Bad Prime Dice Game

Ryan Mouw

February 12, 2021

1 The Game

To play this game you need a single six sided die. Here is how to play the game;

1. Your start with a score of zero
2. Roll the die to determine your starting score (1-6)
3. After your starting score is found, roll the die again
4. If your score plus the value of the die is prime and less than 26, subtract the value of the die from your score. If this subtraction would bring your score to zero or below zero, your score is now zero. (Go back to number 1)
5. If your score plus the value on the die is not prime, then this is your new score.
6. Continue rolling in this manner until your score plus die throw is greater than or equal to 26 (If your score plus your roll is greater than 26, it does not matter if their sum is prime).
7. Your score is now 26. You win!
8. Count the number of rolls it took to reach 26

2 Markov Chain and Transition Matrix

To model this game, we will use a Markov Chain. Markov chains are useful in modeling processes that change discretely among a set of states. **We can model our game with a Markov chain with 27 states (0-26). Each state represents a possible score in our game.** State 26 is called an absorbing state. More on this later.

First we will create what is called a "transition matrix." The rows of our matrix will represent the current state (i.e. our current score (0-26)) and the columns of our matrix will represent our score after the next die throw. Because our score can be a value from 0-26 (27 different values) we will need a 27x27 matrix.

We will refer to our transition matrix as M. The the value of M_{ij} entry represents the probably that the next die roll will take me from a score of i to a score of j. M tells us the probability of transitioning from a score of i to a score of j.

Note (Absorbing State): We say that a score of 26 is **absorbing** because once we reach a score of 26 we never leave the state of 26. If i is an absorbing state, $M_{ii} = 1$.

There are two constraints keeping a score of j from following any particular score of i.

The first constraint is that the die has only 6 values (1-6). So any score of j can be at most 6 away from any score i.

Secondly, if $i + value\ of\ die\ throw = j$ is prime, then the probability of transitioning to j in one step is zero, and instead transitioning to $i - value\ of\ die$ has a 1/6 chance.

Note: This does not mean that a prime j cannot be your score. For example, if $i = 0$ then the next roll is now the score, regardless if it is prime. Or perhaps $i = 18$ and a 1 is rolled, meaning the new score is $i - 1 = 17$ which is prime.

Transition matrix M represents our dice game, and is shown below.

$$\begin{pmatrix}
0 & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{2}{3} & 0 & 0 & 0 & \frac{1}{6} & 0 & \frac{1}{6} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{1}{3} & \frac{1}{6} & 0 & 0 & \frac{1}{6} & 0 & \frac{1}{6} & 0 & \frac{1}{6} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{1}{6} & \frac{1}{6} & 0 & 0 & \frac{1}{6} & 0 & \frac{1}{6} & 0 & \frac{1}{6} & \frac{1}{6} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{1}{6} & 0 & \frac{1}{6} & 0 & 0 & \frac{1}{6} & 0 & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{1}{6} & 0 & 0 & \frac{1}{6} & 0 & 0 & \frac{1}{6} & 0 & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{1}{6} & 0 & 0 & 0 & \frac{1}{6} & 0 & 0 & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & 0 & \frac{1}{6} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{1}{6} & 0 & \frac{1}{6} & 0 & 0 & 0 & 0 & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & 0 & \frac{1}{6} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \frac{1}{6} & 0 & \frac{1}{6} & 0 & 0 & 0 & \frac{1}{6} & \frac{1}{6} & 0 & \frac{1}{6} & 0 & \frac{1}{6} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & \frac{1}{6} & 0 & \frac{1}{6} & 0 & 0 & \frac{1}{6} & 0 & \frac{1}{6} & 0 & \frac{1}{6} & \frac{1}{6} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{6} & 0 & \frac{1}{6} & 0 & 0 & \frac{1}{6} & 0 & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & \frac{1}{6} & 0 & 0 & 0 & \frac{1}{6} & 0 & 0 & \frac{1}{6} & 0 & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{6} & 0 & 0 & 0 & \frac{1}{6} & 0 & 0 & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & 0 & \frac{1}{6} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{6} & 0 & \frac{1}{6} & 0 & 0 & 0 & 0 & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & 0 & \frac{1}{6} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{6} & 0 & \frac{1}{6} & 0 & 0 & 0 & \frac{1}{6} & \frac{1}{6} & 0 & \frac{1}{6} & 0 & \frac{1}{6} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{6} & 0 & \frac{1}{6} & 0 & 0 & \frac{1}{6} & 0 & \frac{1}{6} & 0 & \frac{1}{6} & \frac{1}{6} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{6} & 0 & \frac{1}{6} & 0 & 0 & \frac{1}{6} & 0 & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{6} & 0 & 0 & 0 & \frac{1}{6} & 0 & 0 & \frac{1}{6} & 0 & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{6} & 0 & 0 & 0 & \frac{1}{6} & 0 & 0 & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & 0 & \frac{1}{6} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{6} & 0 & 0 & 0 & 0 & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & 0 & \frac{1}{6} & \frac{1}{6} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{6} & 0 & 0 & 0 & \frac{1}{6} & \frac{1}{6} & 0 & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{6} & 0 & 0 & \frac{1}{6} & 0 & \frac{1}{6} & \frac{1}{6} & \frac{1}{3} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{6} & 0 & 0 & \frac{1}{6} & \frac{1}{6} & \frac{1}{2} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{6} & \frac{1}{6} & \frac{2}{3} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{6} & \frac{5}{6} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{pmatrix} \quad [M]$$

A powerful feature of our transition matrix M, is that M^x gives the set of probabilities of being in state j, x steps after being in state i.

For example, to determine the state of the game after 7 throws of the die, we take M^7. Now each M_{ij} entry tells us the probability that we will have a score j after x throws if we had started with a score of i.

3 The N Matrix

If our transition matrix has an absorbing state, then we can determine even more things about our game. To do this, we will define a new matrix, N.

First we require that our transition matrix M is in canonical form. Canonical form requires that M have all absorbing states represented in the last row(s) of the transition matrix.

Let $N = (I - Q)^{-1}$

Where Q is our original matrix M, less the last row and last column(because in our case there is only one absorbing state) and I is the corresponding identity matrix.

The sum of row i of the matrix N is the mean number of steps it takes to get from a score of i to our absorbing state (26 in our game). Below is a table that shows the expected number of throws for each winning score. Also included is a simulated mean, based off of 100,000 simulated games for each winning score.

Win score	Expected (N Matrix)	Simulated
10	54883/12942(4.24)	4.22
26	$\frac{1888114326143690885}{124917931460431158}$(15.11)	15.01
50	$\frac{3740404172903390758913990434245821153}{129898119682846339661951903730902225}$(28.79)	28.92
100	$\frac{351169668874734134845469087510840763876241206150803104863890036633628 6973}{66135556757612096211808471329282084628077234548110931971058262821842290}$(53.10)	53.31
150	$\frac{125919544176482389588233990951282549063624480555345371527994533716278666590455251617816555682927727822260680}{1605096034187367966009151865024882755048973183197827667994389754076826245146468385271591981323421119350511}$(78.45)	78.77
200	$\frac{5896732493126348758923263217318206803596443010907083765368808947067678407774519175011538494850317948358434116159013764915536582475775493030469}{564524389718675936195190018414047053943383729955913762186023133582529520281778397529459330037127548223063229752703296334531181494878908401 47}$(104.45)	104.62

As shown in the table, our simulated mean roll number is very close to the value given from the N matrix.

Now that we know the mean number of throws needed to reach our absorbing state, we can use the M matrix to answer other questions as well, such as;

What is the probability that at least half the expected number of rolls are needed to get to 26. To say it differently; what is the probability of having to roll a minimum of 8 times to win the game?

To answer this question, we ask it's complement. What is the probability that we will reach a score of 26 in 7 or fewer rolls? The last entry in the first row of M^7 is $\frac{29839}{279936}$ or roughly .1066. This means that if we start at zero, after 7 throws of the die we have a roughly 10.66% chance of being in our absorbing state of 26. So to determine the probability that at least half (8) the expected rolls are needed to get to 26, we take the complement of 10.66%, which is 89.34%.

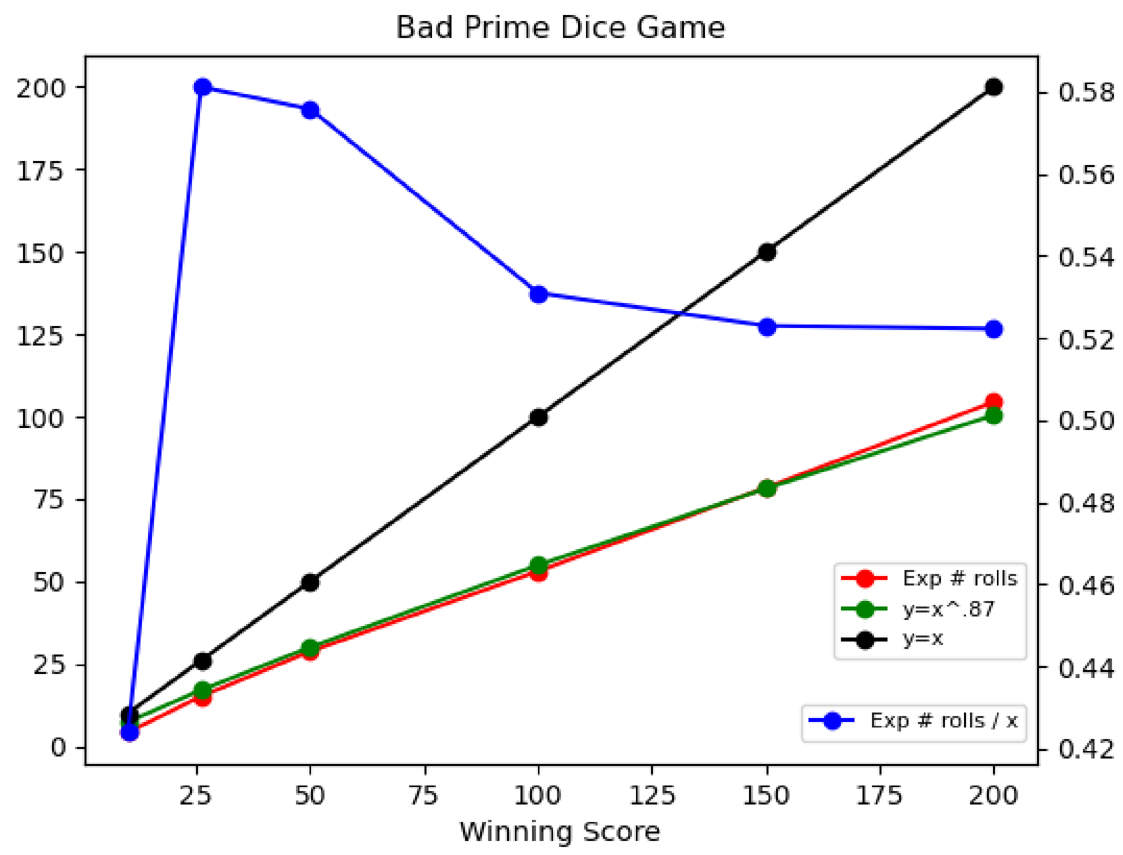
4 Graphs, Tables, and $y = x^a$

The expected number of rolls (red line) seemed to grow at a decreasing rate when compared to the line $y = x$. This was evidence that the equation governing the expected number of rolls given a winning score is of the form $y = x^a$. This would imply that $\frac{log y}{log x} = a$. Below is a table showing the values of $\frac{log y}{log x}$.

Winning Score	Expected (N Matrix)	$\frac{\log y}{\log x}$
10	4.24	.627
26	15.11	.833
50	28.79	.859
100	53.10	.863
150	78.45	.871
200	104.45	.877

a is growing as the winning score is increased. To plot this, we picked a value of a from our table. In this case we chose $a = .87$. Below is a graph showing the different plots.

Note: The blue line uses the right y-axis. All other lines use the left y-axis



As we can see $a = .87$ overestimates the first few expected values of y, and underestimates y values for $x > 150$, leading me to believe that an estimate of $a = .87$ would likely not do well outside of our values of x. The difference between values of a as x increases seems to be getting smaller, suggesting that a limit may exist.

5 Code

The below code was run in Sage using Jupyter Notebooks. Sage allows the option to force the use of rational values for all entries in our matrix. This creates very large fractional entries but ensures we are computing exact values.

Matplotlib was also used to create the graphs.

Comments are in the code to describe what is being done.

```
    ##Transition matrix


import numpy as np
import scipy
import matplotlib.pyplot as plt
%matplotlib notebook

#set matrix size (winning score +1)
n=27
#Create sage matrix that forces rational entries (QQ)
A=zero_matrix(QQ,n);

#Create transition matrix
for i in range(6):
    A[0,i+1]=1/6
A[n-1,n-1]=1;
for i in range(1,n-1):
    score = i
    for j in range(6):
        roll=j+1
        if score + roll in list(primes(1,n)):
            if score-roll <= 0:
                A[i,0]+=1/6
            else:
                A[i,score-roll]+=1/6
        else:
            if score + roll >= n:
                A[i,n-1]+=1/6
            else:
                A[i,score+roll]+=1/6



#find prob of being in absorb state after 7 rolls
B=A^7
print(B[0,-1])
print(CC(B[0,-1]))
print(1-B[0,-1])
CC(1-B[0,-1])

#Sum all rows to test that matrix is indeed Markov chain
sum_1=0
for i in range(n):
    for j in range(n):
        sum_1 += A[i,j]
sum_1

#define Q
Q=A[0:-1,0:-1]

#define identity matrix
I = matrix.identity(n-1)

#define N
N=(I-Q)
N=N.inverse()

#sum first row of N to determine mean rolls to winning score
sum_2=0
for j in range(n-1):
    sum_2 += N[0,j]
print(sum_2)
print(CC(sum_2))


import random

#Simulate games 100000 times
games=100000 # number of games simulated
means = []
for i in range(games):
    score = 0
    roll=0
    while score < n-1:
        roll+=1
        die_value = random.randint(1,6)
        if score == 0:
            score = die_value
        elif score + die_value in list(primes(1,n-1)):
            if score - die_value <=0:
                score = 0
            else:
                score = score-die_value
        else:
            score = score + die_value
    means.append(roll)

#find the mean score of the 100000 games
CC(mean(means))

#Plot results using Matplotlib

x=np.array([10,26,50,100,150,200])
expected=np.array([4.24,15.11,28.79,53.10,78.45,104.45])

fig, ax =plt.subplots()

ax.plot(x,expected, ’ro-’, label = "Exp # rolls")
ax.plot(x,(expected)^(1.12), ’go-’, label="(Exp # rolls)^1.13125")
ax.plot(x,x, ’ko-’, label="y=x")

ax2=ax.twinx()
ax2.plot(x, (expected/x), ’bo-’, label= "Exp # rolls / x")

ax.set_xlabel(’Winning Score’, fontsize = 10.5)
ax.set_title(’Bad Prime Dice Game’, fontsize = 11.3)
ax.legend(bbox_to_anchor = (1,.25), prop = {’size’:8})
ax2.legend(bbox_to_anchor = (1,.1), prop = {’size’:8})
```