In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
```

In [2]:
```
import warnings
warnings.filterwarnings('ignore')
```

Discrete Probability Distribution: Binomial

- A T-shirt manufacturing company regularly conducts quality checks at specified period on the products it manufactures. Their empirical data of manufacturing report says,4% of their products were defective. Suppose a random sample of 250 T-shirts were picked from the manufacturing unit,what is the probability that,
- a) None of the T-shirts are defective
- b) 5 or fewer T-shirts are defective
- c) 20 or more T-shirts are defective
- d) What is the mean value of this Binomial Distribution
- e) Plot the Binomial Distribution

In [4]:
```
import scipy.stats as stats
```

In [23]:
```
#a) we need to calculate P(x=0) = n C x*P^x*Q^(n-x)
#Here n=250, x=0, P=defective % = 4%=0.04, Q= Non-defective % = 96% = 0.96
n=250
p=0.04
stats.binom.pmf(0,n,p)
```

Out[23]:
```
3.696649374485171e-05
```

In [ ]:
```
#b) 5 or fewer => P(x<=5)= P(x=0)+P(x=1)+P(x=2)+P(x=3)+P(x=4)+P(x=5)
stats.binom.cdf(5,n,p)
#OR
stats.binom.pmf(0,n,p)+stats.binom.pmf(1,n,p)+stats.binom.pmf(2,n,p)+stats.binom.pmf(3,n,p)
+stats.binom.pmf(4,n,p)+stats.binom.pmf(5,n,p)
```

Out[56]:
```
0.06329328366342199
```

In [17]:
```
#c) 20 or more => P(x>=20)= P(x=20)+........+ all cummulative probabilities in right hand side
1-stats.binom.cdf(19,n,p)
# by default function can calculate only from the left hand side
```

Out[17]:
```
0.002814404938111137
```

In [ ]:
```
#d)Mean value of Binomial distribution = n*p
#Here, mean implies 4% of 250
0.04*250
```

Out[58]:
```
10.0
```

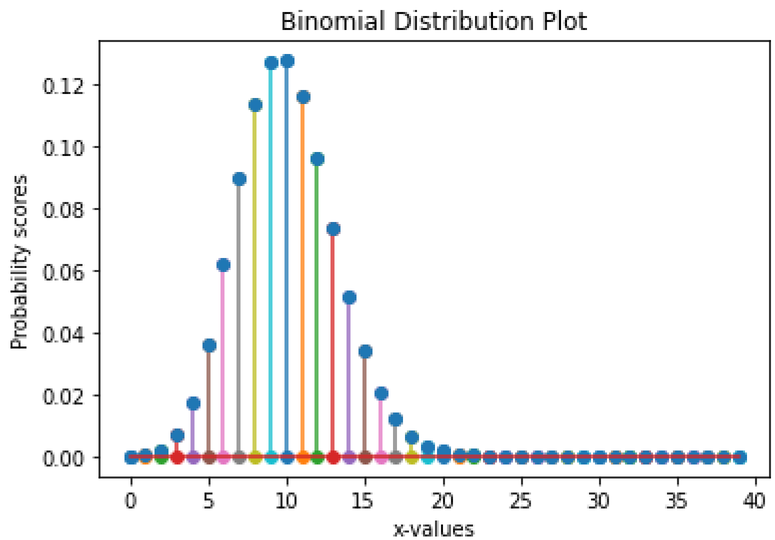
In [ ]:
```
stats.binom.pmf(10,n,p)
```

Out[63]:
```
0.12768812205208388
```

In [24]:
```
binom_dist
```

Out[24]:
```
array([3.69664937e-05, 3.85067643e-04, 1.99753840e-03, 6.88041004e-03,
       1.77027217e-02, 3.62905794e-02, 6.17443886e-02, 8.96763739e-02,
       1.13496661e-01, 1.27158296e-01, 1.27688122e-01, 1.16080111e-01,
       9.63303699e-02, 7.34827821e-02, 5.18316053e-02, 3.39784968e-02,
       2.07941321e-02, 1.19260464e-02, 6.43233520e-03, 3.27259159e-03,
       1.57493470e-03, 7.18720202e-04, 3.11717663e-04, 1.28752948e-04,
       5.07411791e-05, 1.91125108e-05, 6.89153034e-06, 2.38225740e-06,
       7.90540774e-07, 2.52155247e-07, 7.73976522e-08, 2.28864025e-08,
       6.52620072e-09, 1.79635323e-09, 4.77706680e-10, 1.22838861e-10,
       3.05675405e-11, 7.36650188e-12, 1.72046590e-12, 3.89678174e-13])
```

In [27]:
```
#e) to generate the plot,we need to define x-axis
x=np.arange(0,40)
binom_dist=stats.binom.pmf(x,n,p)
plt.stem(x,binom_dist,'o-')
plt.xlabel('x-values')
plt.ylabel('Probability scores')
plt.title('Binomial Distribution Plot')
plt.show()
```



In [28]:
```
x
```

Out[28]:
```
array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
       17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
       34, 35, 36, 37, 38, 39])
```

In [29]:
```
x.dot(binom_dist) #this is analogous to the notion summation of x.p(x)
```

Out[29]:
```
9.999999999995323
```

In [30]:
```
#mean value in Binom distn = n*p
250*.04
```

Out[30]: 10.0

In [ ]:
```
np.argmax(binom_dist),np.max(binom_dist)
```

Out[19]: (10, 0.12768812205208388)

In [31]:
```
sd=np.sqrt(((x-10)**2).dot(binom_dist))  #more generic formula
sd
```

Out[31]: 3.098386676949792

In [32]:
```
np.sqrt(n*p*(1-p))
```

Out[32]: 3.0983866769659336

- We could notice,the peak probability score of 12.76% at the mean value (10 T-shirts)

Discrete Probability Distribution: Poisson

A customer care unit of a fiber optic broadband service provider receives on an average 30 emails on a day regarding complaint in their service. To strategically plan the resource of service engineers, the following calculations are performed

---

a) What is the probability of receiving 100 complaints in a given day?

b)What is the probability of receiving 20 or less complaints in a given day?

c)What is the probability of receiving 35 or more complaints in a given day?

d)Plot the poisson distribution

In [33]:
```
#a) we need to calculate P(x=100)
#Here Lambda = 30 emails (average)
L=30
stats.poisson.pmf(100,L)
```

Out[33]:
```
5.16758180183853 96e-24
```

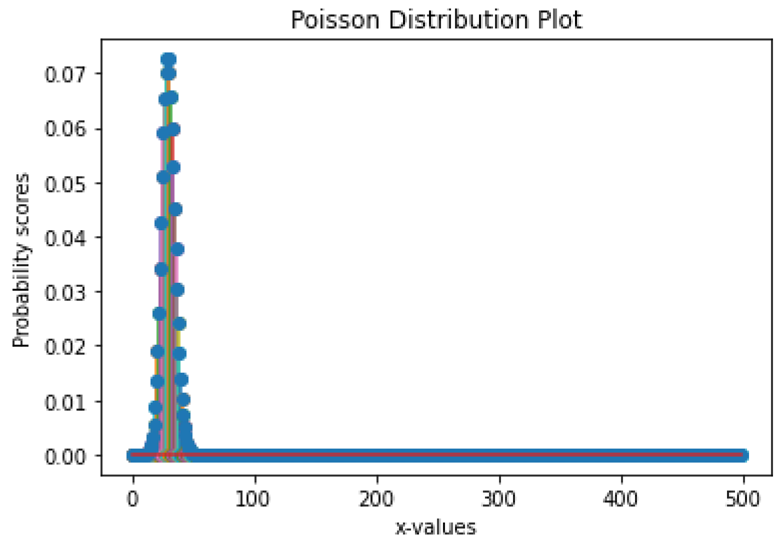
In [35]:
```
#b)P(x<=20)
stats.poisson.cdf(20,L)
```

Out[35]:
```
0.035284618454228846
```

In [36]:
```
#c) P(x>=35)
1-stats.poisson.cdf(34,L)
```

Out[36]:
```
0.20269167451688286
```

In [44]:
```
x=np.arange(0,500)
pois_dist=stats.poisson.pmf(x,L)
plt.stem(x,pois_dist,'o-')
plt.xlabel('x-values')
plt.ylabel('Probability scores')
plt.title('Poisson Distribution Plot')
plt.show()
```



In [ ]:
```
np.argmax(pois_dist)
```

Out[73]: 29

In [42]:
```
pois_dist[29]
```

Out[42]: 0.07263452647159181

In [45]:
```
pois_dist[30]
```

Out[45]: 0.07263452647159181

- We can notice the highest probability centered around the mean (ie) 30 complaints

In [48]:
```
np.sqrt(x.dot(pois_dist))
```

Out[48]:
```
5.47722557505168
```

In [ ]:
```
np.sqrt(x.dot(pois_dist)) #cross checking the mean value from histogram perspective x.P(x)
```

Out[80]:
```
5.477220430202007
```

In [47]:
```
np.sqrt(((x-30)**2).dot(pois_dist))
```

Out[47]:
```
5.477225575051677
```

In [ ]: