Week 5: Simulation Modeling

Session 10: Case Study in Epidemic Modeling

Brief Notes on Syntax

The following notes clarify potential confusions when looking at the practice problems and practice midterms posted on Blackboard.

Random Number Generation

Sometimes you see Python code that generate random numbers using an older syntax, which is slowly being phased out but should still work for now.

```
[1]: # New Syntax (from last class)
     from numpy.random import default_rng
     rng=default_rng()
     rng.binomial(10,0.4)
```

```
3
```

```
[2]: # Old Syntax that is still supported but may be phased out in the future.
     from numpy import random
     random.binomial(10,0.4)
```

```
4
```

Accessing Entries of a Pandas DataFrame

```
[3]: import pandas as pd
     pd.DataFrame([[5,3,4],[1,2,2]])
```

```
   0  1  2
0  5  3  4
1  1  2  2
```

```
[4]: # Labeling the rows and columns
     df=pd.DataFrame([[5,3,4],[1,2,2]],index=['MSBA','MBA'],columns=['USC','UCLA','Caltech'])
     df
```

```
      USC  UCLA  Caltech
MSBA    5     3        4
MBA     1     2        2
```

```
[5]: # .loc used to access by label
     df.loc['MSBA','UCLA']
```

```
3
```

```
[6]: # .iloc used to access by position
     df.iloc[0,1]
```

```
3
```

```
[7]: # direct indexing for accessing a column
     df['UCLA']
```

```
MSBA    3
MBA     2
Name: UCLA, dtype: int64
```

```
[8]: df['UCLA'].iloc[0]
```

```
3
```

```
[9]: df.loc['MSBA',:]
```

```
USC        5
UCLA       3
Caltech    4
Name: MSBA, dtype: int64
```

```
[10]: df.loc['MSBA',:].iloc[1]
```

```
3
```

```
[11]: # Iterating through a DataFrame via a for loop
      for row in df.index:
          for col in df.columns:
              print(df.loc[row,col],end='\t')
          print()
```

```
5       3       4
1       2       2
```

Case Study: Epidemics Modeling

A new virus has broken out in a city and has an incubation period of $d = 5$ days. Starting from $a = 2$ days after infection to the last day of the incubation period, each infected patient has close contact with $n = 4$ uninfected people per day, and infects each of them with probability $p = 0.2$ 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.

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 input arguments:

- `m`: the number of days to simulate.
- `initial` (default value 1): the number of individuals who are newly infected on Day 0.
- `a` (default value 2): the first day after infection when a patient becomes contagious.
- `d` (default value 5): the last day after infection when a patient may infect others (not in isolation).
- `n` (default value 4): the number of uninfected individuals a person has close contact with each day.
- `p` (default value 0.2): the probability of infecting each uninfected individual during a close contact.

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.

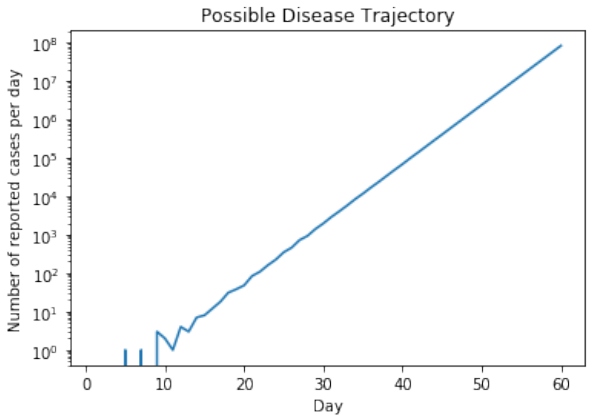
To illustrate the timeline. Suppose that a person is infected on Day 5, then the person becomes contagious on Day $5 + a = 7$ and starting on that day, has the capacity to infect up to n people per day. At the end Day $5 + d = 10$, after possibly infecting new people on that day, the patient reports to the hospital and enters into isolation.

Exercise 5.4 Applying Algorithmic Thinking to Epidemic Modeling

Download the Jupyter notebook attached to the link for this exercise on Blackboard and submit it there after completing it. The notebook asks you to apply the four steps of algorithmic thinking to solve the above problem.

In-Class exercise: Do Steps 1 and 2.

```
[28]: # Code to plot one possible disease trajectory. Re-run to see a different trajectory.
      import pandas as pd
      import matplotlib.pyplot as plt
      m=60
      s=pd.Series(simulateNewCases(m),index=range(1,m+1))
      s.plot(logy=True,title='Possible Disease Trajectory')
      plt.xlabel('Day')
      plt.ylabel('Number of reported cases per day')
      plt.show()
```



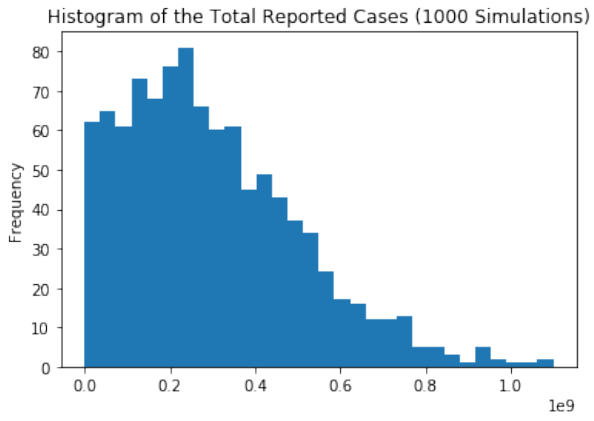
The following set of testing code uses your function to forecast the total number of reported cases within 60 days.

```
[20]: totalReports=pd.Series([sum(simulateNewCases(60)) for i in range(1000)])
```

```
[21]: print('Number of total reported cases in 60 days.')
      print(f'  Forecast mean: {totalReports.mean():.0f}')
      print(f'  Forecast standard deviation: {totalReports.std():.0f}')
```

```
Number of total reported cases in 60 days.
  Forecast mean: 299295014
  Forecast standard deviation: 206283849
```

```
[22]: totalReports.plot(kind='hist',bins=30,\
                       title="Histogram of the Total Reported Cases (1000 Simulations)")
      plt.show()
```

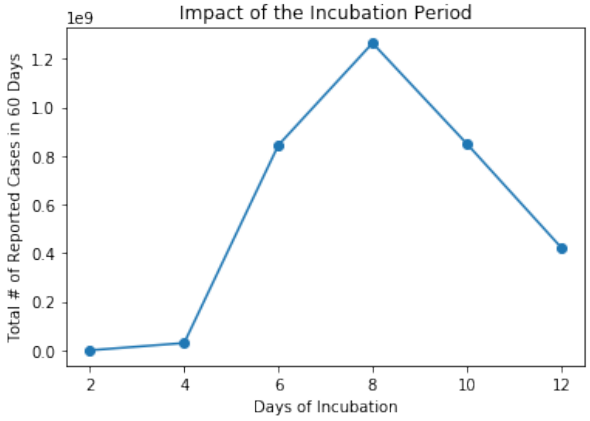


Answering Policy Relevant Questions

i) Impact of the Incubation Period

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

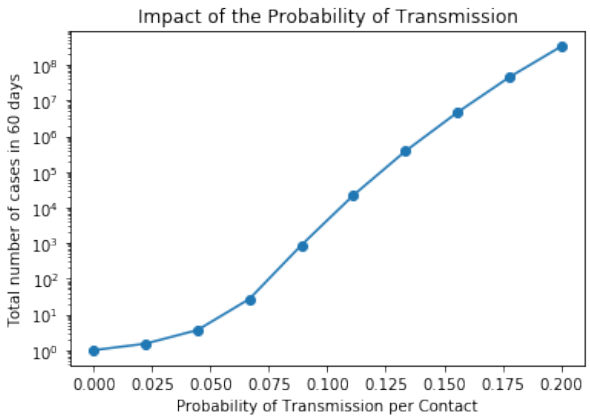
```
[23]: import pandas as pd
      import numpy as np
      import matplotlib.pyplot as plt
      numCases=pd.Series(dtype=float)
      for d in range(2,13,2):
          meanPrediction=np.mean([sum(simulateNewCases(60,d=d)) for i in range(1000)])
          numCases.loc[d]=meanPrediction
      numCases.plot(style='o-',title='Impact of the Incubation Period')
      plt.xlabel('Days of Incubation')
      plt.ylabel('Total # of Reported Cases in 60 Days')
      plt.show()
```



ii) Impact of Better Hygiene

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

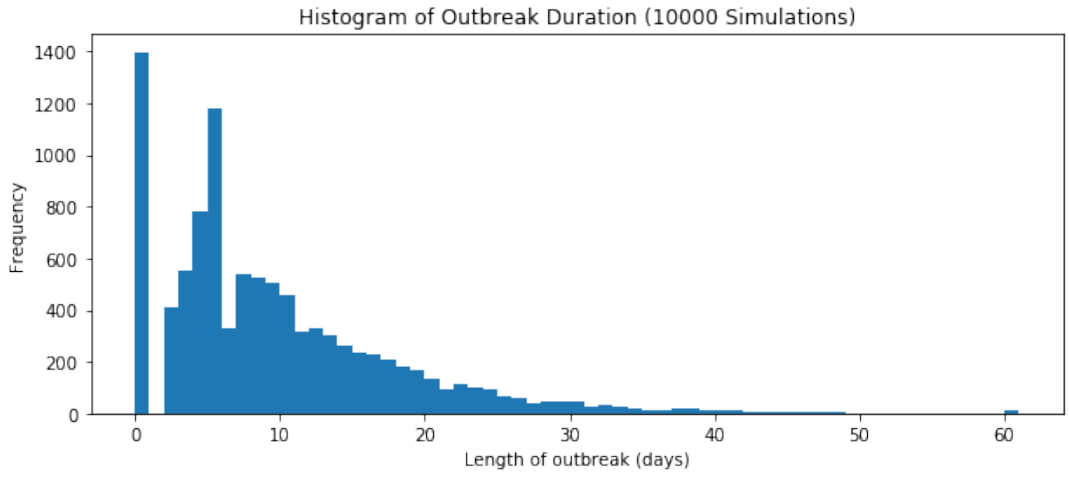
```
[24]: import numpy as np
      pList=np.linspace(0,0.2,10)
      numCases=pd.Series(dtype=float)
      for p in pList:
          numCases.loc[p]=np.mean([sum(simulateNewCases(60,p=p)) for i in range(1000)])
      numCases.plot(style='o-',title='Impact of the Probability of Transmission')
      plt.xlabel('Probability of Transmission per Contact')
      plt.ylabel('Total number of cases in 60 days')
      plt.yscale('log')
      plt.show()
```



iii) Length of an Outbreak after Herd Immunity

Suppose that most people in the city have been vaccinated, so that the parameter p is reduced by 5 fold to 0.04. Moreover, the city has been case free for a while. However, on day 0, 3 new people become infected. How many days would this outbreak last? For simplicity, we say that the outbreak has ended if there are no hospital reports for 14 days.

```
[25]: import pandas as pd
      import numpy as np
      length=[]
      for i in range(10000):
          reports=simulateNewCases(78,initial=3,p=0.04)
          noCaseDays=0
          endDay=61
          day=0
          for cases in reports:
              if cases==0:
                  noCaseDays+=1
                  if noCaseDays>=14:
                      endDay=day-14-4
                      break
              else:
                  noCaseDays=0
              day+=1
          length.append(endDay)
      pd.Series(length).plot(kind='hist',bins=61,\
              title='Histogram of Outbreak Duration (10000 Simulations)',figsize=(10,4))
      plt.xlabel('Length of outbreak (days)')
      plt.show()
```



(Optional) Additional Analysis

What is another interesting question that might be answered using the simulation model above? How would you go about doing the analysis? I encourage you to do the analysis in your own time for practice, but you don't have to hand in anything.

Information about the Midterm Exam

As described in the Syllabus, the midterm will be held during class time on Tuesday Oct. 5. It tests your mastery of skills taught in Weeks 1-5, which culminates in creating simulation models using Python and algorithmic thinking. It is administered on Blackboard, so both in-person and remote students can participate. There are three questions, worth a total of 24 points. See the two practice midterms posted online for the format. There are also nine extra practice problems that are posted in the folder for Week 6, and you can treat these as three additional midterms.

You have 80 minutes from the time you clicked "Begin" on Blackboard. After completing the Jupyter notebook, make sure to save it, then attach the .ipynb file to Blackboard and submit. Immediately after submitting, email your solution file to the professor in case Blackboard malfunctions. You are allowed to consult any resources posted online as well as any book or notes, but you CANNOT ask someone for help, or share your solutions with anyone other than the professor. Any violation of academic integrity will result in a failing grade for everyone involved.

*For each problem, you should place all your final code in one cell, with the comment "# Final Solution" on top. This cell should include all necessary imports and definitions, so that if one restarts the kernel and only runs that cell, the test code will work. Only the correctness of your final code matters: as long as you fulfill all the specifications described in the problem description, it doesn't matter how you solve the problem or how efficient is your code. However, if you cannot solve a problem, you may get partial credits for submitting whatever you have, including any parts of the four steps of algorithmic thinking.

We will go over the solutions of Practice Midterm A on Tue. 9/28 and Practice Midterm B on Thu. 9/30. Please attempt these practice midterms yourself before these sessions, as this will be best for your learning.