Session 12: Problem Solving with Simulation (Solutions Only)

Q1. Forecasting the Spread of a New Virus

A new virus has broken out in a city and has an incubation period of $d = 14$ days. Starting from $a = 2$ days after infection to the last day of the incubation period, each infected patient has close contact with $n = 6$ uninfected people, and infects each of them with probability $p = 0.05$ independently from others. At the end of the last day of incubation, each infected person reports to the hospital and enters isolation, which means that they stop infecting others. Suppose that there is exactly 1 infected person on Day 0, create a function that simulates the number of patients who report to the hospital at the end of Day 1 through Day m.

The function should be called `simulateNewCases` and has the following arguments:

- `m`: the number of days to simulate.
- `a` (default value 2): the minimum day after infection when a patient becomes contagious.
- `d` (default value 14): the day after infection in which symptoms begin.
- `n` (default value 6): the number of uninfected individuals a person contacts each day.
- `p` (default value 0.05): the probability of infecting a close contact each day.

The function should return a list of m integers, representing the number of infected individuals who exit incubation and report to the hospital at the end of Day 1 through Day m.

To illustrate the timeline. Suppose that a person is infected on Day 5, then the person becomes contagious on Day $5 + a = 7$ and starting on that day, has the capacity to infect up to n people per day. At the end Day $5 + d = 19$, after possibly infecting new people on that day, the patient reports to the hospital and enters into isolation.

```
[4]: import numpy as np
     def simulateNewCases(m,a=2,d=14,n=6,p=0.05):
         infections=[1]
         newCases=[]
         for day in range(1,m+1):
             if day>=a:
                 beginDay=max(day-d,0)
                 endDay=day-a
                 numContagious=sum(infections[beginDay:endDay+1])
                 infections.append(np.random.binomial(n*numContagious,p))
             else:
                 infections.append(0)
             if day>=d:
                 newCases.append(infections[day-d])
             else:
                 newCases.append(0)
         return newCases
```

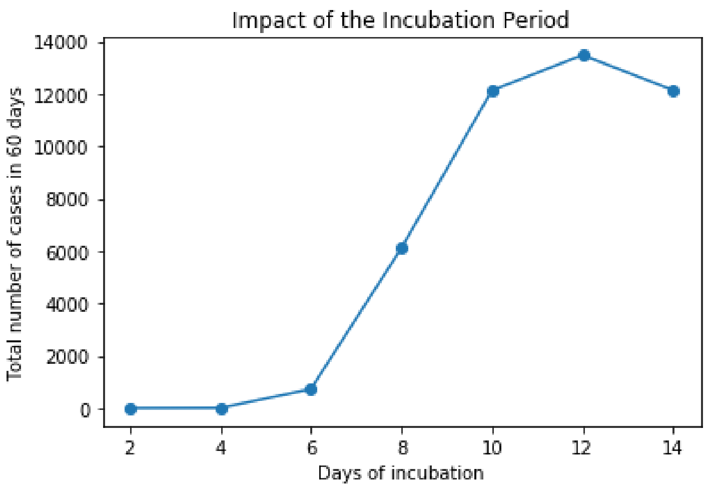
Q3. Self-Directed Exploration

Here are some examples of potentially interesting analyses.

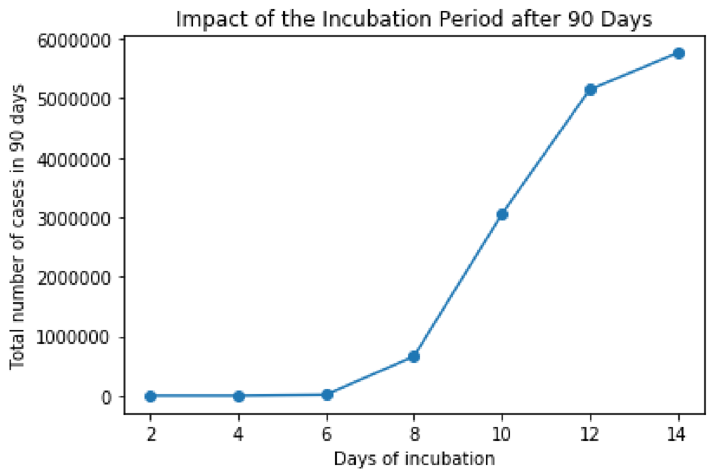
Impact of the Incubation Period

What is the impact of the incubation period on the number of cases in the first 60 days?

```
[9]: import pandas as pd
     import matplotlib.pyplot as plt
     dList=range(2,15,2)
     numCases=[]
     for d in dList:
         meanPrediction=pd.Series(\
             [sum(simulateNewCases(60,d=d)) for i in range(1000)]).mean()
         numCases.append(meanPrediction)
     plt.title('Impact of the Incubation Period')
     plt.xlabel('Days of incubation')
     plt.ylabel('Total number of cases in 60 days')
     plt.plot(dList,numCases,'o-')
```



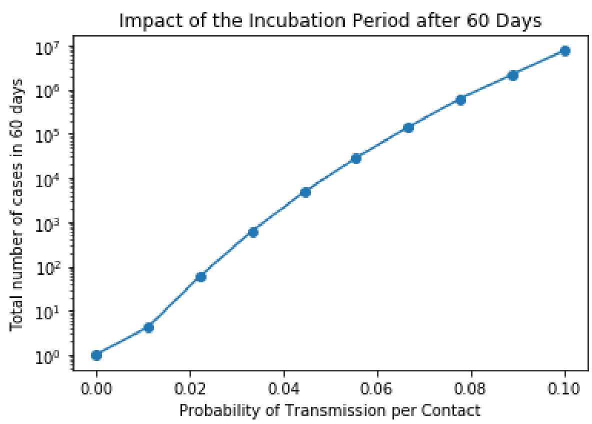

```
[10]: numCases=[]
      for d in dList:
          meanPrediction=pd.Series([\
              sum(simulateNewCases(90,d=d)) for i in range(1000)]).mean()
          numCases.append(meanPrediction)
      plt.title('Impact of the Incubation Period after 90 Days')
      plt.xlabel('Days of incubation')
      plt.ylabel('Total number of cases in 90 days')
      plt.plot(dList,numCases,'o-')
```



Impact of Ease of Contagion

How does the expected total number of cases change based on p?

[11]:
```
import numpy as np
pList=np.linspace(0,0.1,10)
numCases=[]
for p in pList:
    meanPrediction=np.mean(\
        [sum(simulateNewCases(60,p=p)) for i in range(1000)])
    numCases.append(meanPrediction)
plt.title('Impact of the Incubation Period after 60 Days')
plt.xlabel('Probability of Transmission per Contact')
plt.ylabel('Total number of cases in 60 days')
plt.yscale('log')
plt.plot(pList,numCases,'o-')
```



Benefit of Early Intervention

Suppose that after 100 cases have been reported, the government starts a campaign encouraging citizens to take preventative measures such as frequent hand washing and face masks, which reduces the chance of transmission from $p = 0.05$ to $p = 0.02$, and reduces the number of contacts from $n = 6$ to $n = 2$. What would be the impact of this intervention? What about if the intervention starts only after 1000 cases have been reported?

```
[12]: import numpy as np
      def simulateNewCases2(m,a=2,d=14,n1=6,n2=2,p1=0.05,p2=0.02,thresh=100):
          p=p1
          n=n1
          infections=[1]
          newCases=[]
          for day in range(1,m+1):
              if sum(newCases)>=thresh:
                  p=p2
                  n=n2
              if day>=a:
                  beginDay=max(day-d,0)
                  endDay=day-a
                  numContagious=sum(infections[beginDay:endDay+1])
                  infections.append(np.random.binomial(n*numContagious,p))
              else:
                  infections.append(0)
              if day>=d:
                  newCases.append(infections[day-d])
              else:
                  newCases.append(0)
          return newCases
```

```
[13]: import pandas as pd
      df=pd.DataFrame()
      df['Intervention']=['Baseline',\
          'Intervention after 1000 Cases','Intervention after 100 Cases']
      df['Number of Cases']=\
          [np.mean([sum(simulateNewCases(60)) for i in range(1000)]),\
          np.mean([sum(simulateNewCases2(60,thresh=1000)) for i in range(1000)]),\
          np.mean([sum(simulateNewCases2(60,thresh=100)) for i in range(1000)])]
      df.plot(x='Intervention',y='Number of Cases',kind='barh',legend=False)
      plt.title('Number of Total Cases after 60 Days')
```