run_ant

July 5, 2022

[ ]:

```
from ant import Ant
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
from functools import lru_cache
```

[ ]:

```
def boundary_1(x, y):
    if (x**2 + y**2) <= 2:
        return True
    return False

def boundary_2(x, y):
    if ((x-2.5)/30)**2 + ((y-2.5)/40)**2 < 1:
        return True
    return False

def boundary_3(x,y):
    if ((x-0.25)/3)**2 + ((y-0.25)/4)**2 < 1:
        return True
    return False
```

1 Bill: A Clueless Hungry Ant

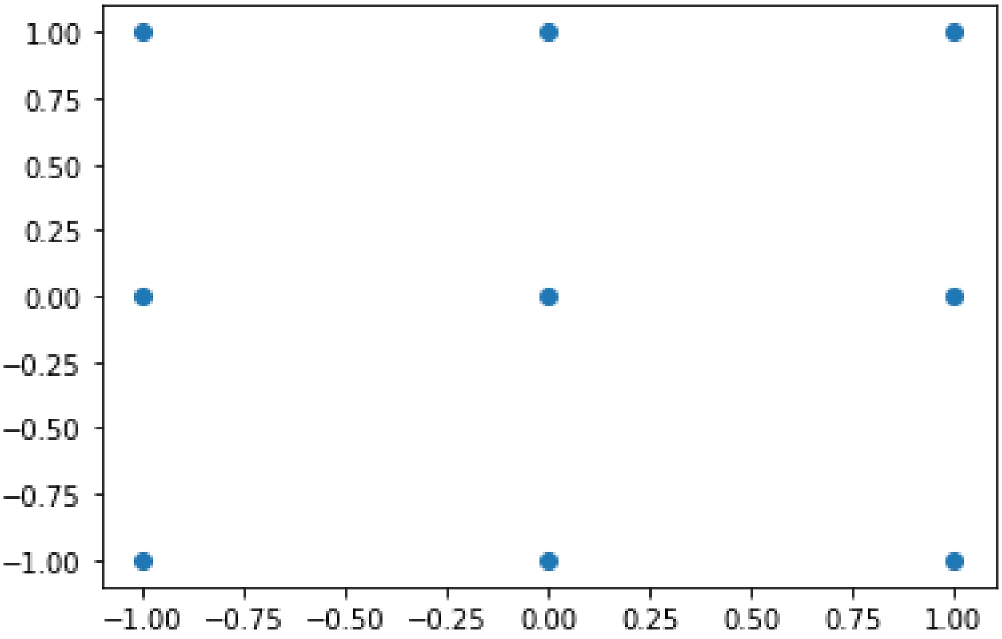
2 Question 1

2.1 Initialize Bill

[ ]:

```
bill = Ant(boundary_function=boundary_1, x0=0, y0=0, workers=8)
print('Below are the admissible transition points')
bill.show_transition_points()
```

```
Hi! My name is Bill, and I am a clueless ant is search for food!
In my search for food, I will start from (0, 0)
Please wait while I calculate all the transition points that I can visit (even I
should not be aware of them xD)
Ok, I am ready to go!
Below are the admissible transition points
```



2.2 Solutions

2.2.1 Markov Solution

[ ]:

```
bill.get_markov_solution()
```

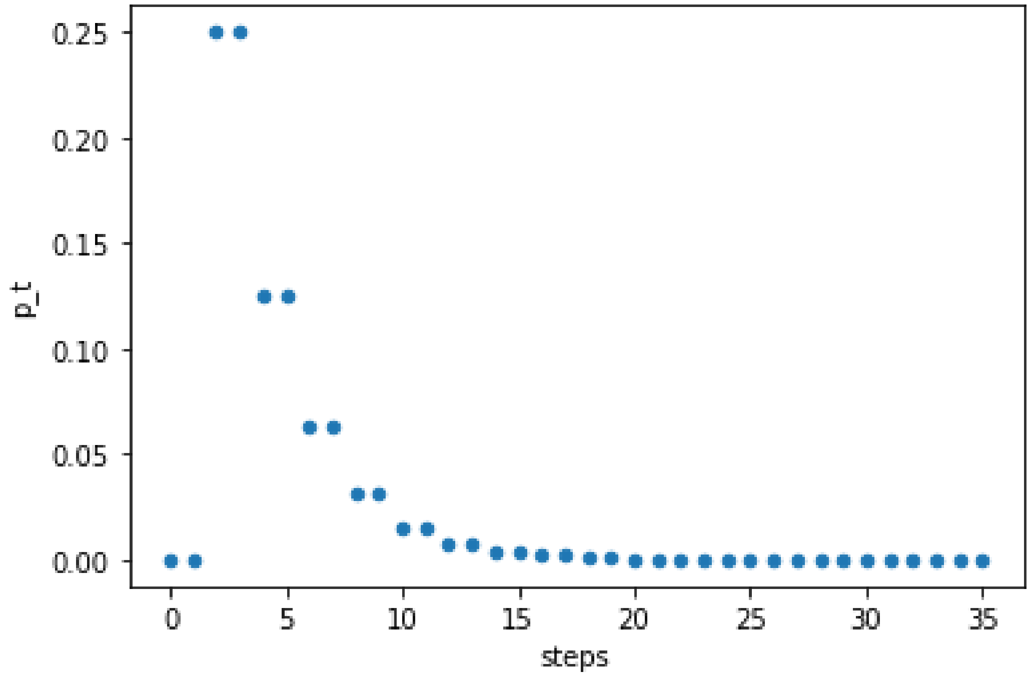
```
Bill has to make 4.5 steps on average to reach the food. Poor Bill!
```

2.2.2 Recursive Solution

[ ]:

```
avg_steps, data_recursive = bill.calculate_avg_steps_recursive(0.00001)
fig, ax = plt.subplots(1)
data_recursive.plot.scatter(x='steps', y='p_t', ax = ax)
plt.show()
```

```
Bill has to make 4.499706268310547 steps on average to reach the food. Poor
Bill!
```

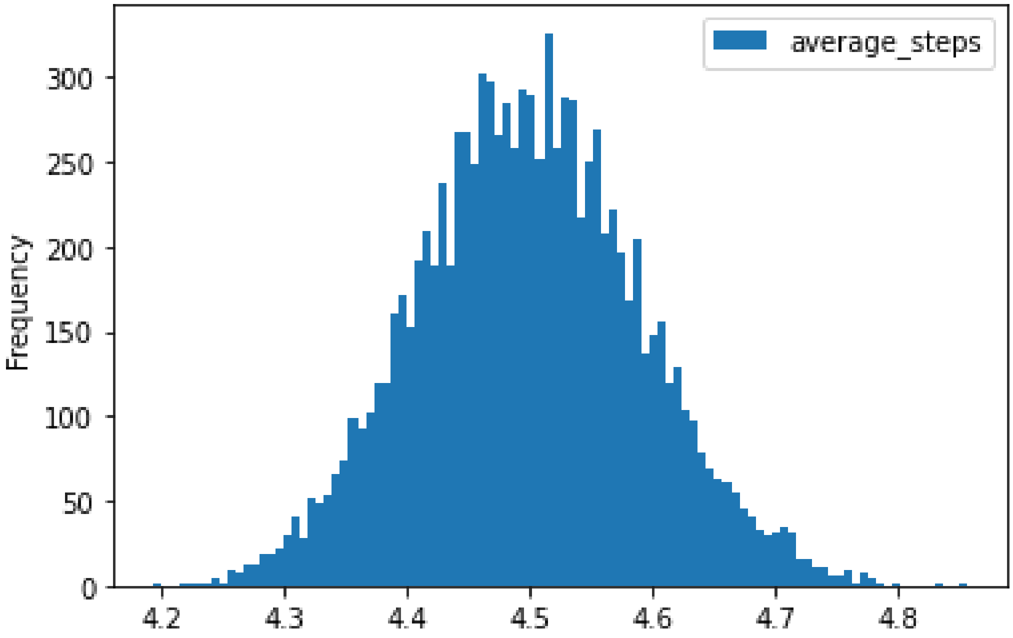


2.2.3 Monte Carlo Simulation

[ ]:

```
monte_carlo = bill.do_monte_carlo_avg_steps(n_samples=10000)
fig, ax = plt.subplots(1)
monte_carlo.plot.hist(bins = int(np.floor(np.sqrt(len(monte_carlo)))), column = ['average_steps'], ax = ax)
plt.show()
```

```
Bill has to make 4.4993782 steps on average to reach the food. Poor Bill!
```



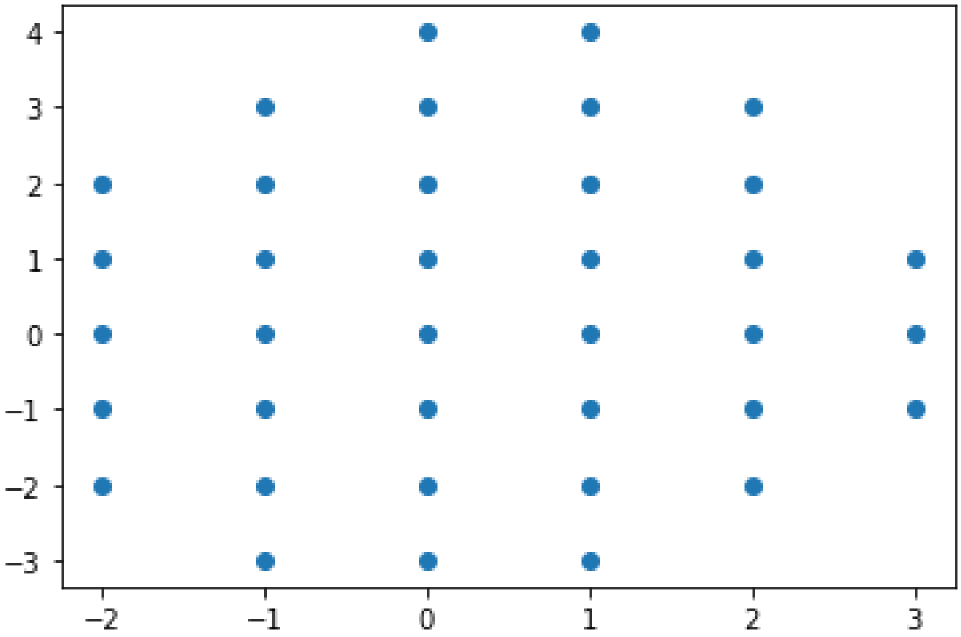
3 Question 3

3.1 Initialize Bill

[ ]:

```
bill = Ant(boundary_function=boundary_3, x0=0, y0=0, workers=8)
print('Below are the admissible transition points')
bill.show_transition_points()
```

```
Hi! My name is Bill, and I am a clueless ant is search for food!
In my search for food, I will start from (0, 0)
Please wait while I calculate all the transition points that I can visit (even I
should not be aware of them xD)
Ok, I am ready to go!
Below are the admissible transition points
```



3.2 Solutions

3.2.1 Markov Solution

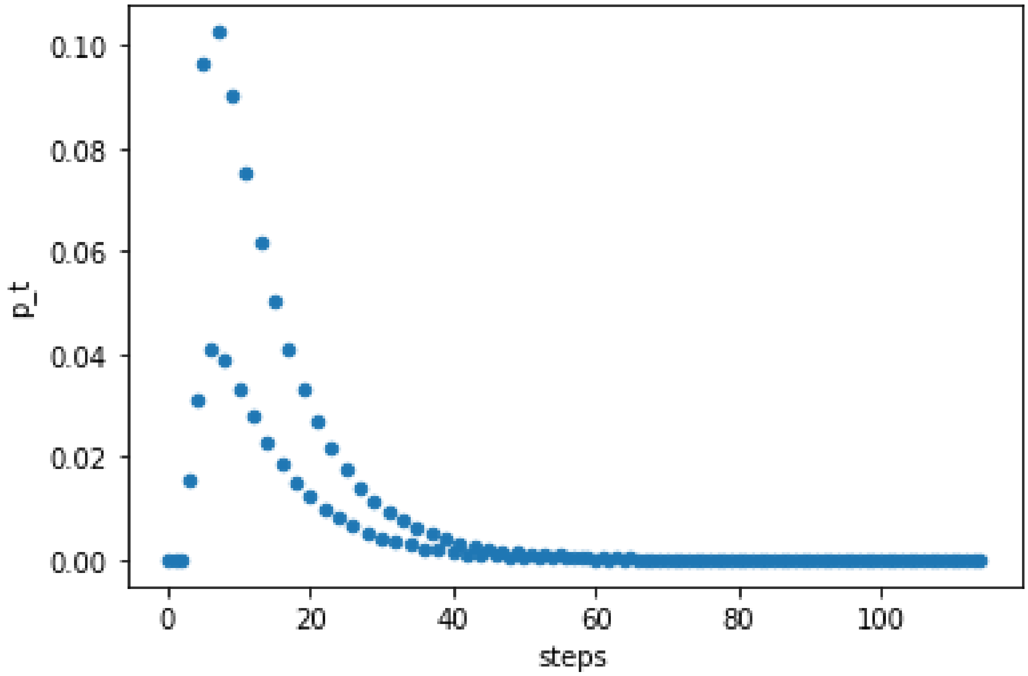
```
[ ]: bill.get_markov_solution()
```

```
Bill has to make 13.992053058411814 steps on average to reach the food. Poor
Bill!
```

3.2.2 Recursive Solution

```
[ ]: avg_steps, data_recursive = bill.calculate_avg_steps_recursive(0.00001)
     fig, ax = plt.subplots(1)
     data_recursive.plot.scatter(x='steps', y='p_t', ax = ax)
     plt.show()
```

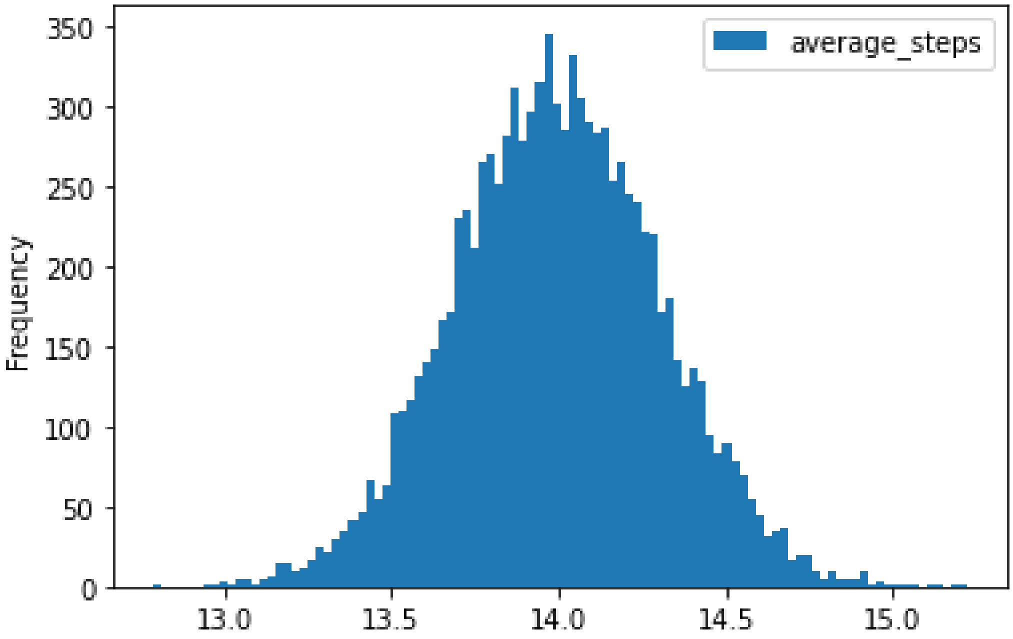
```
Bill has to make 13.990874270320726 steps on average to reach the food. Poor
Bill!
```



3.2.3 Monte Carlo Solution

[ ]:
```
monte_carlo = bill.do_monte_carlo_avg_steps(n_samples=10000)
fig, ax = plt.subplots(1)
monte_carlo.plot.hist(bins = int(np.floor(np.sqrt(len(monte_carlo)))), column =␣
  ↪['average_steps'], ax = ax)
plt.show()
```

```
Bill has to make 13.99642 steps on average to reach the food. Poor Bill!
```



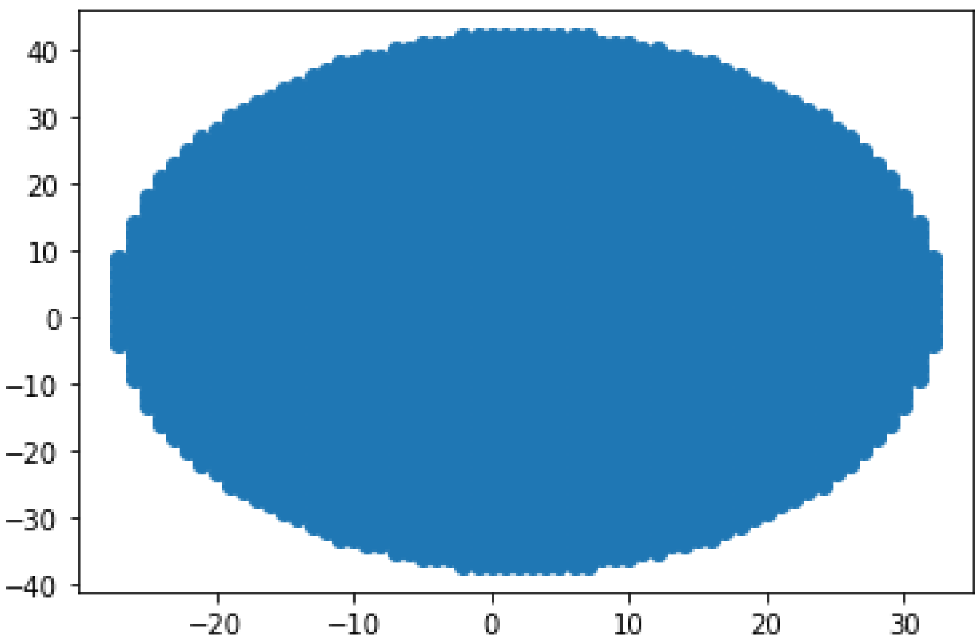
4 Bonus Section. An expensive scenario: large boundary

4.1 Initialize Bill

[ ]:

```
bill = Ant(boundary_function=boundary_2, workers=7)
print('Below are the admissible transition points')
bill.show_transition_points()
```

```
Hi! My name is Bill, and I am a clueless ant is search for food!
In my search for food, I will start from (0, 0)
Please wait while I calculate all the transition points that I can visit (even I
should not be aware of them xD)
Ok, I am ready to go!
Below are the admissible transition points
```



4.2 Solutions

4.2.1 Markov Solution

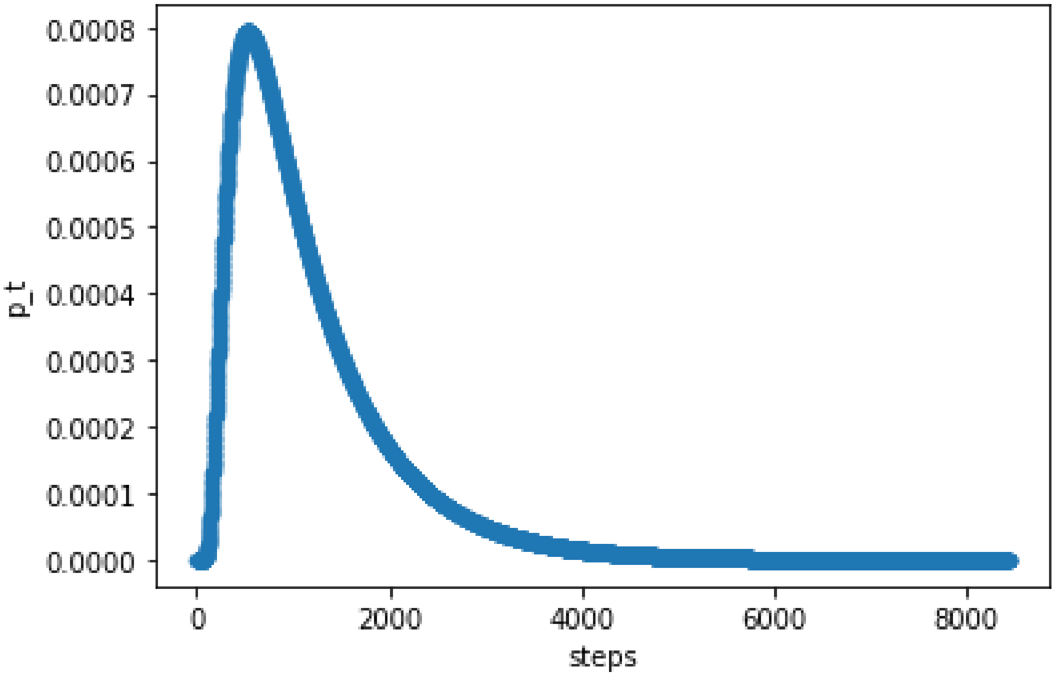
```
[ ]: bill.get_markov_solution()
```

```
Bill has to make 1163.726257998394 steps on average to reach the food. Poor
Bill!
```

4.2.2 Recursive Solution

```
[ ]: avg_steps, data_recursive = bill.calculate_avg_steps_recursive(0.00005)
     fig, ax = plt.subplots(1)
     data_recursive.plot.scatter(x='steps', y='p_t', ax = ax)
     plt.show()
```

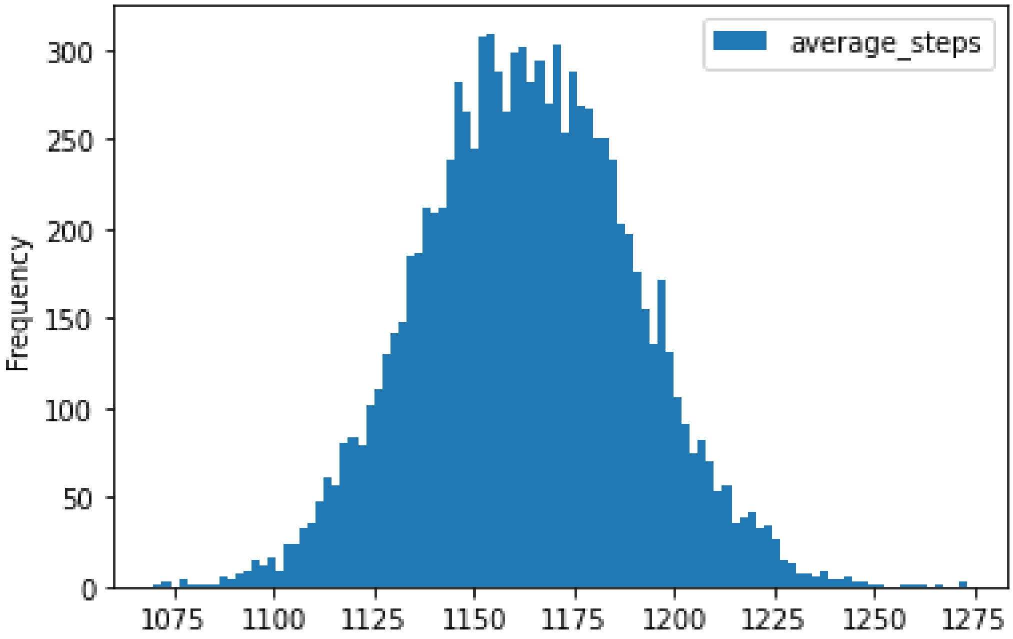
```
Bill has to make 1163.2636139069193 steps on average to reach the food. Poor
Bill!
```



4.2.3 Monte Carlo Simulation

```
[ ]: monte_carlo = bill.do_monte_carlo_avg_steps(n_samples=10000)
     fig, ax = plt.subplots(1)
     monte_carlo.plot.hist(bins = int(np.floor(np.sqrt(len(monte_carlo)))), column =␣
      ↪['average_steps'], ax = ax)
     plt.show()
```

```
Bill has to make 1163.5556565999998 steps on average to reach the food. Poor
Bill!
```


average_steps
Frequency
300
250
200
150
100
50
0
1075
1100
1125
1150
1175
1200
1225
1250
1275

```
[ ]:
```