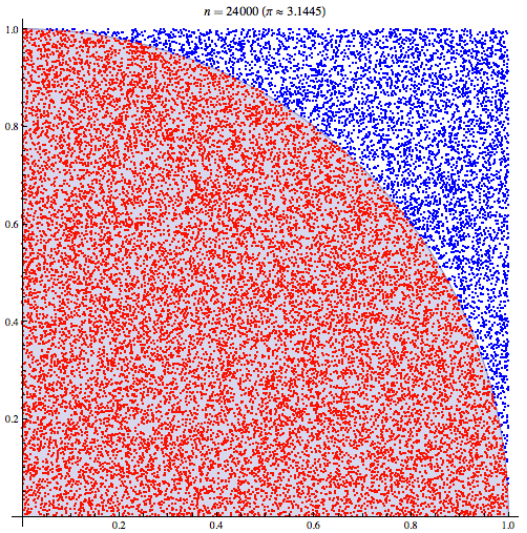

Monte Carlo Method
SNOLab Workshop
Brian Mong

What is the Monte Carlo Method?

- Simulation of an event, repeated over and over, to determine the probability of a particular outcome.
  - Simulate N events
  - Record outcome of each event
  - Probability density function (PDF) is outcome/N
- Generally accuracy increases with N (statistical)

Example - Compute Pi (python)

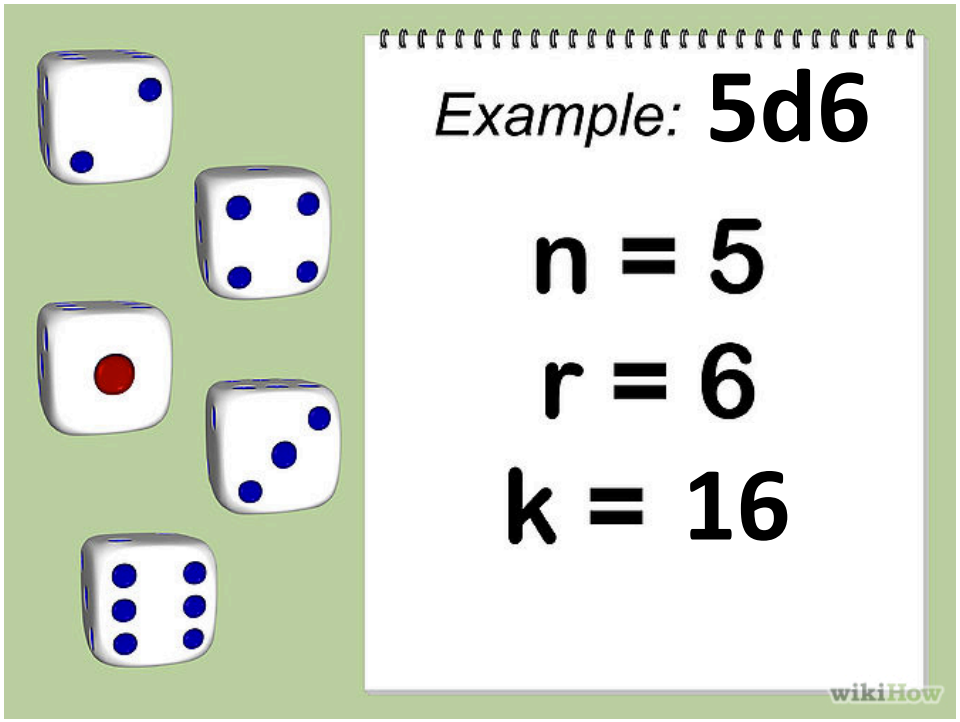



```
# We know A = Pi*r^2
# Randomly select x&y from [0,1]
# Check if point is in circle quad
import numpy as np
def compute_pi(N):
    success = 0
    for i in range(N):
        r = np.sum(np.random.rand(2)**2)**.5
        if r<1: success+=1
    return success/float(N)*4
```

Slow, but explicit!

```
compute_pi(10000) 3.1488
compute_pi(1000000) 3.1419
```

Doughnuts & Dice (example)



Analytically:

1. Write out each permutation and its sum (k) (~8000)
2. Calculate probability of rolling each permutation using binomial theorem
3. Sum probabilities with the same k

--or—

Write 5 lines of (python) code to simulate it

Try it yourself, or dice.py

http://www.wikihow.com/Calculate-Multiple-Dice-Probabilities

Find the PDF for 3d8, 4d6, 8d3

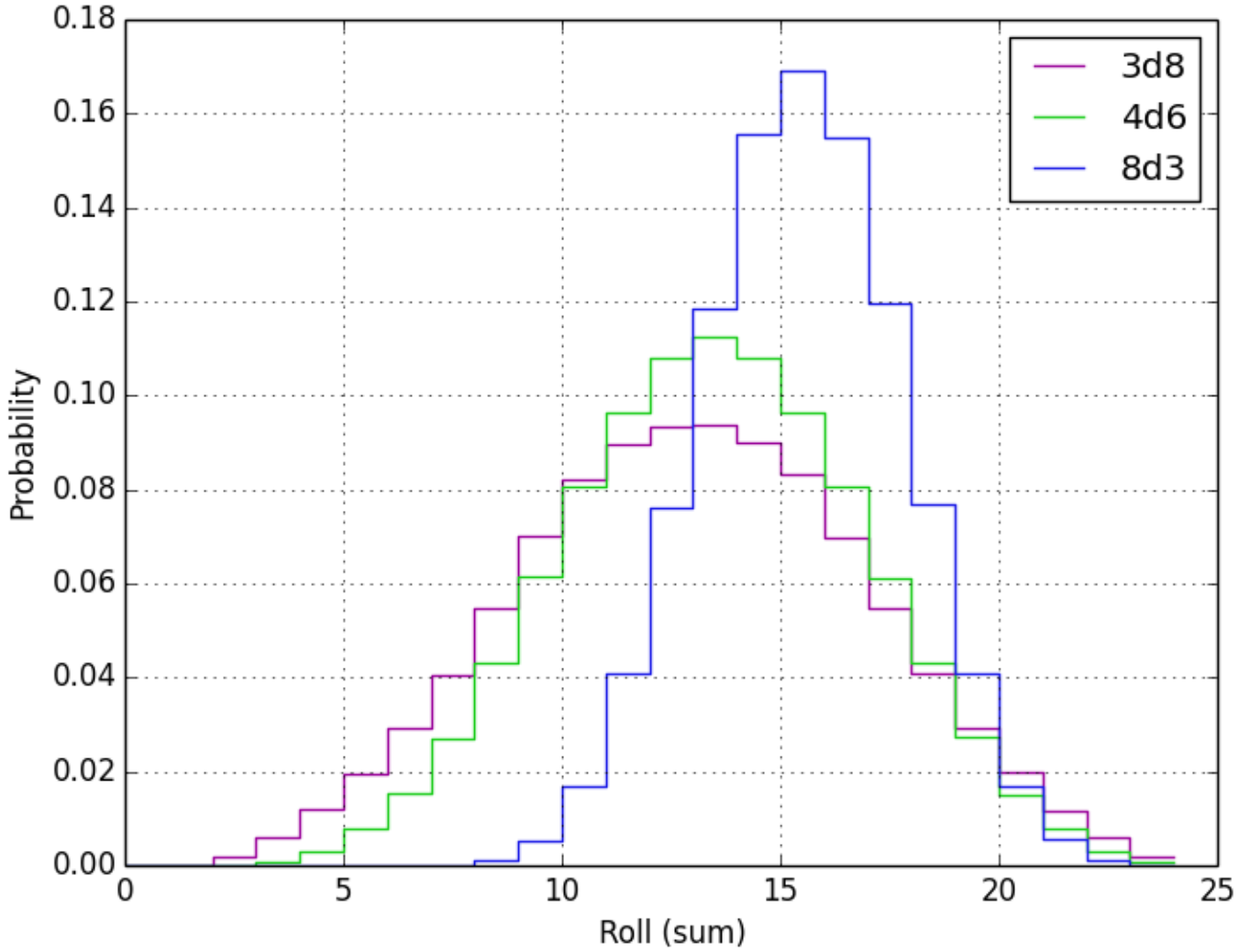
```
import numpy as np
Nsims = 500000

# nroll: n=#of dice, d=#sides per dice
nroll = lambda n,d: sum([int(np.ceil(np.random.rand()*d)) for i in range(n)])

r3d8, r4d6, r8d3 = np.zeros(25), np.zeros(25), np.zeros(25)
for i in range(Nsims):
    r3d8[nroll(3,8)]+=1
    r4d6[nroll(4,6)]+=1
    r8d3[nroll(8,3)]+=1

import matplotlib.pyplot as plt
plt.step(numpy.arange(25),r3d8/float(Nsims),color=(.6,.0,.6),label='3d8')
plt.step(numpy.arange(25),r4d6/float(Nsims),color=(0.1,.8,.1),label='4d6')
plt.step(numpy.arange(25),r8d3/float(Nsims),color=(0.1,.1,.9),label='8d3')
plt.legend()
plt.xlabel('Roll (sum)')
plt.ylabel('Probability')
plt.grid()
plt.savefig('dice_dnd.png')
```

Timbits best served by 8d3 (or 24d1)



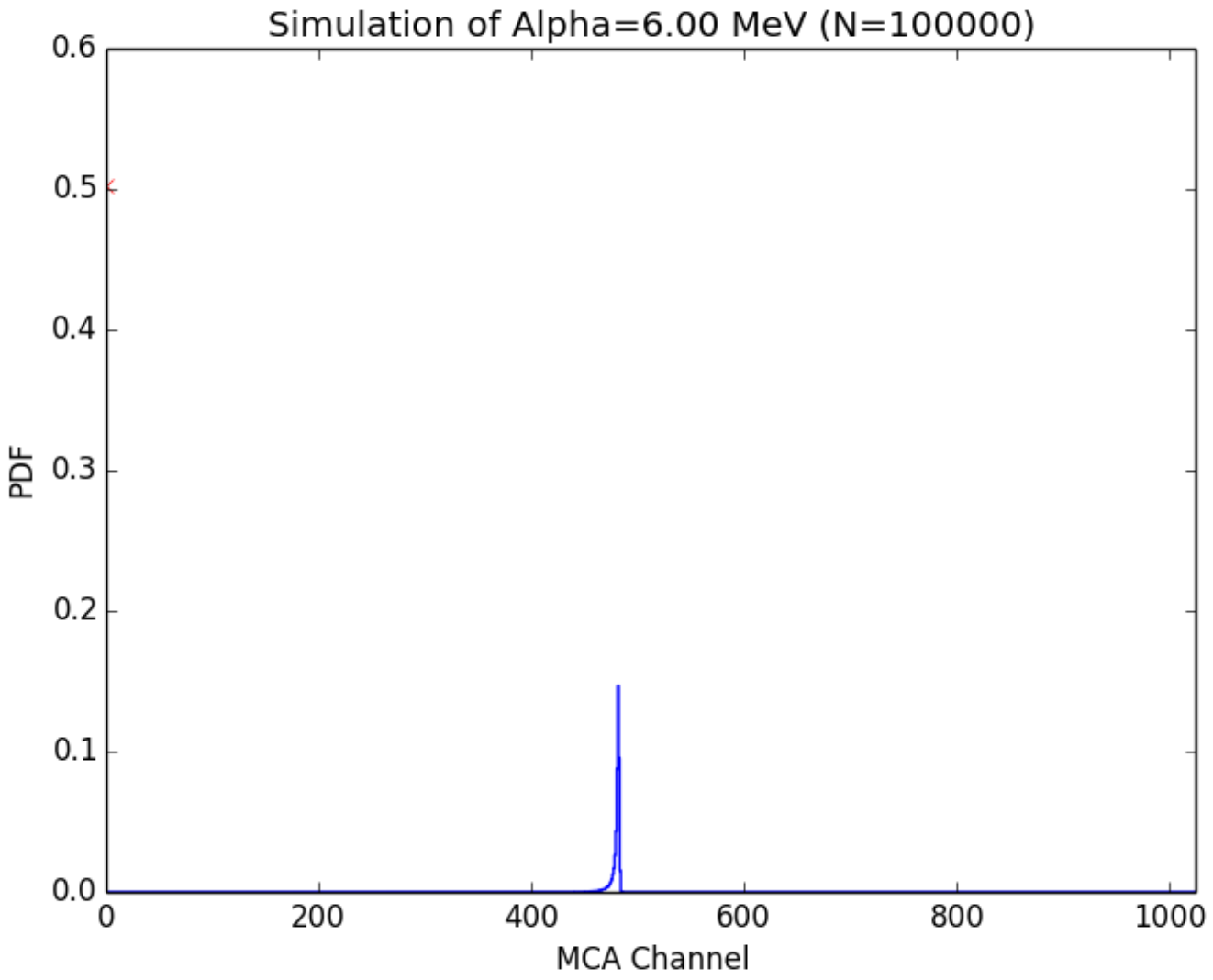
Example: Silicone Alpha Detector

- Simulate alpha at constant energy & random direction.
- Hint, if you use spherical don't assume random theta/phi are correct! (see link)

http://mathworld.wolfram.com/SpherePointPicking.html

Example (alpha counter)

GIT: 2014PAG_ComputerDay/master/MonteCarlo/diode_alpha.py



Geant4

- Tools and libraries to help you simulate particles propagating through matter
  - cross-sections/half-lives/physical-processes/…
- Tools to assist in creation of detector signals
- Flexible and arbitrary
  - Language: C++
- *Every collaboration seems to do something unique*