Stochastic Thinking and Random Walks, Segment 2

Implementing a Random Process

```
import random

def rollDie():
    """returns a random int between 1 and 6"""
    return random.choice([1,2,3,4,5,6])

def testRoll(n = 10):
    result = ''
    for i in range(n):
        result = result + str(rollDie())
    print(result)
```

Probability of Various Results

- Consider `testRoll(5)`
- Which of the following outputs would surprise you?

```
11111
54424
```

- What is the probability of each?

Probability Is About Counting

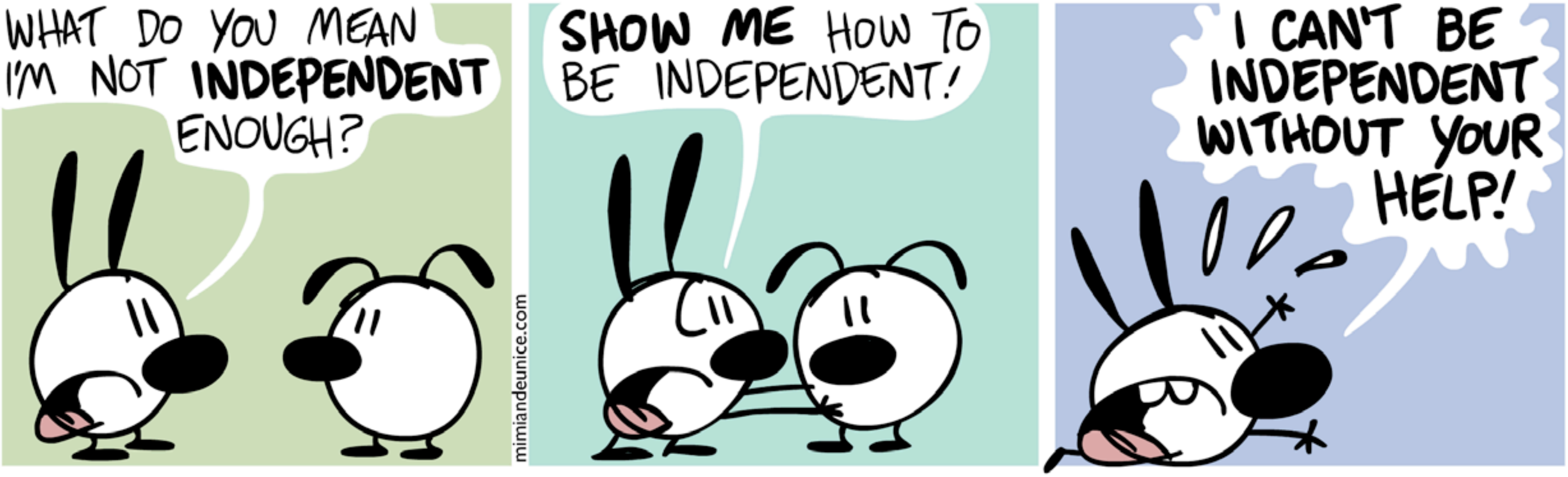
- Count the number of possible events
- Count the number of events that have the property of interest
- Divide one by the other
- Probability of 11111?
    - 11111, 11112, 11113, …, 11121, 11122, …, 66666
    - 1/(6**5)
    - ~0.0001286
- Probability of 54425?

Three Basic Facts About Probability

- Probabilities are always in the range 0 to 1. 0 if impossible, and 1 if guaranteed.
- If the probability of an event occurring is p, the probability of it not occurring must be 1-p.
- When events are independent of each other, the probability of all of the events occurring is equal to a product of the probabilities of each of the events occurring.

Independence

- Two events are independent if the outcome of one event has no influence on the outcome of the other.



Will One of Real Madrid or Barça Lose?

- Both good teams
- Assume that both are playing
- Assume each wins, on average, 7 out of 8 games
- Probability of both winning is 7/8 * 7/8 = 49/64
- Probability of at least one losing is 1 – 49/64 = 15/64
- But suppose they are playing each other?
  - Outcomes are not independent
  - Probability of one of them losing is much higher than 15/64!

Will One of Real Madrid or Barça Lose?

- Both good teams
- Assume that both are playing
- Assume each wins, on average, 7 out of 8 games
- Probability of both winning is 7/8 * 7/8 = 49/64
- Probability of one losing is 1 – 49/64 = 15/64
- But suppose they are playing each other?
  - Outcomes are not independent
  - Probability of one of them losing is much higher than 15/64

A Simulation

```
def runSim(goal, numTrials):
    total = 0
    for i in range(numTrials):
        result = ''
        for j in range(len(goal)):
            result += str(rollDie())
        if result == goal:
            total += 1
    print('Actual probability =',
          round(1/(6**len(goal)), 8))
    estProbability = round(total/numTrials, 8)
    print('Estimated Probability  =',
          round(estProbability, 8))

runSim('11111', 1000)
```

Output of Simulation

- Actual probability = 0.0001286
- Estimated Probability = 0.0
- Actual probability = 0.0001286
- Estimated Probability = 0.0

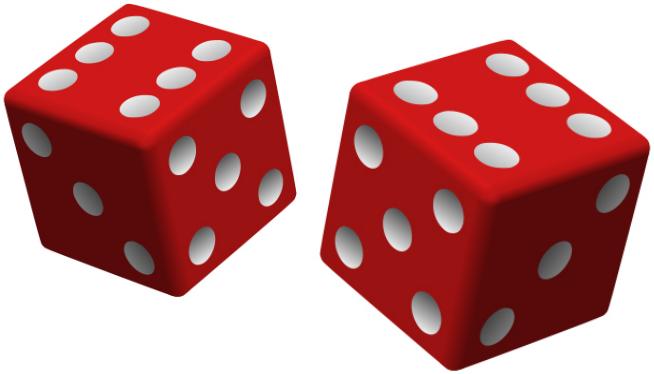
- How did I know that this is what would get printed?
- Why did simulation give me the wrong answer?

Let's try 1,000,000 trials

How Common Are Boxcars?

How Common Are Boxcars?

- 6^2 possible combinations of two die
  - One 1 with two 6's
  - Hence probability is 1/36
- Another way of computing it
  - Probability of rolling 6 with one die = 1/6
  - Probability of rolling 6 with other die = 1/6
  - Since these events are independent, probability of rolling a 6 with both die = $1/6 * 1/6 = 1/36 \cong 0.02778$



Approximating Using a Simulation

```
def fracBoxCars(numTests):
    numBoxCars = 0.0
    for i in range(numTests):
        if rollDie() == 6 and rollDie() == 6:
            numBoxCars += 1
    return numBoxCars/numTests

print('Frequency of double 6 =',
      str(fracBoxCars(100000)*100) + '%')
```

Morals

- Moral 1: It takes a lot of trials to get a good estimate of the frequency of occurrence of a rare event. We'll talk lots more in later lectures about how to know when we have enough trials.
- Moral 2: One should not confuse the sample probability with the actual probability
- Moral 3: There was really no need to do this by simulation, since there is a perfectly good closed form answer. We will see many examples where this is not true.
- But simulations are often useful, as we will see