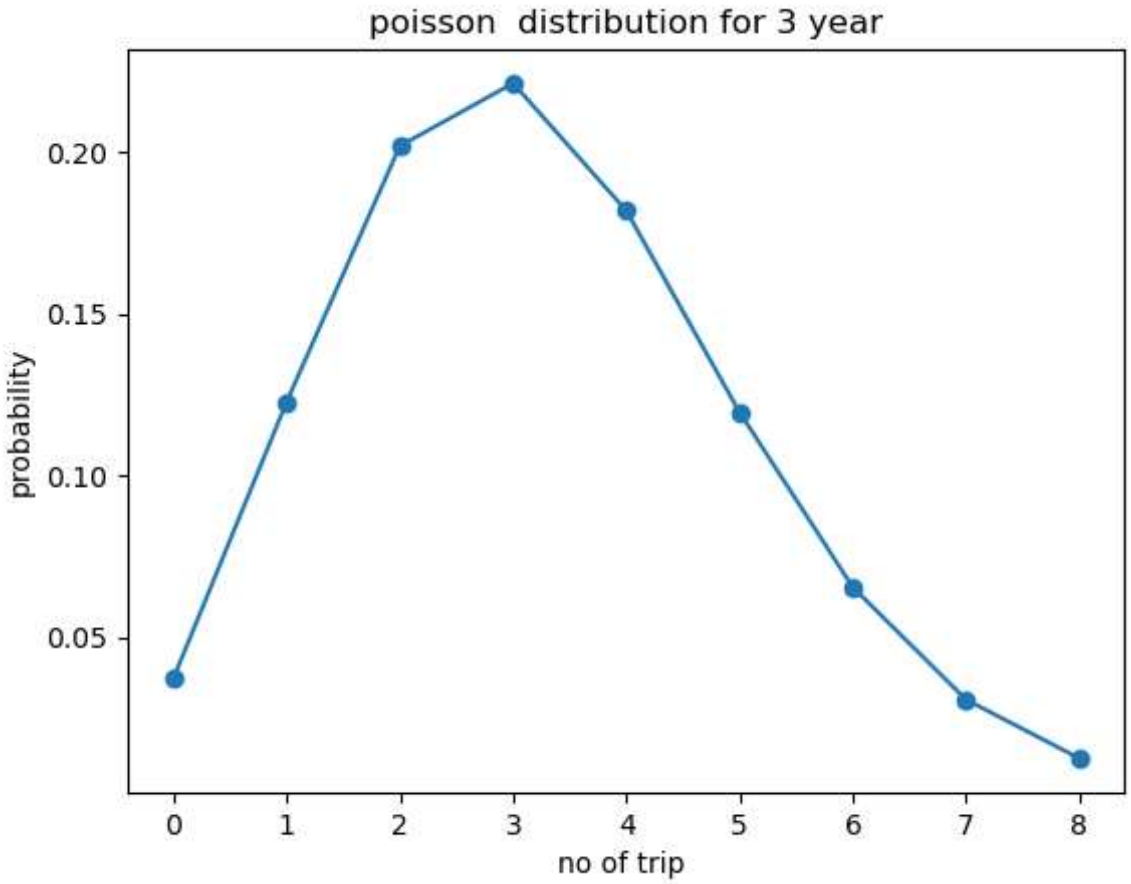
In [46]:

```
plt.plot(poisson_data2,'o-')
plt.title("poisson  distribution for 3 year")
plt.xlabel("no of trip ")
plt.ylabel("probability")
plt.show()
```



In [47]:

```
plt.plot(poisson_data3,'o-')
plt.title("poisson  distribution for 3 year")
plt.xlabel("no of trip ")
plt.ylabel("probability")
plt.show()
```



In [ ]: