Simulation and Modelling

Spring 2024
CS4056

Muhammad Shahid Ashraf

Notes

Monte Carlo Simulations

Notes

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.

Notes

Try It for Roulette

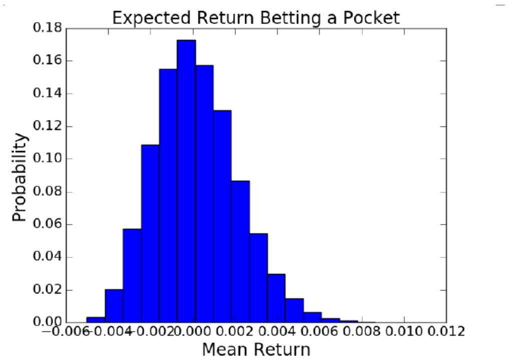
```
numTrials = 50000
numSpins = 200
game = FairRoulette()

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')
```


Notes

Means Close to Normally Distributed!



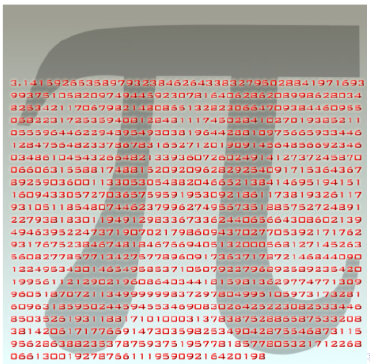
Notes

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

Notes

Estimating PI



Notes

Estimating PI

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

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

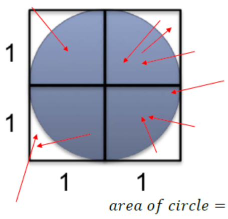
Notes

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

Notes

Estimating PI Bufoon-Laplace



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

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

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

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

Notes

Arrows Are More Fun than Needles

Notes

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))
```


Notes

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)
```


Notes

Estimating PI

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


Notes

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
```


Notes

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?

Notes

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
- $statiscally\ valid\ \neq true$

Notes

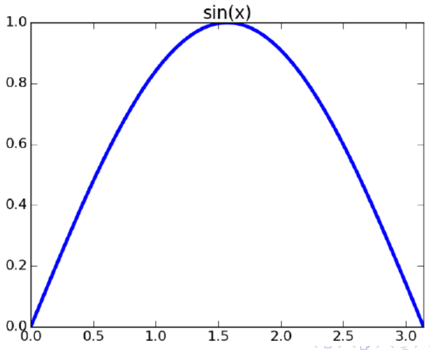
Generally Useful Technique

- 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

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

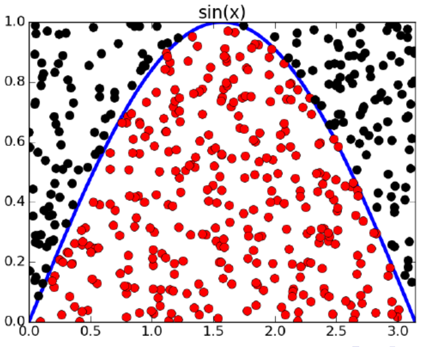
Notes

Sin X



Notes

Random Points



Notes