Scilab Manual for
Probability Theory and Random Processes
by Prof Shital Thakkar
Others
Dharmsinh Desai University[1]

Solutions provided by
Prof Shital Thakkar
Others
Dharmsinh Desai University

April 20, 2024

[1]Funded by a grant from the National Mission on Education through ICT, http://spoken-tutorial.org/NMEICT-Intro. This Scilab Manual and Scilab codes written in it can be downloaded from the "Migrated Labs" section at the website http://scilab.in

Contents

List of Experiments

List of Figures

Experiment: 1

Write a program to find probability of tossing a coin and rolling a die through large no. of experimentation.

Scilab code Solution 1.1 Calculates probability of sum of tossing two dice

```
// Operating System : Windows XP or later,
// Scilab           : 5.3.3

//Program of Tossing two dice and observing Probability of sum of their front face.
//for e.g. Probability of sum of two dice = 2 is 1/36 as there are 36 possibilities and sum = 2 can// occur only one combination that is both face = 1
clc;
clear;
clf;
N = 10000;  // Number of times tossing of die performed
count = 0;  // Counter for counting number of times
```

```
sum of die
11 for i = 1:N
12     y1 = ceil(rand(1)*6); // output of die 1
13     y2 = ceil(rand(1)*6); // output of die 2
14     if((y1+y2) == 3)      // check for sum of front face of both die is == 3(change sum and)
15         count = count + 1;//increment the count value when sum = 3 occurs
16     end
17     prob1(i) = count/i;       // no. of times sum of die = 3/total no. trials
18 end
19 plot(prob1)
20 xlabel('Number of Trials');
21 ylabel('Probability');
22 title('Probability of getting sum of dots on faces of a dice to be 3');
23
24
25 //Assignment : Program can be checked for other values of sum at line number 10.
```

Scilab code Solution 1.2 Finds probability of getting Head when a coin is tossed

```
// Operating System : Windows XP or later ,
// Scilab           : 5.3.3


//This program find probability of getting Head when a coin is tossed.
//Probability = 1/2 = 0.5 as there are two possible outcomes in coin tossing experiment.
```

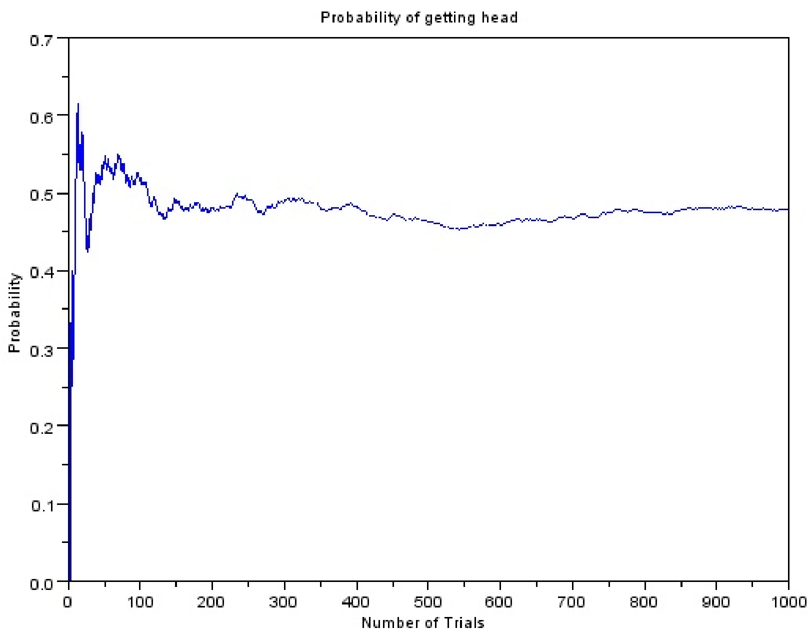

Figure 1.1: Finds probability of getting Head when a coin is tossed

```
clc;
clear all;
clf;
a1 = 1000;
count1 = 0;
for i = 1:a1
x = round(rand(1));
    //round: the elements to nearest integer
    //rand:returns a pseudorandom, scalar value drawn from a uniform
    //distribution on the unit interval.
    if(x==1)// HEAD— '1',condition that detects 'HEAD' comes or not
        count1 = count1 + 1;//increment the count value when head occurs
    end
    p(i)= count1/i;//probability of head occuring at ith trail
end
plot(1:a1,p)
//plot the prob. at ith trail(plots discrete sequence)
xlabel('Number of Trials');
ylabel('Probability');
title('Probability of getting head');
//Assignment:
//1.   perform above experiment with n = 100,1000 .
//2.   Extend the above experiment to find probability of 3 heads in 4 coin tosses.
//     Match the result theoretically.
```

Experiment: 2

To generate Uniform, Gaussian and Exponential distributed data for given mean and variance.

Scilab code Solution 2.1 Generation of Uniform Data

```
//To generate Uniform, Gaussian and Exponential distributed data.
// Operating System : Windows XP or later,
// Scilab           : 5.3.3
//NOTE:EXECUTE ONE BY ONE SEGEMENT

//[1] Uniform Data Generation
clc;
clear all;
//b = input('higher limit of uniform r.v. b = ')// Enter higher limit of uniform r.v.
//a = input('lower limit of uniform r.v. a =')// Enter lower limit of uniform r.v.
```

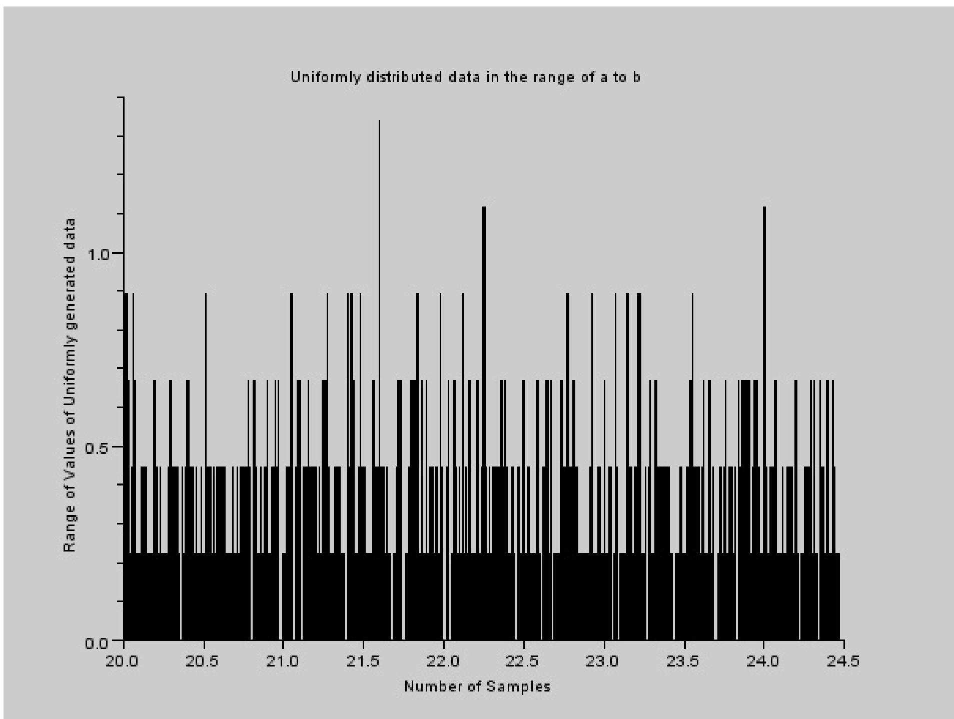

Figure 2.1: Generation of Uniform Data

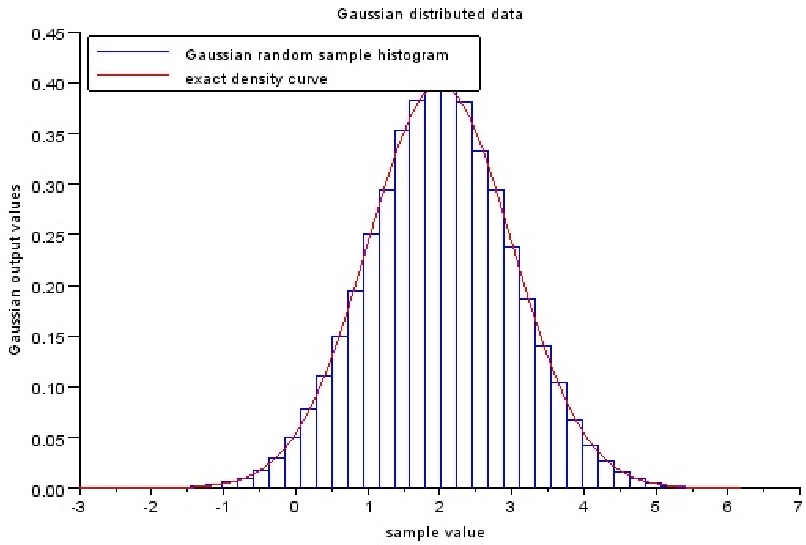

Figure 2.2: Gaussian Data Generation

```
clf();
b=4;//higher range of uniform data
a=2;//lower range of uniform data
x=a + (b-a).*rand(1,1000,'uniform'); //uniform data
    generation of length 100
histplot(1000,x)//histogram of generated data
xlabel('Number of Samples');
ylabel('Range of Values of Uniformly generated data'
    );
title('Uniformly distributed data in the range of a
    to b');
```

Scilab code Solution 2.2 Gaussian Data Generation

```
//To generate Uniform, Gaussian and Exponential distributed data.
// Operating System : Windows XP or later,
// Scilab           : 5.3.3

//[2] Expoenential data generation & Mean and Variance Calcultaion of Exponential distributed data.
clc;
clear all;
clf();
//(i) Exponential data generation
lambda = 2;//or lambda = input('enter lemda value for exponential r.v.')//lemda of exponential data;
X = grand(10000,1,"exp", 1/lambda);
Xmax = max(X);
histplot(40, X, style=2)
x = linspace(0,max(Xmax),100)';
plot2d(x,lambda*exp(-lambda*x),strf="000",style=5)
legend(["exponential random sample histogram" "exact density curve"]);
xlabel('sample value');
ylabel('Exponential output values');
title('Exponential distributed data');
```

Scilab code Solution 2.3 Exponential Data Generation

```
//To generate Uniform, Gaussian and Exponential distributed data.
// Operating System : Windows XP or later,
// Scilab           : 5.3.3

```

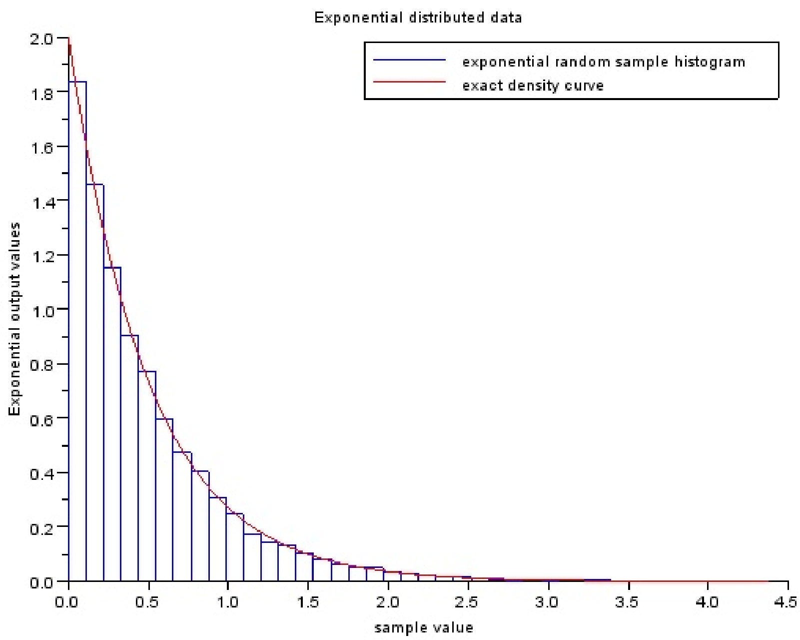

Figure 2.3: Exponential Data Generation

```
//[3]Gaussian data generation
//Gaussian data generation
clc;
clear all;
clf();
//m = input('enter mean value for Gaussian r.v.')
//vari = input('enter mean value for Gaussian r.v.')
m = 2;//mean value of gaussian data
sd = 1;//standard deviation
vari = sd^2;
X = grand(100000,1,"nor", m,sd);
Xmax = max(X);
clf()
histplot(40, X, style=2)
x = linspace(-10,max(Xmax),100)';
plot2d(x,(1/(sqrt(2*%pi*vari)))*exp(-0.5*(x-m).^2/
   vari),strf="000",style=5)
xlabel('sample value');
ylabel('Gaussian output values');
title('Gaussian distributed data');
legend(["Gaussian random sample histogram" "exact
   density curve"],2);
```

Experiment: 3

Write a program to generate M trials of a random experiment having specific number of outcomes with specified probabilities.

Scilab code Solution 3.1 Random experiment with outputs in specific range

```
// Operating System : Windows XP or later ,
// Scilab            : 5.3.3

//Write a program to generate M trials of a random
   experiment having specific
//number of outcomes with specified probabilities.
//here No. of trials = 1000,no. outcomes(rv) = 3
   with specied probability entered by user
clc;
clear all;
rand('seed')//check
M = 1000; //Number of trials of random experiment
outcomes = 3;  //Possible number of outcomes of
```

```
random experiment
for i = 1:outcomes-1
    r(i) = input('enter upper range of probability of r.v. (values in the 0<r<1) :')//enter values in the 0<r<1
    if r(i)>1 then
        error('Enter values in the range 0<r<1')
    end
end
x = zeros(M,1);
for i = 1:M
    u = rand(1,1);//random outcome
    if u <= r(1) then
        x(i,1)=1;//assign rv value = 1 if u<=r(1)
    elseif u > r(1) & u<= r(2)
        x(i,1)=2;//second rv value
    else
        x(i,1)=3; //third rv value
    end
end
count1=0;count2=0;count3=0;
for i=1:1000
    if x(i,1)==1 then
        count1 = count1 + 1;
    elseif x(i,1)== 2
        count2 = count2 + 1;
    else
        count3 = count3 + 1;
    end
end
estP1 = count1/M;disp(estP1)//estimated probability of generated random variable
estP2 = count2/M;disp(estP2)//estimated probability of generated random variable
estP3 = count3/M;disp(estP3)//estimated probability of generated random variable

//Assignment:
```

```
//1. Extend this program for 4 number of random
    variable
//2. Extend this program for more number of trials.
    i.e. M = 5000,10000 etc.
```

Experiment: 4

To find estimated and true mean of Uniform, Gaussian and Exponential distributed data.

Scilab code Solution 4.1 Comaparison of True and estimated statics of Uniform Data

```
// Operating System : Windows XP or later,
// Scilab           : 5.3.3
//here we generate uniformly distributed data
   compare its statistics with calculated using
   equation.



//[1] Mean and Variance Calcultaion of Uniformly
   distributed data

clc;
clear all;
b=4;//higher range of uniform data
a=2;//lower range of uniform data
x=a + (b-a).*rand(1,1000); //Uniform data generation
```

```
    using function
Uni_true_mean=mean(x)
mprintf('Uniform True Mean = %f',Uni_true_mean)
Uni_true_var = variance(x)
mprintf('\n Uniform True Mean = %f',Uni_true_var)
px = 1/(b-a)//pdf calcultaion of uniform r.v.
m_uniform=integrate('x*px','x',a,b)//mean
   calcultaion of uniform r.v.
fsq_uniform=integrate('(x^2)*px','x',a,b)//mean
   square value of uniform r.v.
var_uniform = fsq_uniform - (m_uniform).^2//variance
    of uniform r.v.

mprintf('\n Uniform Calculated Mean = %f',m_uniform)
mprintf('\n Uniform Calculated Variance = %f',
   var_uniform)
```

Scilab code Solution 4.2 Comaparison of True and estimated statics of Gaussian Data

```
// Operating System : Windows XP or later,
// Scilab           : 5.3.3

//here we generate Gaussian distributed data compare
    its statistics with calculated using equation.

//[3] Mean & Variance calculation of Gaussian Data

clc;
clear all;
m = 2;//mean value of gaussian data
sd = 1;//standard deviation
vari = sd^2;
X = grand(10000,1,"nor", m,sd);//gaussian data
   generation using function
```

```
Gaus_true_mean=mean(X)
mprintf('Gaussian True Mean = %f',Gaus_true_mean)
Gaus_true_var=variance(X)
mprintf('\n Gaussian True Variance = %f',Gaus_true_var)

//Mean and variance calcultaion using formula
gs_mean=integrate('x*(1/sqrt(2*%pi*vari)*exp(-(x-m)^2/(2*vari)))','x',0,100)
gs_fsq=integrate('(x^2)*(1/sqrt(2*%pi*vari)*exp(-(x-m)^2/(2*vari)))','x',0,100);
gs_var = gs_fsq - gs_mean.^2;
mprintf('\n Gaussian Calculated Mean = %f',gs_mean)
mprintf('\n Gaussian Calculated Variance = %f',gs_var)
```

Scilab code Solution 4.3 Comaparison of True and estimated statics of Exponential Data

```
// Operating System : Windows XP or later ,
// Scilab           : 5.3.3

//here we generate Exponentially distributed data compare its statistics with calculated using equation.

//[2] Mean and Variance Calculation of Exponential data
clc;
clear all;
lambda = 2;//or lambda = input('enter lemda value for exponential r.v.')//lemda of exponential data;
X = grand(1000,1,"exp", 1/lambda);//Exponential data generation using function
```

```
Expo_true_mean=mean(X)
mprintf('Exponential True Mean = %f',Expo_true_mean)
Expo_true_var=variance(X)
mprintf('\n Exponential True Variance = %f',Expo_true_var)

//Mean and variance calcultaion using formula
ex_mean=integrate('x*(lambda*exp(-lambda*x))','x',0,100)
ex_fsq=integrate('(x^2)*(lambda*exp(-lambda*x))','x',0,100);
ex_var = ex_fsq - ex_mean.^2
mprintf('\n Exponential Calculated Mean = %f',ex_mean)
mprintf('\n Exponential Calculated Variance = %f',ex_var)
```

Experiment: 5

To find density and distribution function of a function of random variable Y = 2X + 1. where X is gaussian R.V.

Scilab code Solution 5.1 Density and Distribution plot generation for one function of Random Variable

```
// Operating System : Windows XP or later,
// Scilab           : 5.3.3

//To find density(pdf) and distribution(cdf)
   function of a function of random variable
//Y = 2X + 1(having form Y = aX + b). where X is
   gaussian R.V.

clc;
clear all;
clf();
mean_x = 1;//mean value of gaussian data
```

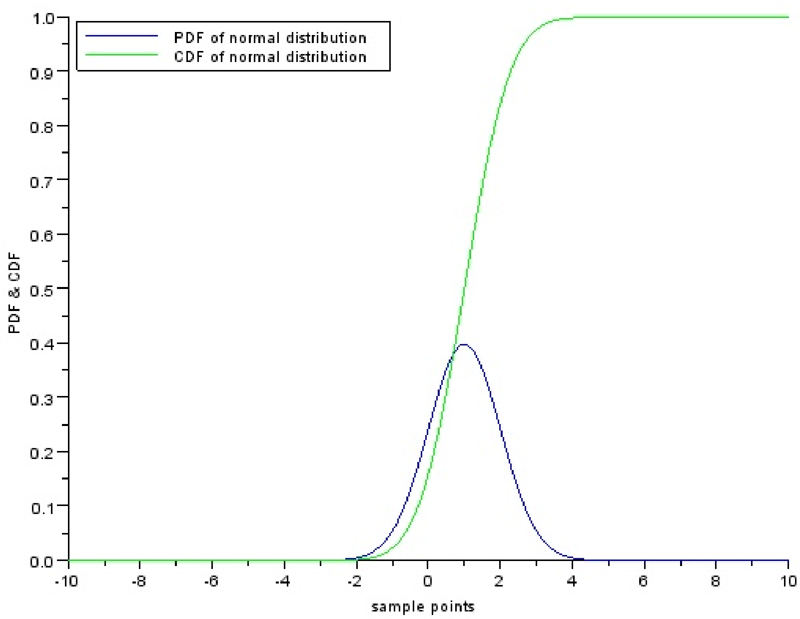

Figure 5.1: Density and Distribution plot generation for one function of Random Variable

```
sd_x = 1;//standard deviation
vari_x = sd_x.^2;
lgd = [];
//PDF and CDF of Gaussian Random Variable X
x = linspace(-10,10,100);
plot2d(x,((1/(sqrt(2*%pi*vari_x)))*exp(-0.5*(x-mean_x).^2/vari_x)),2);//plots pdf of X
set(gca(),"auto_clear","off")
plot2d(x,cdfnor("PQ",x,mean_x*ones(x),sd_x*ones(x)),3);//cdf of gaussian RV X
set(gca(),"auto_clear","on")
xlabel('sample points');
ylabel('PDF & CDF');
//title('density and distribution function for Gaussian function');
legend(['PDF of normal distribution';'CDF of normal distribution'],2);

//PDF and CDF of Y = aX + b where a = 2, b = 1
a = 2;
b = 1;
y = a*x+b;//Function of One Random Variable
mean_y=a*mean_x+b;
vari_y=(a*sd_x).^2;
figure(2,"BackgroundColor",[1,1,1]);
plot2d(y,((1/(sqrt(2*%pi*vari_y*a.^2)))*exp(-0.5*(y-mean_y).^2/vari_y)),2);//pdf of Y
set(gca(),"auto_clear","off")
plot2d(x,cdfnor("PQ",y,(a*mean_x+b)*ones(x),(a*sd_x)*ones(x)),3);//cdf of y
set(gca(),"auto_clear","on")
xlabel('sample points');
ylabel('PDF & CDF of Y = 2X + 1');
legend(['PDF of Y = 2X + 1';'CDF of Y = 2X + 1'],2);


//Assignment :
//1. Perform the operation for function Y = 5X + 1.
```

```
//2. Generate pdf and cdf of nonlinear function
   between Y and X.
```

Experiment: 6

Estimate the mean and variance of Y = 2X + 1, where X is a gaussian random variable.

Scilab code Solution 6.1 Statistics of Function of one random variable

```
// Operating System : Windows XP or later,
// Scilab           : 5.3.3

//True and Estimated value of mean and variance of
   function of one random
//variable having form of Y = aX +b.
clc;
clear all;
rand('seed',getdate('s'))
m = 0;//mean of random variable x
vari = 1;//variance of random variable x
m_est = 0;
var_est = 0;
for i = 1:1000
    y(i,1)= 1 +2*rand(1,1,"normal");// Y = 2X + 1
```

```
where x is gaussian data
    m_est = m_est + ((1/1000)*y(i));//estimation by averaging
    var_est = var_est +((1/1000)*(y(i)-m_est)^2);
end
printf('Estimated mean of Y(=2X + 1) is: Est_mean=%f',m_est)
printf('\n Estimated variance of Y) is: Est_variance=%f',var_est)
//Calculation of true mean of Y
y_mean=integrate('(2*x+1)*(1/sqrt(2*%pi*vari)*exp(-(x-m)^2/(2*vari)))','x',-100,100);
printf('\n True mean of Y(=2X + 1) is: True_mean=%f',y_mean)
//Calculation of true variance of Y
//for a function like Y = aX + b the variance of Y is a^2*Variance of X.
gs_mean=integrate('x*(1/sqrt(2*%pi*vari)*exp(-(x-m)^2/(2*vari)))','x',-50,100);
gs_fsq=integrate('((x^2)*(1/sqrt(2*%pi*vari)*exp(-(x-m)^2/(2*vari))))','x',-50,100);
gs_var = gs_fsq - (gs_mean).^2;
var_y = 2^2*gs_var;//here a = 2
printf('\n True variance of Y(=2X + 1) is: True_variance=%f',var_y)

//Expectation of Y is E(Y)=E(2X+1)=2E(X)+1. That's why answer is 1.
//True variance of Y in this format is equal to a^2*variance of X.
//Assignment:
//1. Assume X is uniform random variable between a to b. find mean and variance.
```

Experiment: 7

Plot Joint density and distribution function of sum of two Gaussian random variable (Z = X + Y).

Scilab code Solution 7.1 Joint Density and Distribution of Function of two random varible

```
// Operating System : Windows XP or later ,
// Scilab           : 5.3.3


//Plot Joint density and distribution function of
   sum of two Gaussian random variable (Z = X + Y).
clc;
clear all;
clf();

```

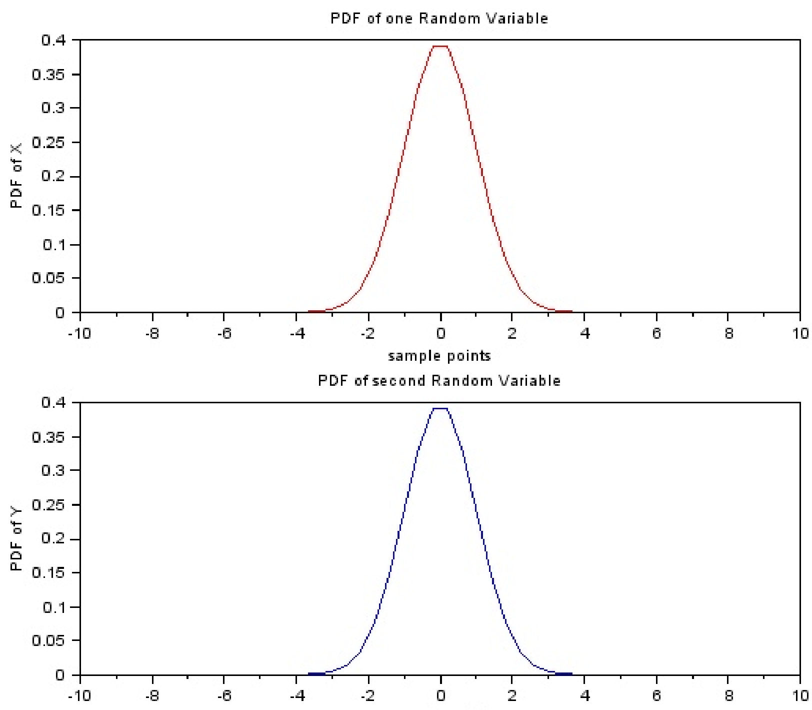

Figure 7.1: Joint Density and Distribution of Function of two random varible

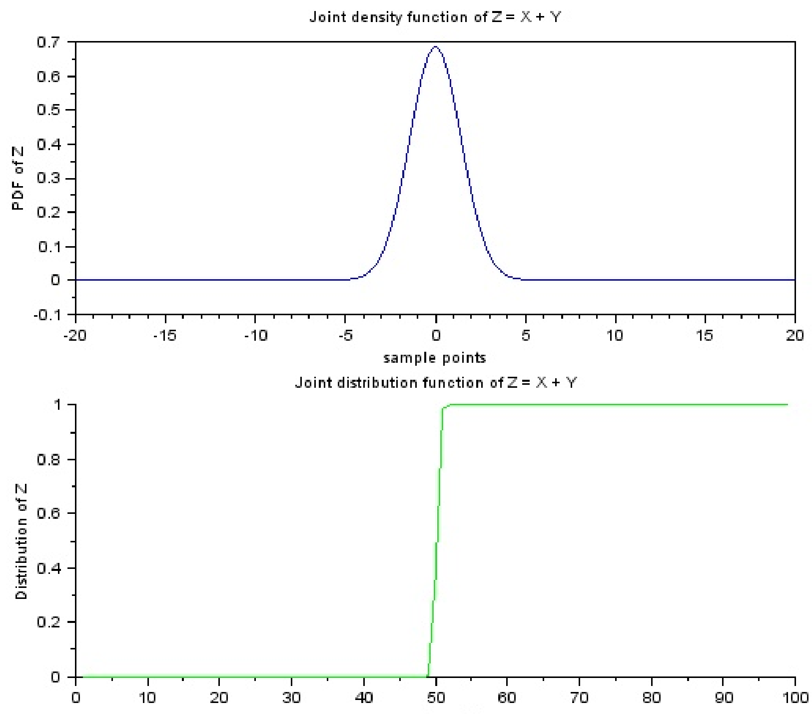

Figure 7.2: Joint Density and Distribution of Function of two random varible

```
//PDF of Gaussian Random Variable X
mean_x = 0;//mean of first gaussian RV
sd_x = 1;//standard deviation of first gausssian RV
vari_x = sd_x.^2;

x = linspace(-10,10,50);//generating linearly spaced data as Random output
X = ((1/(sqrt(2*%pi*vari_x)))*exp(-0.5.*(x-mean_x).^2/vari_x));//finding gaussian pdf of above data
subplot(2,1,1);
plot(x,X,'r')
xlabel('sample points');
ylabel('PDF of X');
title('PDF of one Random Variable')


//PDF of Gaussian Random Variable Y
mean_y = 0;
sd_y = 1;
vari_y = sd_y.^2;

y = linspace(-10,10,50);
Y = ((1/(sqrt(2*%pi*vari_y)))*exp(-0.5.*(y-mean_y).^2/vari_y));
subplot(2,1,2);
plot(y,Y,'b')
xlabel('sample points');
ylabel('PDF of Y');
title('PDF of second Random Variable')

// Joint pdf of sum of random variable X & Y
// When two IID random variable are summen up, their Joint PDF is convolution between individual pdfs of Random variables
z = convol(X,Y);
figure(2,"BackgroundColor",[1,1,1]);
subplot(2,1,1);plot(linspace(-20,20,99),z)//Joint PDF
```

```
xlabel('sample points');
ylabel('PDF of Z');
title('Joint density function of Z = X + Y')
P = cdfnor("PQ",linspace(-20,20,99),mean(z)*ones(z),
    sqrt(variance(z))*ones(z));
set(gca(),"auto_clear","off")
subplot(2,1,2);plot2d(1:length(P),P,3);
xlabel('sample points');
ylabel('Distribution of Z');
title('Joint distribution function of Z = X + Y')
set(gca(),"auto_clear","on")
//Assignment:
//Change mean and variance of individual random
    variable X and Y and observe output
```

Experiment: 8

Estimate the mean and variance of a R.V. Z = X+Y. Where X and Y are also random variables.

Scilab code Solution 8.1 Estimation of mean and variance of sum of two random variable

```
// Operating System : Windows XP or later,
// Scilab            : 5.3.3

//Concept :Estimation of mean and variance of sum of two random variable Z = X + Y, where X and Y are random variable.
// Above concept is explained with example as follows.
//Example: A large circular dartboard is set up with a "bullseye" at the center of the circle,which is at the coordinate(0,0). A dart is thrown at the center but lands at (X,Y) are two different Gaussian random variables. What is average distance of the dart from the bullseye? What is
```

```
variance of data?
//Distance from center is given as sqrt(X^2+Y^2)

clc;
clear all;
rand('seed',0)//setting seed of random generator to zero
m_est = 0;
for i = 1:1000
    R(i,1)=sqrt(rand(1,1,'normal')^2+rand(1,1,'normal')^2);//calculation of distance from center
    m_est=m_est+(1/1000)*R(i);//estimation of mean from data
end
m_est
mprintf('Mean of Sum of Two Random variable that is Mean of Z = %f',m_est)
v_est = variance(R)//variance calculation
mprintf('\n Variance of Sum of Two Random variable that is Mean of Z = %f',v_est)
```

Experiment: 9

Simulation of Central Limit Theorem.

Scilab code Solution 9.1 Summation of two random variable leads to Gaussian density function

```
// Operating System : Windows XP or later ,
// Scilab           : 5.3.3


//Simulation of Central Limit Theorem.
//(Which says that if we keep on adding independant
   Random Variables then it density function
//approches to gaussian distribution)
//here two uniform RVs are added.

clc;
clear all;
clf();
n = 0:0.01:1;
x = zeros(length(n),1);
i = 1:50;//length of Uniform rv 1
```

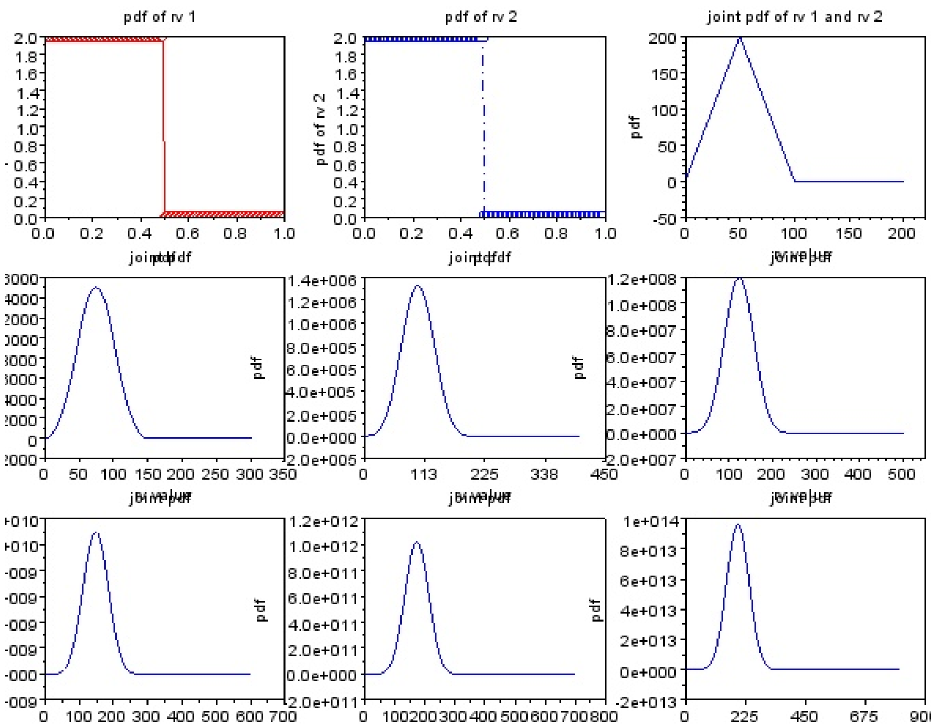

Figure 9.1: Summation of two random variable leads to Gaussian density function

```
x(i)= 2;
subplot(3,3,1);
plot(n,x,'r-d')
xlabel('pdf');ylabel('pdf of rv 1')
title('pdf of rv 1 ');

y = zeros(length(n),1);
j = 1:50;
y(j)= 1*2;//length of Uniform rv 2
subplot(3,3,2);
plot(n,y,'bo-.')
xlabel('pdf');
ylabel('pdf of rv 2')
title('pdf of rv 2');

z1 = convol(x,y);
subplot(3,3,3)
//When two independent RVs are added their joint density is convolution of marginal density
plot(z1)
xlabel('rv value');
ylabel('pdf')
title('joint pdf of rv 1 and rv 2');

for i = 4:9 // adding rv 9 times
    subplot(3,3,i)
    z1 = convol(z1,y);
    plot(z1)
    xlabel('rv value');
    ylabel('pdf')
title('joint pdf');
end
```