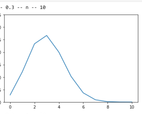
Exercise 3

Problem 3.1

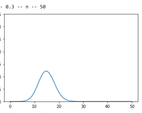
In [1]:
```
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from scipy.stats import binom
%matplotlib inline

def binomial_func():
    p_list = [0.3, 0.5, 0.8]
    n_list = [10, 50]
    for i in p_list:
        for j in n_list:
            print('p --',i,'-- n --',j)
            X = range(j+1)
            Y = binom.pmf(X, n = j, p = i)
            plt.plot(X,Y,'-')
            plt.ylim([0, 0.35])
            plt.show()

binomial_func()
```

missing x/y-labels -1

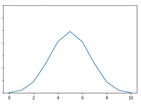
p -- 0.3 -- n -- 10



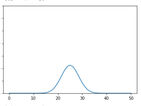
p -- 0.3 -- n -- 50



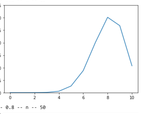
p -- 0.5 -- n -- 10



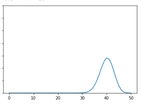
p -- 0.5 -- n -- 50



p -- 0.8 -- n -- 10



p -- 0.8 -- n -- 50

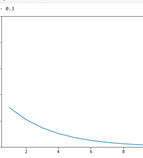


Problem 3.2

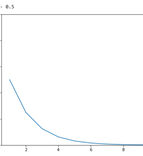
Geometric distribution

In [2]:
```
from scipy.stats import geom
import matplotlib.pyplot as plt
import seaborn as sns

X = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]

def geom_distribution():
    p_list = [0.3, 0.5, 0.8]
    for i in p_list:
        print('p --',i)
        geom_pd = geom.pmf(X, i)
        fig, ax = plt.subplots(1, 1, figsize=(8, 6))
        ax.plot(X, geom_pd, '-', ms=8, label='geom cdf')
        plt.ylim([0, 1])
        plt.show()

geom_distribution()
```

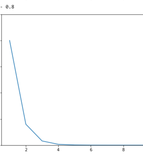
p -- 0.3



p -- 0.5



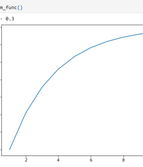
p -- 0.8



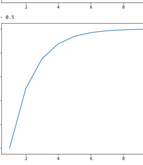
cumulative distribution

In [3]:
```
from scipy.stats import geom
import matplotlib.pyplot as plt
import seaborn as sns

X = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]

def geom_func():
    p_list = [0.3, 0.5, 0.8]
    for i in p_list:
        print('p --',i)
        geom_pd = geom.cdf(X, i)
        fig, ax = plt.subplots(1, 1, figsize=(8, 6))
        ax.plot(X, geom_pd, '-', ms=8, label='geom cdf')
        plt.show()

geom_func()
```

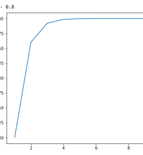
p -- 0.3



p -- 0.5



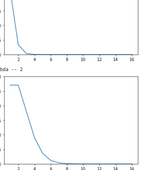
p -- 0.8



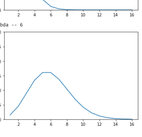
Problem 3.3

In [4]:
```
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from scipy.stats import poisson
%matplotlib inline

X = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16]

def poisson_func():
    lambdas = [0.3, 2, 6]
    for i in lambdas:
        print('Lambda =',i)
        Y = poisson.pmf(X, i)
        plt.plot(X,Y,'-')
        plt.ylim([0, 0.3])
        plt.show()

poisson_func()
```

Lambda -- 0.3



Lambda -- 2



Lambda -- 6

1/12