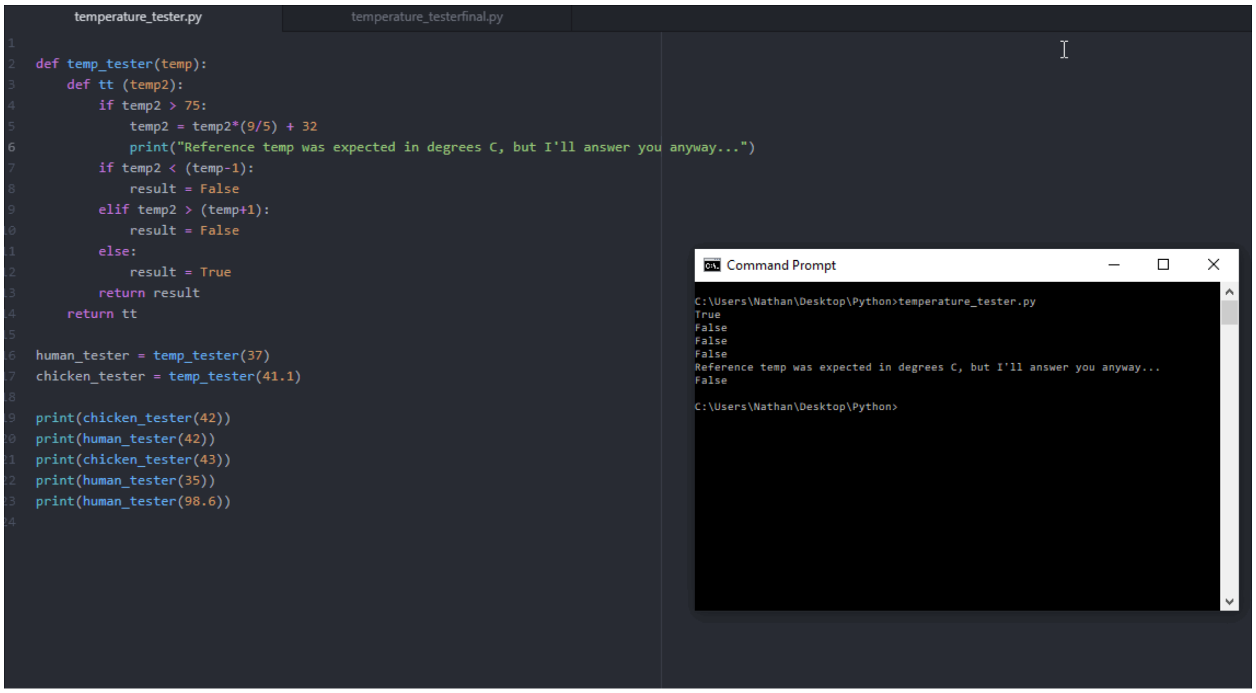
Nathan C. Walk, MD

9/16/20

BIS 634

Exercise 1:



temperature_tester.py

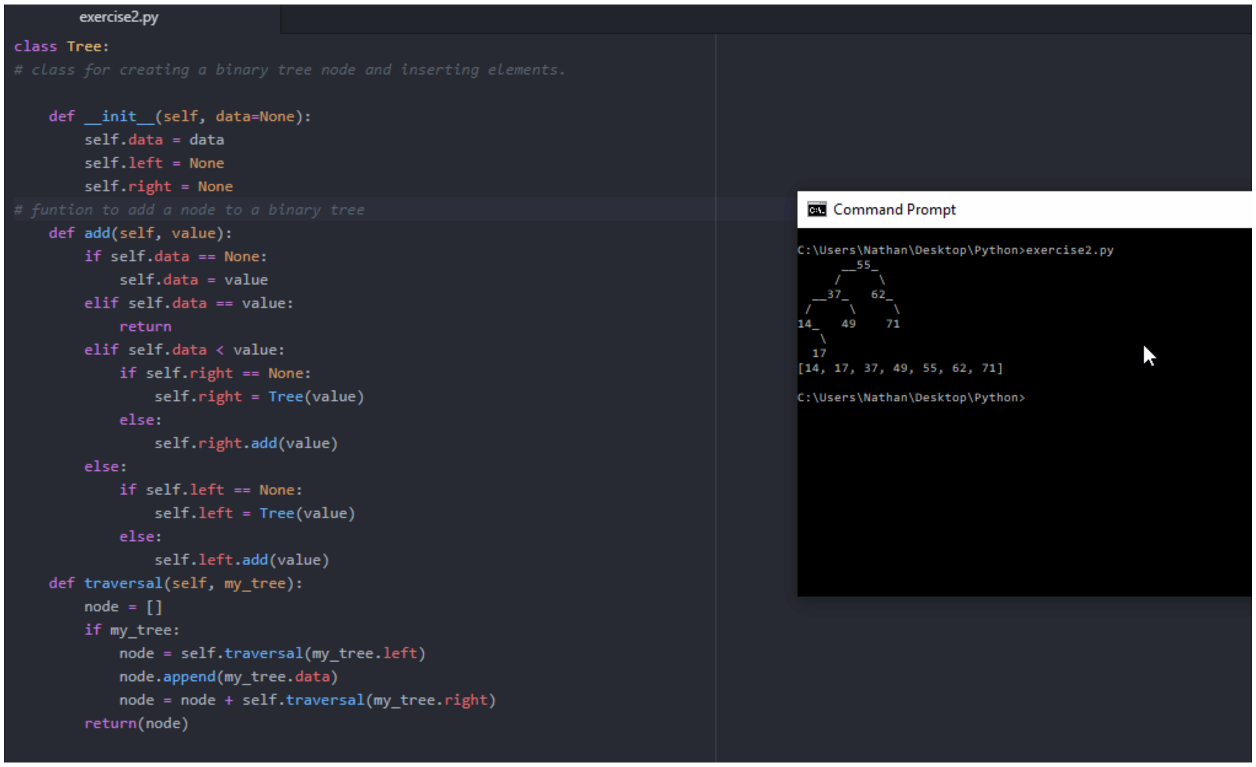
temperature_testerfinal.py

```python
def temp_tester(temp):
    def tt (temp2):
        if temp2 > 75:
            temp2 = temp2*(9/5) + 32
            print("Reference temp was expected in degrees C, but I'll answer you anyway...")
        if temp2 < (temp-1):
            result = False
        elif temp2 > (temp+1):
            result = False
        else:
            result = True
        return result
    return tt

human_tester = temp_tester(37)
chicken_tester = temp_tester(41.1)

print(chicken_tester(42))
print(human_tester(42))
print(chicken_tester(43))
print(human_tester(35))
print(human_tester(98.6))
```

Command Prompt

```
C:\Users\Nathan\Desktop\Python>temperature_tester.py
True
False
False
False
Reference temp was expected in degrees C, but I'll answer you anyway...
False

C:\Users\Nathan\Desktop\Python>
```

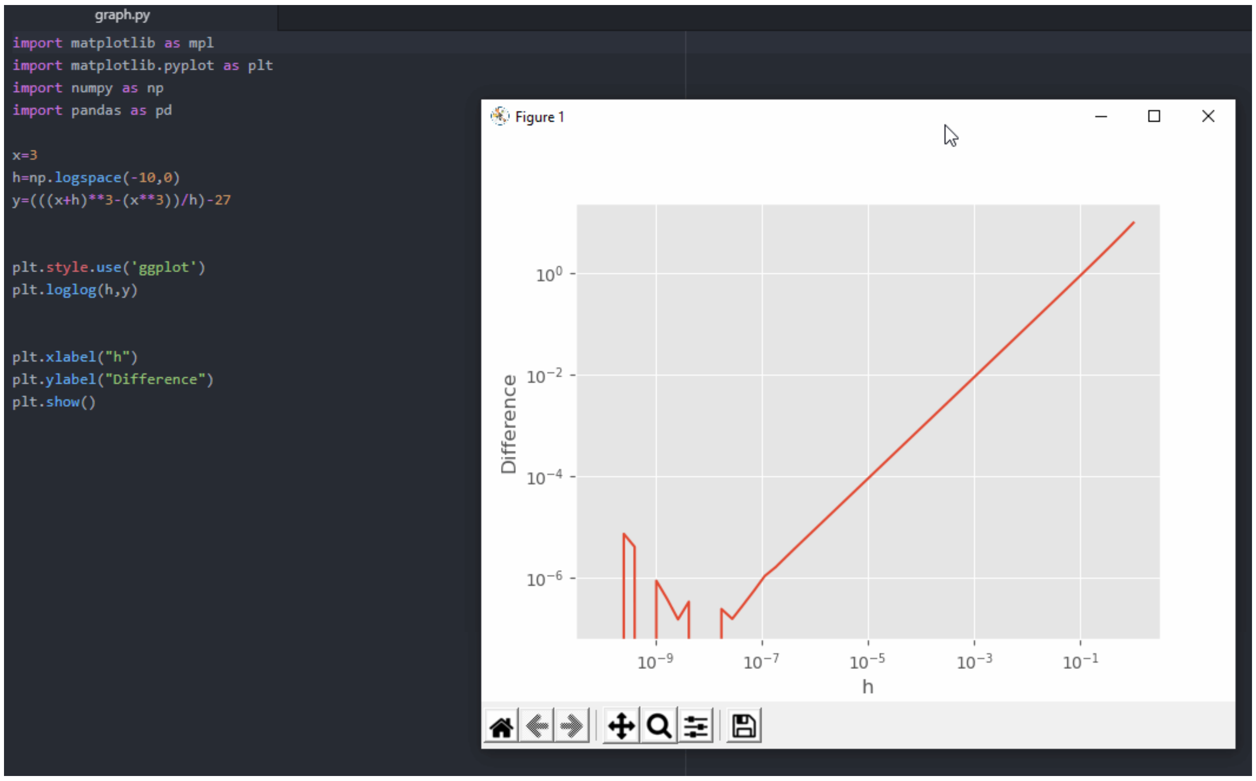
Exercise 2:



```python
class Tree:
# class for creating a binary tree node and inserting elements.

    def __init__(self, data=None):
        self.data = data
        self.left = None
        self.right = None
# funtion to add a node to a binary tree
    def add(self, value):
        if self.data == None:
            self.data = value
        elif self.data == value:
            return
        elif self.data < value:
            if self.right == None:
                self.right = Tree(value)
            else:
                self.right.add(value)
        else:
            if self.left == None:
                self.left = Tree(value)
            else:
                self.left.add(value)
    def traversal(self, my_tree):
        node = []
        if my_tree:
            node = self.traversal(my_tree.left)
            node.append(my_tree.data)
            node = node + self.traversal(my_tree.right)
        return(node)
```

```
C:\Users\Nathan\Desktop\Python>exercise2.py
      __55_
     /     \
  __37_   62_
 /     \     \
14_   49    71
   \
  17
[14, 17, 37, 49, 55, 62, 71]

C:\Users\Nathan\Desktop\Python>
```

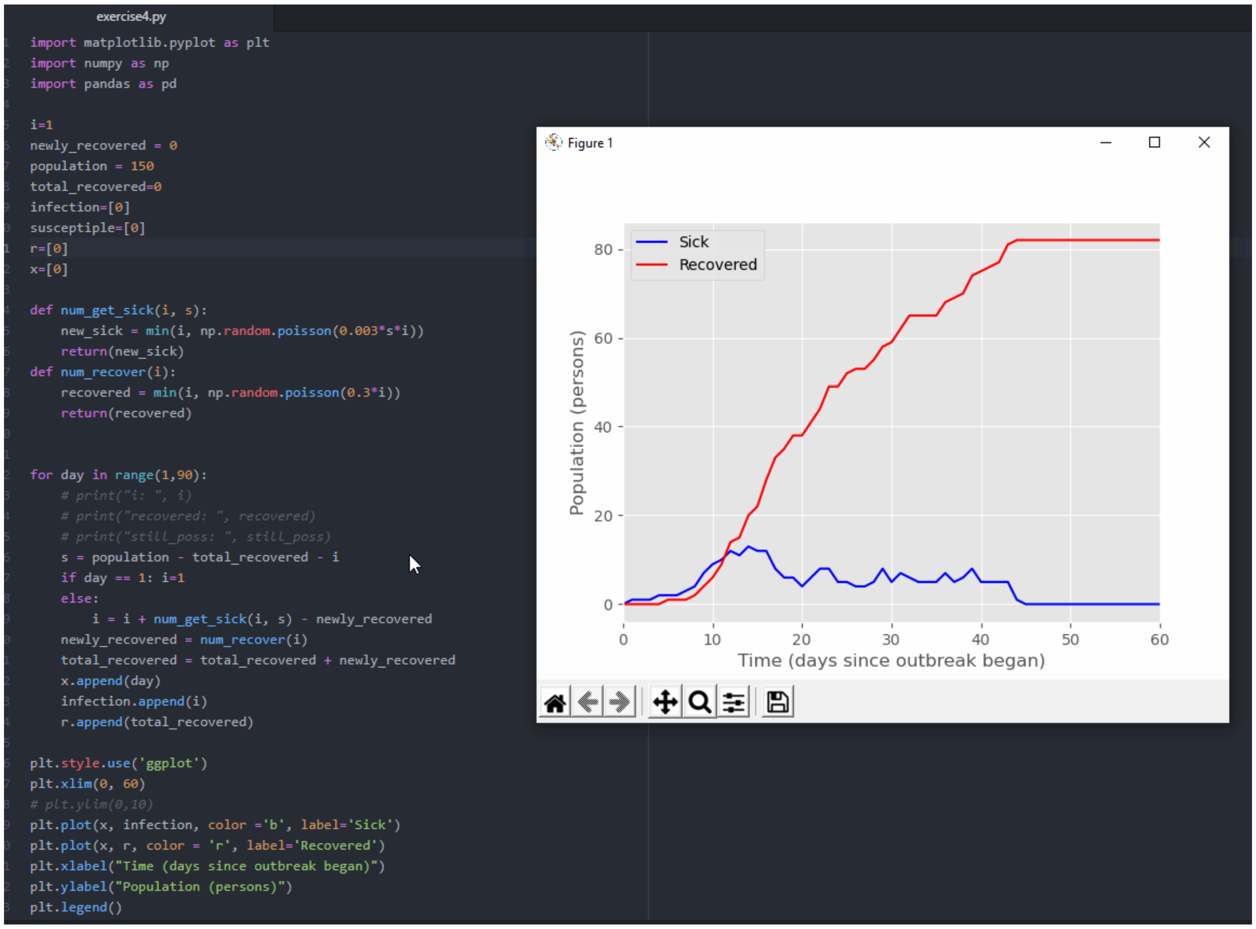
Exercise 3:



```
import matplotlib as mpl
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd

x=3
h=np.logspace(-10,0)
y=(((x+h)**3-(x**3))/h)-27


plt.style.use('ggplot')
plt.loglog(h,y)


plt.xlabel("h")
plt.ylabel("Difference")
plt.show()
```

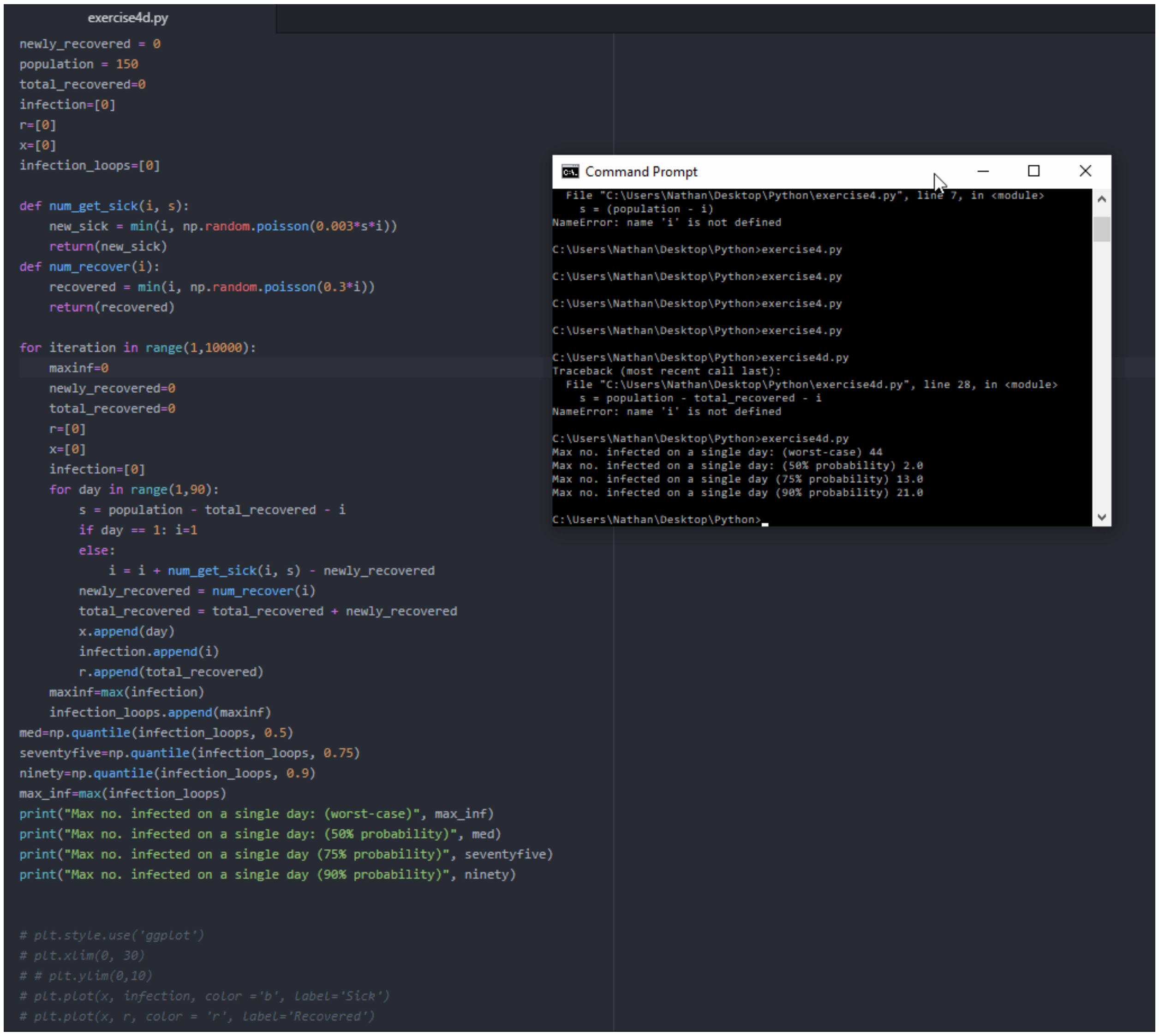
Exercise 4:

Single outbreak:



```python
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd

i=1
newly_recovered = 0
population = 150
total_recovered=0
infection=[0]
susceptiple=[0]
r=[0]
x=[0]

def num_get_sick(i, s):
    new_sick = min(i, np.random.poisson(0.003*s*i))
    return(new_sick)
def num_recover(i):
    recovered = min(i, np.random.poisson(0.3*i))
    return(recovered)


for day in range(1,90):
    # print("i: ", i)
    # print("recovered: ", recovered)
    # print("still_poss: ", still_poss)
    s = population - total_recovered - i
    if day == 1: i=1
    else:
        i = i + num_get_sick(i, s) - newly_recovered
    newly_recovered = num_recover(i)
    total_recovered = total_recovered + newly_recovered
    x.append(day)
    infection.append(i)
    r.append(total_recovered)

plt.style.use('ggplot')
plt.xlim(0, 60)
# plt.ylim(0,10)
plt.plot(x, infection, color ='b', label='Sick')
plt.plot(x, r, color = 'r', label='Recovered')
plt.xlabel("Time (days since outbreak began)")
plt.ylabel("Population (persons)")
plt.legend()
```

100,000 simulations – quantiles:


exercise4d.py

```python
newly_recovered = 0
population = 150
total_recovered=0
infection=[0]
r=[0]
x=[0]
infection_loops=[0]

def num_get_sick(i, s):
    new_sick = min(i, np.random.poisson(0.003*s*i))
    return(new_sick)
def num_recover(i):
    recovered = min(i, np.random.poisson(0.3*i))
    return(recovered)

for iteration in range(1,10000):
    maxinf=0
    newly_recovered=0
    total_recovered=0
    r=[0]
    x=[0]
    infection=[0]
    for day in range(1,90):
        s = population - total_recovered - i
        if day == 1: i=1
        else:
            i = i + num_get_sick(i, s) - newly_recovered
        newly_recovered = num_recover(i)
        total_recovered = total_recovered + newly_recovered
        x.append(day)
        infection.append(i)
        r.append(total_recovered)
    maxinf=max(infection)
    infection_loops.append(maxinf)
med=np.quantile(infection_loops, 0.5)
seventyfive=np.quantile(infection_loops, 0.75)
ninety=np.quantile(infection_loops, 0.9)
max_inf=max(infection_loops)
print("Max no. infected on a single day: (worst-case)", max_inf)
print("Max no. infected on a single day: (50% probability)", med)
print("Max no. infected on a single day (75% probability)", seventyfive)
print("Max no. infected on a single day (90% probability)", ninety)


# plt.style.use('ggplot')
# plt.xlim(0, 30)
# # plt.ylim(0,10)
# plt.plot(x, infection, color ='b', label='Sick')
# plt.plot(x, r, color = 'r', label='Recovered')
```

Command Prompt

```
  File "C:\Users\Nathan\Desktop\Python\exercise4.py", line 7, in <module>
    s = (population - i)
NameError: name 'i' is not defined

C:\Users\Nathan\Desktop\Python>exercise4.py

C:\Users\Nathan\Desktop\Python>exercise4.py

C:\Users\Nathan\Desktop\Python>exercise4.py

C:\Users\Nathan\Desktop\Python>exercise4.py

C:\Users\Nathan\Desktop\Python>exercise4d.py
Traceback (most recent call last):
  File "C:\Users\Nathan\Desktop\Python\exercise4d.py", line 28, in <module>
    s = population - total_recovered - i
NameError: name 'i' is not defined

C:\Users\Nathan\Desktop\Python>exercise4d.py
Max no. infected on a single day: (worst-case) 44
Max no. infected on a single day: (50% probability) 2.0
Max no. infected on a single day (75% probability) 13.0
Max no. infected on a single day (90% probability) 21.0

C:\Users\Nathan\Desktop\Python>
```