

# PSTAT160A F21 Python HW 5

November 30, 2021

## 1 Python Homework 5

**Due date:** Tuesday, November 30, via GradeScope

**Instruction:** Please upload your .pdf file on Gauchospace with filename “PythonHW5\_YOURPERMNUMBER.pdf”.

## 2 Problem 1 (10 pt)

Consider the cat and snake problem from Assignment #6.

Simulate  $N \doteq 10^4$  realizations of the respective Markov Chain where Teddy the cat begins in room 1 and wanders around the maze until finding his pet snake. Each Markov chain trajectory should be simulated until Teddy reaches his pet snake (so you only have to run it until he reaches his pet snake).

Let us suppose that each of the areas contains cat treats. In particular, each time Teddy visits area  $k$ , he gets  $k$  cat treats. So for example every time (including at the very beginning) that Teddy is in area 1, he gets 1 cat treat. We want to compute the total number of cat treats Teddy will get on average before he reaches the snake.

Estimate the mean and variance of the total snacks the cat treats Teddy will get based on the above  $N$  simulated trajectories.

**Note:** an exact answer to this question is also possible using first-step analysis in case you really wanted to double-check yourself.

As usual, we start with loading some packages:

```
[1]: import numpy as np
      from numpy import linalg
      import scipy.linalg
```

```
[2]: # WRITE YOUR OWN CODE HERE!
      # You need to type in the transition matrix and the function to simulate_
      ↪trajectories of Teddy wondering across the areas
      P = np.array([[ 0,   1/2,   0,   1/2,   0,   0,   0,   0,   0],
                    [1/3,   0,   1/3,   0,   1/3,   0,   0,   0,   0],
                    [ 0,   1/2,   0,   0,   0,   1/2,   0,   0,   0],
                    [1/3,   0,   0,   0,   1/3,   0,   1/3,   0,   0],
```

```

[ 0, 1/4, 0, 1/4, 0, 1/4, 0, 1/4, 0],
[ 0, 0, 1/3, 0, 1/3, 0, 0, 0, 1/3],
[ 0, 0, 0, 1/2, 0, 0, 0, 1/2, 0],
[ 0, 0, 0, 0, 1/3, 0, 1/3, 0, 1/3],
[ 0, 0, 0, 0, 0, 1/2, 0, 1/2, 0]]

```

```

[8]: initState = 1
N = 10**4
totalTreats = []
# WRITE YOUR OWN CODE HERE to figure out the tT (total Treats) on that
↳ trajectory!
for i in range(N):
    P_0 = P
    current_state = initState

    k = np.random.choice(9, 1, p = P[0]) # area

    while k != 8:
        k = np.random.choice(9, 1, p=P[k[0]])
        current_state += (k+1)
        P_0 = np.matmul(P_0,P)

    totalTreats.append(current_state)

```

Finally estimate the mean and variance of totalTreats

```

[9]: mean = np.mean(totalTreats)
var = np.var(totalTreats)
print("the mean is ",mean,)
print("the mean is ",var,)

```

```

the mean is 78.0978
the mean is 3579.4440351599997

```

### 3 Problem 2 (10 pt)

You start with five dice. Roll all the dice and put aside those dice that land on 6. Then, roll the remaining dice, putting aside those dice that land on 6. And so on, until no dice are left.

Using  $10^5$  experiments, estimate the probability that it will take *more than 10 rounds* to end this game.

**Hint:** you can work directly with rolling the dice (via `dieRoll = 1 + np.random.choice(a = range(6))`) and tracking which ones remain, or you can setup a Markov Chain ( $X_n$ ) for the number of dice remaining and work with the respective transition matrix. If you do the latter, be sure to clearly type out  $P$  as part of your solution. Note that Markov Chain approach gets a bit cumbersome if you increase the number of dice to roll.

[32]: *# Write you own code here*

```
N = 10**5
results = []

for i in range(N):
    num_Rolls = 0
    num_Dice = 5

    while num_Dice > 0:
        num_Rolls += 1
        roll = [(1+ np.random.choice(a = range(6))) for dice in range(num_Dice)]
        num_Dice = num_Dice - roll.count(6)
    results += [num_Rolls]

filtered = []

for result in results:
    if result > 10:
        filtered.append(result)

p = len(filtered)/N
print("The probability is",p)
```

The probability is 0.58395

[ ]: