Statistical and Mathematical Methods

Statistical and Mathematical Methods for Data Science
DS5003

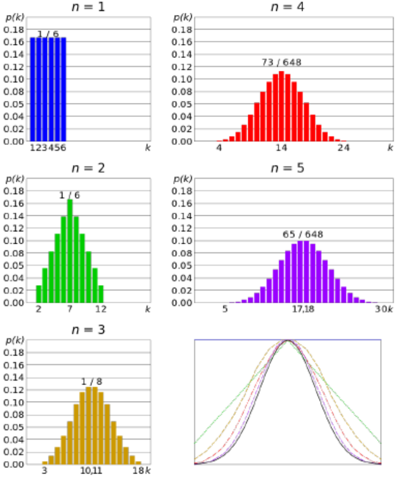
Dr. Nasir Touheed

Monte Carlo Simulations

The first thoughts and attempts I made to practice [the Monte Carlo Method] were suggested by a question which occurred to me in 1946 as I was convalescing from an illness and playing solitaires. The question was what are the chances that a Canfield solitaire laid out with 52 cards will come out successfully? After spending a lot of time trying to estimate them by pure combinatorial calculations, I wondered whether a more practical method than "abstract thinking" might not be to lay it out say one hundred times and simply observe and count the number of successful plays. This was already possible to envisage with the beginning of the new era of fast computers, and I immediately thought of problems of neutron diffusion and other questions of mathematical physics, and more generally how to change processes described by certain differential equations into an equivalent form interpretable as a succession of random operations. Later ... [in 1946, I] described the idea to John von Neumann, and we began to plan actual calculations.

Why Did the Empirical Rule Work?

- Because we are reasoning not about a single spin, but about the mean of a set of spins
- And the central limit theorem applies



The Central Limit Theorem (CLT)

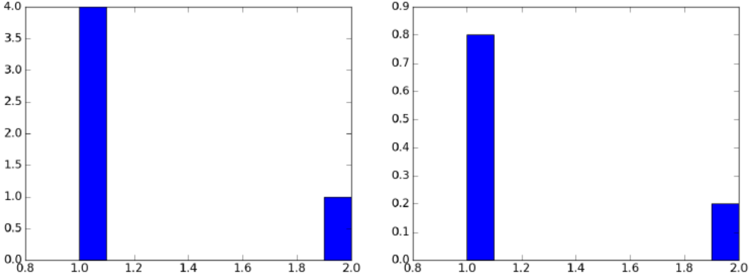
- Given a sufficiently large sample:

  1) The means of the samples in a set of samples (the sample means) will be approximately normally distributed,

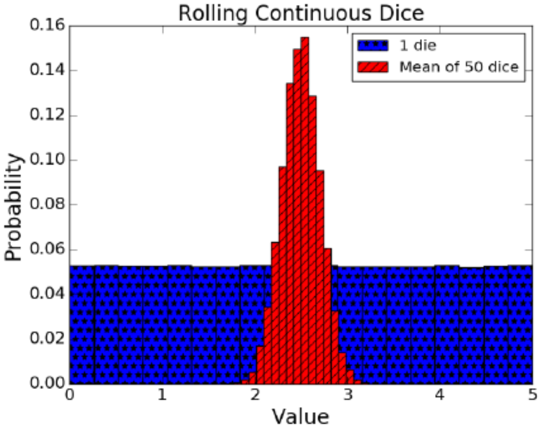
  2) This normal distribution will have a mean close to the mean of the population, and

  3) The variance of the sample means will be close to the variance of the population divided by the sample size.

Weighting the Bins



Mean of rolling 1 die = 2.49759575528, Std = 1.4439045633
Mean of rolling 50 dice = 2.49985051798, Std = 0.204887274645



Monte Carlo Simulation

- Monte Carlo Simulation, also known as the Monte Carlo Method or a multiple probability simulation.
- MCS is a mathematical technique, which is used to estimate the possible outcomes of an uncertain event.
- The Monte Carlo Method was invented by John von Neumann and Stanislaw Ulam during World War II to improve decision making under uncertain conditions.
- It was named after a well-known casino town, called Monaco, since the element of chance is core to the modeling approach, similar to a game of roulette.

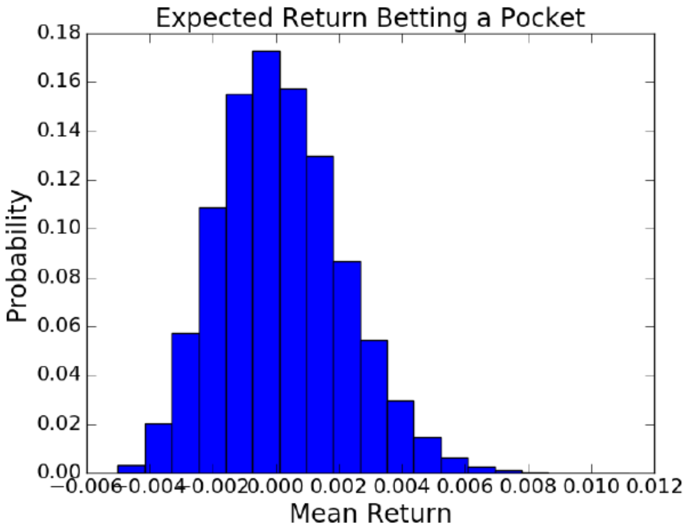
Try It for Roulette

```
numTrials = 50000
numSpins = 200
game = FairRoulette()

means = []
for i in range(numTrials):
    means.append(findPocketReturn(game, 1,
   numSpins)[0]/numSpins)

pylab.hist(means, bins = 19,
           weights = pylab.array(len(means)*[1])/len(means))
pylab.xlabel('Mean Return')
pylab.ylabel('Probability')
pylab.title('Expected Return Betting a Pocket')
```

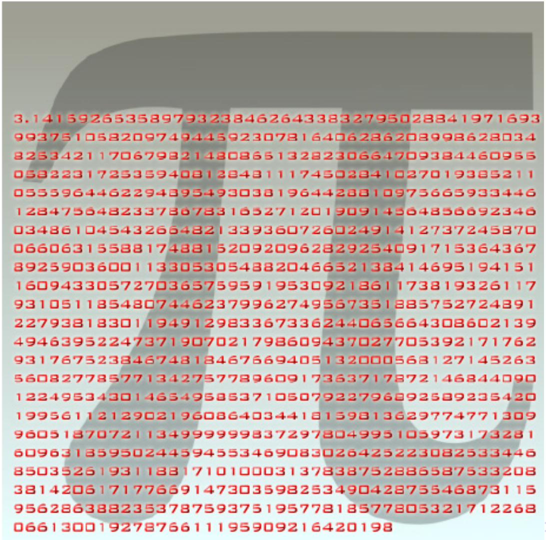
Means Close to Normally Distributed!



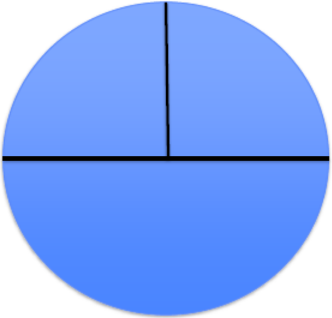
Try It for Roulette

- It doesn't matter what the shape of the distribution of values happens to be
- If we are trying to estimate the mean of a population using sufficiently large samples
- The CLT allows us to use the empirical rule when computing confidence intervals

Estimating PI



Estimating PI



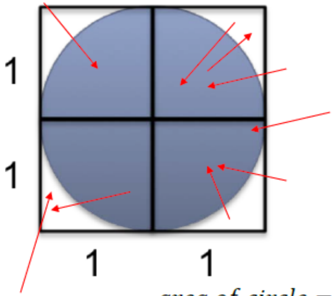
$$\frac{circumference}{diameter} = \Pi$$

$$area = \Pi * radius^2$$

Bufoon-Laplace

- Buffon's needle problem is a question first posed in the 18th century by Georges-Louis Leclerc, Comte de Buffon
- Suppose we have a floor made of parallel strips of wood, each the same width, and we drop a needle onto the floor.
- What is the probability that the needle will lie across a line between two strips?
- Buffon's needle was the earliest problem in geometric probability to be solved;
- it can be solved using integral geometry. The solution for the sought probability p, in the case where the needle length *l* is not greater than the width t of the strips, is $p = \frac{2}{\pi}\frac{l}{t}$
- This can be used to design a Monte Carlo method for approximating the number π, although that was not the original motivation for de Buffon's question

Estimating PI Bufoon-Laplace



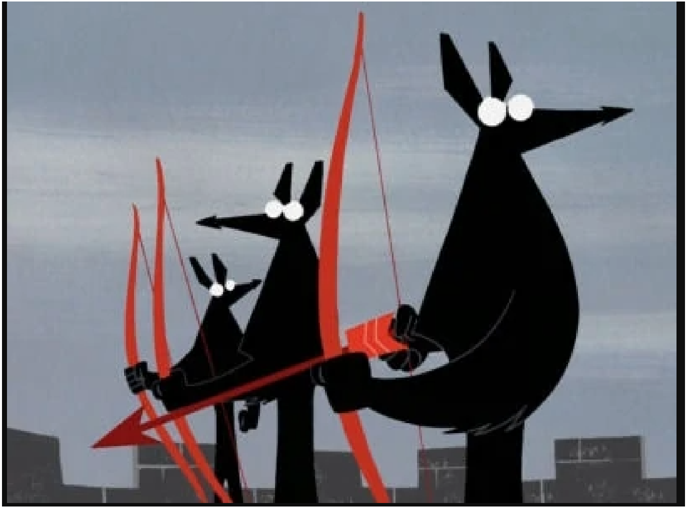
$$A_s = 2*2 = 4$$
$$A_c = \pi r^2 = \pi$$

$$\frac{needles\ in\ circle}{needles\ in\ square} = \frac{area\ of\ circle}{area\ of\ square}$$

$$area\ of\ circle = \frac{area\ of\ square * needles\ in\ circle}{needles\ in\ square}$$

$$area\ of\ circle = \frac{4 * needles\ in\ circle}{needles\ in\ square}$$

Arrows Are More Fun than Needles



Simulating Buffon-Laplace Method

```python
def throwNeedles(numNeedles):
    inCircle = 0
    for Needles in range(1, numNeedles + 1, 1):
        x = random.random()
        y = random.random()
        if (x*x + y*y)**0.5 <= 1.0:
            inCircle += 1
    return 4*(inCircle/float(numNeedles))
```

Simulating Buffon-Laplace Method

```
def getEst(numNeedles, numTrials):
    estimates = []
    for t in range(numTrials):
        piGuess = throwNeedles(numNeedles)
        estimates.append(piGuess)
    sDev = numpy.std(estimates)
    curEst = sum(estimates)/len(estimates)
    print('Est. = ' + str(curEst) +\
          ', Std. dev. = ' + str(round(sDev, 6))\
          + ', Needles = ' + str(numNeedles))
    return (curEst, sDev)
```

Estimating PI

```
def estPi(precision, numTrials):
    numNeedles = 1000
    sDev = precision
    while sDev >= precision/1.96:
        curEst, sDev = getEst(numNeedles, numTrials)
        numNeedles *= 2
    return curEst
```

Estimating PI

```
Est. = 3.1484400000000012, Std. dev. = 0.047886, Needles = 1000
Est. = 3.1391799999999987, Std. dev. = 0.035495, Needles = 2000
Est. = 3.1410799999999997, Std. dev. = 0.02713, Needles = 4000
Est. = 3.141435, Std. dev. = 0.016805, Needles = 8000
Est. = 3.141355, Std. dev. = 0.0137, Needles = 16000
Est. = 3.1413137500000006, Std. dev. = 0.008476, Needles = 32000
Est. = 3.141171874999999, Std. dev. = 0.007028, Needles = 64000
Est. = 3.1415896874999993, Std. dev. = 0.004035, Needles = 128000
Est. = 3.1417414062499995, Std. dev. = 0.003536, Needles = 256000
Est. = 3.14155671875, Std. dev. = 0.002101, Needles = 512000
```

Being Right is Not Good Enough

- Not sufficient to produce a good answer
- Need to have reason to believe that it is close to right
- In this case, small standard deviation implies that we are close to the true value of π

Right?

Is it Correct to State

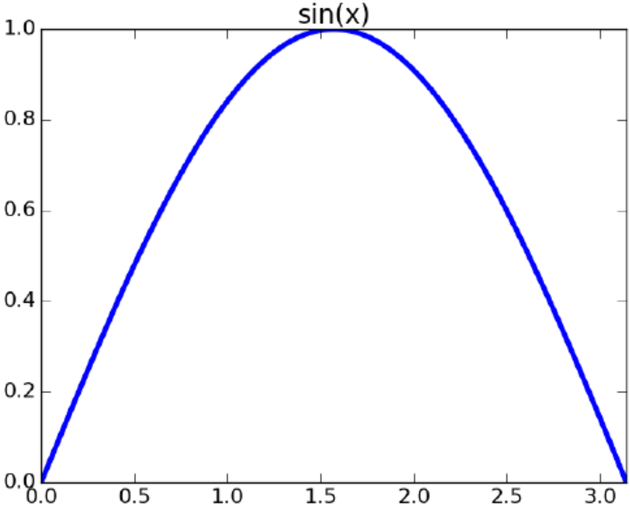
- 95% of the time we run this simulation, we will estimate that the value of pi is between 3.13743875875 and 3.14567467875?
- With a probability of 0.95 the actual value of π is between 3.13743875875 and 3.14567467875?
- Both are factually correct
- But only one of these statement can be inferred from our simulation
- $statiscally\ valid \neq true$

Generally Useful Technique

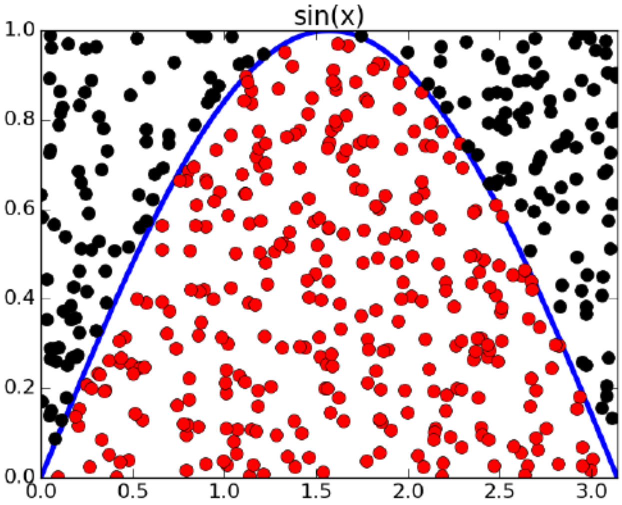
- To estimate the area of some region, R
  - Pick an enclosing region, E, such that the area of E is easy to calculate and R lies completely within E
  - Pick a set of random points that lie within E
  - Let F be the fraction of the points that fall within R
  - Multiply the area of E by F
- Way to estimate integrals

$$\int_0^{\pi} \sin(x)$$

Sin X



Random Points



Monte Carlo Simulations

- Regardless of what tool you use, Monte Carlo techniques involves three basic steps:
  - Set up the predictive model, identifying both the dependent variable to be predicted and the independent variables (also known as the input, risk or predictor variables) that will drive the prediction.
  - Specify probability distributions of the independent variables. Use historical data and/or the analyst's subjective judgment to define a range of likely values and assign probability weights for each.
  - Run simulations repeatedly, generating random values of the independent variables. Do this until enough results are gathered to make up a representative sample of the near infinite number of possible combinations.

Monte Carlo Simulations

- We can run as many Monte Carlo Simulations as we wish by modifying the underlying parameters you use to simulate the data.
- However, you'll also want to compute the range of variation within a sample by calculating the variance and standard deviation, which are commonly used measures of spread.
- Variance of given variable is the expected value of the squared difference between the variable and its expected value.
- Standard deviation is the square root of variance. Typically, smaller variances are considered better.