Research Methods

Robert E Simpson

Singapore University of Technology & Design

robert_simpson@sutd.edu.sg

October 3, 2018

SINGAPORE UNIVERSITY OF TECHNOLOGY AND DESIGN

Objectives

You should be able to:

- Design a Monte Carlo algorithms to compute the propagation of error
- Write and run Monte Carlo programs to show the difference between standard error and standard deviation.
- Write a program that demonstrates how the central limit theorem produces a normal sample distribution irrespective of the parent distribution function.

无关的

Propagation of Error Equation

$$\boxed{\sigma_x^2 = \sigma_u^2 \left(\frac{\partial x}{\partial u}\right)^2 + \sigma_v^2 \left(\frac{\partial x}{\partial v}\right)^2} \tag{1}$$

Assumptions:

- Errors are small (truncated taylor series)
- Measurement errors in the input variables are statistically independent
- Successive measurement errors are statistically independent

Disadvantages:

- Unsuspected covariances
- Differentiation/Calculation errors
- Some underlying equations cannot be analytically differentiated

Coefficient of variation

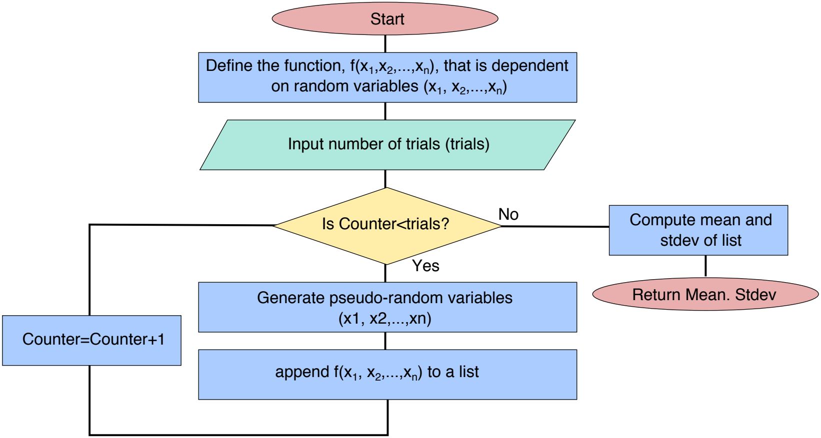
The coefficient of variation is measure of the relative magnitude of the standard deviation. It is useful way to understand if the standard deviation is large compared to the measured value. The larger the Coefficient of Variation, the more imprecise the measurements are.

$$C.o.V = \frac{\sigma}{\bar{x}}$$

Monte Carlo Algorithms

- Generate a probability distribution which describes some stochastic process
- Pseudorandom variables are used to calculate the value of some underlying process
- The process is repeated many times, the different outcomes are plotted on a histogram
- The histogram can then be analysed
- For PoE, the main advantage is that we do not need to differentiate the underlying equation
- The main disadvantage is that the algorithm is computationally expensive, since the SNR scales with $\sqrt{N}$

Monte Carlo Flow



SINGAPORE UNIVERSITY OF TECHNOLOGY AND DESIGN

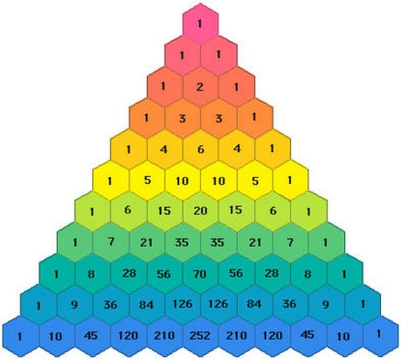
Useful Functions

`scipy.stats.norm.rvs`	Normal random variable
`np.random.binomial(n,p)`	Binomial random variable
`np.mean(x)`	Arithmetic Mean
`np.std(x, ddof=1)`	Sample Variance
`matplotlib.pyplot.savefig('fname.pdf')`	Save the figure as a PDF

Case Problem 4.6

CLT &Pascal's/Yanghui'sTriangle

- Any other number is found by adding the two numbers above it.
- Successive rows of Pascal's Triangle predict the number of successes in a binomial experiment with an increasing numbers of trials.
- The 1^{st} number in a row of Pascal's Triangle corresponds to the frequency of 0 successes, therefore, the 3^{rd} number in a row corresponds to the frequency of 2 successes, and not 3.



SINGAPORE UNIVERSITY OF TECHNOLOGY AND DESIGN

Case Problem 4.6

CLT &Pascal's/Yanghui'sTriangle

Pascal's triangle can show the link between Binomial and Gaussian distributions. Assume a triangular arrangement of pins as shown in the diagram below. A ball dropped into the device strikes the top pin and has a 50% probability of striking either of the two pins below in the next row. The ball bounces down until reaches the bottom and is collected in one of the bins.

(a) Find a general expression for the probability of a ball landing in a given bin after dropping through N rows of pins.

(b) Write a Monte-Carlo algorithm to predict the number of balls in each bin.

(c) Simulate the 100 and 100k balls falling through 8 layers and 32 layers. Compute the C.o.V and plot (normalised) histograms to show the probability of occurrences. What do you notice?

(d) Plot Gaussian/Normal PDF using the theoretical mean and standard deviation for 100 and 100k balls falling through 8 layers and 32 layers.

Case Problem 4.7

Monte Carlo PoE

The resonant frequency of an electronic inductor–capacitor (LC) filter is given by:

$$f_0 = \frac{1}{2\pi\sqrt{LC}}$$

The inductor (L) has an inductance, $L = 2.5\ mH$ with a relative tolerance of $\pm 10\%$ and the capacitor as capacitance of $C = 10\mu F$ with a retaliative tolerance of $\pm 5\%$.

i. Draw a flow diagram to show how the Monte Carlo method can be used to compute the expected frequency and frquency spread.

ii. Use the Monte Carlo method to plot the frequency, f_0, probability distribution for the circuit if L and C are selected randomly.

iii. If 36 circuits are selected at random, what is the mean of f0 and what is the standard error in the estimate of the true mean of the circuit population?

iv. Use an MC algorithm to show graphically that increasing the number of samples taken from the manufacturing process decreases the standard error ($\sigma_{\bar{c}}$).

SINGAPORE UNIVERSITY OF TECHNOLOGY AND DESIGN

Case Problem 4.8

Monte Carlo Boat

Remember the boat question?

Use the Monte Carlo approach to compute the standard error in predicting the time for a boat to move 30° around a curve of radius 10 m, moving at a speed of 0.6 ms^{-1}. Assume that the captain is able to estimate angle with an error of 5°, radius with an error of 0.1 m, and velocity with an error of 0.1 ms^{-1}.

(a) Plot a histogram and find the standard error, using 500 simulations and a sample size of 10.

(b) Plot a histogram and find the standard error, using 500 simulations and a sample size of 40.

(c) What is the difference between (a) and (b)

Case Problem 4.9

Exponential growth and uncertainty

The consumption of resources is modelled as:

$$Q(t) = \int_0^t P_0 e^{rt} dt = \frac{P_0}{r}(e^{rt} - 1) \tag{2}$$

Where P_0 is the initial consumption rate, and r is the exponential rate of growth. The world coal consumption in 1986 was equal to 5.0 billion tonnes per a year and the estimated recoverable reserves of coal were estimated at 1000 billion tonnes.

(a) If the growth rate is 2.7 % per a year, how many years before the coal reserves are depleted?

(b) Assume that the growth rate, r, and the recoverable reserves, Q, are subject to random uncertainty with $\sigma_r = 0.2$ % (absolute) and $\sigma_Q = 10\%$ (relative) respectively. Compute the standard error in the estimated time before the coal reserves are depleted, σ_t.

Summary

We have learnt how to use the Monte-Carlo method to compute the build-up of errors. All of the Case Problems use the same same basic routine:

1. Define the function that depends on uncertain variables
2. Loop over the function and each time generate different pseudo-random variables for the inputs to the function
3. Statistically analyse output generated

CP4.6 Pascals Triangle

(a) For N layers, the x^{th} bin probability is:
$$P(x,n,p) = \binom{n}{x} p^x q^{n-x} = \frac{n!}{x!(n-x)!} p^x (1-p)^{n-x}$$

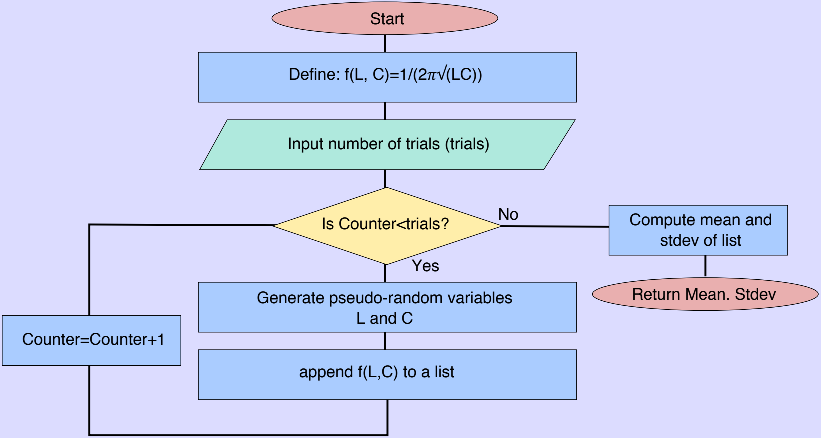
CP4.6 Pascals Triangle

```
def binomialCoeff(n, k):
    result = 1
    for i in range(1, k+1):
        result = result * (n-i+1) / i
    return result

NBalls=1000
Nlayers=32
Prob=0.5

Tmean=Prob* Nlayers
Tstdv=np.sqrt(Nlayers*Prob**2)

print "Theoretical␣mean=", Tmean
print "Theoretical␣Standard␣Dev␣=", Tstdv
print "Coefficient␣of␣Variation", Tstdv/Tmean

data=[]
for i in range(NBalls):
    data.append(np.random.binomial(Nlayers,Prob))

x=np.linspace(0,(Tmean+4*Tstdv),1000)
y=ss.norm(Tmean,Tstdv).pdf(x)

plt.figure()
rob=plt.hist(data, bins=(Nlayers+1), align='left',  range=(0,Nlayers+1), normed
     =1,edgecolor='black')
plt.plot(x,y)
```

SINGAPORE UNIVERSITY OF TECHNOLOGY AND DESIGN

CP4.7 Solution– LC circuit

Part i.

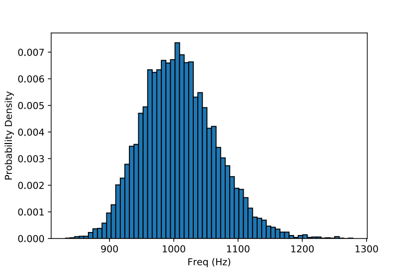


CP4.7 Solution– LC circuit

Part ii.

```
def f0(l,c):
    return 1.0/(2*np.pi*np.sqrt(l*c))

def OneSample(Num): #returns the mean
      and standev of a sample containing
       Num random variables
    freq=[]
    for sample in range(Num):
        l=ss.norm.rvs(2.5e-3, 2.5e-4)
        c=ss.norm.rvs(10e-6, 0.5e-6)
        freq.append(f0(l,c))
    mean=np.mean(freq)
    sigma=np.std(freq, ddof=1) #sample
          standard deviation
    #print mean, sigma
    return mean,sigma, freq


####CASE PROBLEM 4.7(b)
F=OneSample(10000)[2] #take a sample of
      10000 random variables
plt.figure()
plt.hist(F, bins='auto',edgecolor="black
      ", align='left')
plt.xlabel('Freq␣(Hz)')
plt.ylabel('Freq')
plt.xlim(800,1300)
fname='cp4_7b'+str(1000)+'.pdf'
plt.savefig(fname)
```

Probability density vs frequency for 10k trials.



$\bar{x} = 1012.41$

$\sigma = 58.17$

Part iii

if N=36, then:

$$\bar{x} = 1008.93$$

$$\sigma = 56.85$$

The standard error is given by:

$$\sigma_{\bar{x}} = \frac{\sigma}{\sqrt{N}} = \frac{56.85}{\sqrt{36}} = 9.48$$

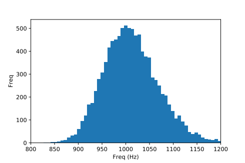
This means that there is a 68.2% chance that the true mean of the population is $1008.93 - 9.48 < \mu < 1008.93 + 9.48$.

$$999.46 < \mu < 1018.41$$

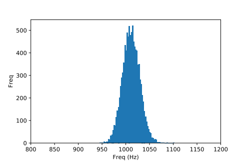
The true mean is actually,
$f_0 = \frac{1}{2\pi\sqrt{LC}} = \frac{1}{2\pi\sqrt{2.5\times10^{-3}\times10\times10^{-6}}} = 1006.58\ \mathrm{Hz}$, which is indeed between 999.46 Hz and 1018.41 Hz.

```
####CASE PROBLEM 4.7(d)
def sampleDist(N_samples,NExpts): # creates a sample distribution (i.e. the distribution of means NExpts)
    freq=[]
    for sample in range(NExpts):
        freq.append((OneSample(N_samples))[0])
    return freq

numofsamples=10
F=sampleDist(numofsamples,500)
plt.figure()
plt.hist(F, bins='auto',edgecolor="black", align='left')
plt.xlabel('Freq (Hz)')
plt.ylabel('Freq')
plt.xlim(800,1200)
fname='cp4_7d'+str(numofsamples)+'.pdf'
plt.savefig(fname)
```

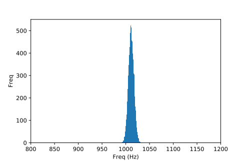
N_samples=1, $\bar{x} = 1011.55$, $\sigma = 58.08$



N_samples=10, $\bar{x} = 1011.43$, $\sigma = 18.1$



N_samples=100, $\bar{x} = 1011.36$, $\sigma = 5.78$



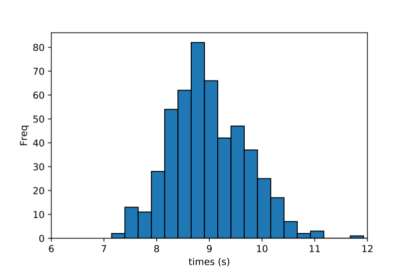
When N=1, $\sigma = 58$. Increasing N to 36, should allow a standard error of approximately $\sigma_{\bar{x}} = \frac{58}{\sqrt{36}} = 9.66$. When N=36, the MC gives $\sigma_{\bar{x}} = 9.49$, which is consistent.

CP4.8-Boat Solution

```
def boat_t(r,angle_rad,v):
    return r*angle_rad*np.pi/(180*v)

def OneSample(r,theta,v,Er,Etheta,Ev,Num):
    t=[]
    for sample in range(Num):
        angle=ss.norm.rvs(theta,Etheta)
        speed=ss.norm.rvs(v,Ev)
        radius=ss.norm.rvs(r, Er)
        t.append(boat_t(radius, angle, speed))
    mean=np.mean(t)
    sigma=np.std(t, ddof=1) #sample standard deviation
    #print "stats:", mean, sigma
    return mean, sigma

def sampleDist(N_samples,NExpts): #creates a sample distribution (i.e. the
      distribution of means NExpts)
    times=[]
    for sample in range(NExpts):
        times.append(OneSample(10,30,0.6,0.1,5,0.1,N_samples)[0])
    return times

boatimes=sampleDist(10,500)
plt.figure()
plt.hist(boatimes, bins='auto', edgecolor='black')
plt.xlabel('times␣(s)')
plt.ylabel('Freq')
plt.xlim(6,12)
```

CP4.8 Solution

Sample of 10, 500 expts



Sample of 40, 500 expts