
Table of Contents

Book Problems	1
Monte Carlo Method	2
a) Random Coordinates	2
b) Check under Gaussian	2
c) Fraction under Gaussian	2
d) Area under Gaussian	2
e) Check with CDF	5

Book Problems

```
% 4.5: 14
% a
mu = 0.652;
std = 0.003;
mean = 0.65;
err = 0.005;
zscore = [(mean+err-mu)/std, (mean-err-mu)/std];
proportionA = 100*(normcdf(zscore(1))-normcdf(zscore(2)));
% b
mu = 0.65;
zscore = [(mean+err-mu)/std, (mean-err-mu)/std];
proportionB = 100* (normcdf(zscore(1))-normcdf(zscore(2)));
% c
proportionC = 99/100;
stdC = -err/(norminv((1-proportionC)/2));

% 4.11: 4
% a
mean = 2000;
std = 500;
n = 625;
mu = 1980;
pA = normcdf((mean-mu)/std);
%b
n = 0.1*n;
zscore=(n-60)/(625*0.1*0.9);
pB = normcdf(zscore);

% 4.11: 8
% a
mean = 30.01;
std = 0.1;
n = 50;
pA=normcdf((mean-1500/n)/(std/sqrt(n)));
% b
n = 80;
pB=normcdf(-(mean-2401/n)/(std/sqrt(n)));
% c
```

```

volC = (mean+norminv(0.90)*std/sqrt(n))*n;

% 4.11: 16
mean = 40;
std = 5;
n = 100;
% a
pA = normcdf(-(40-36.7)/(std));
% b: Kind of, 75% of the times is greater.
% c: No, standard deviation away.
% d
pD = normcdf(-(40-39.8)/(std));
% e: No, 51% of the time is greater.
% f: Yes, close to 40.

% 4.11: 18
mean1 = 0.5;
std1 = 0.4;
mean2 = 0.6;
std2 = 0.5;
n = 100;
% a
pA = normcdf((mean1-55/n)/(std1/sqrt(n)));
% b
pB = normcdf(-(mean2-55/n)/(std2/sqrt(n)));
% c
pC = normcdf((mean1+mean2-115/100)/((std1+std2)/sqrt(n)));
% d
pD =
    sum(sum(normrnd(mean1,std1,n,n^2))>sum(normrnd(mean2,std2,n,n^2)))/
    n^2;

```

Monte Carlo Method

a) Random Coordinates

```

coords = @(n) [-1.3*rand(1,n,'double')+0.5; rand(1,n,'double')];

```

b) Check under Gaussian

```

underG = @(x,y) y < 1/(sqrt(2*pi))*exp(-x.^2/2);

```

c) Fraction under Gaussian

```

P = @(x,y,n) sum(underG(x,y))/n;

```

d) Area under Gaussian

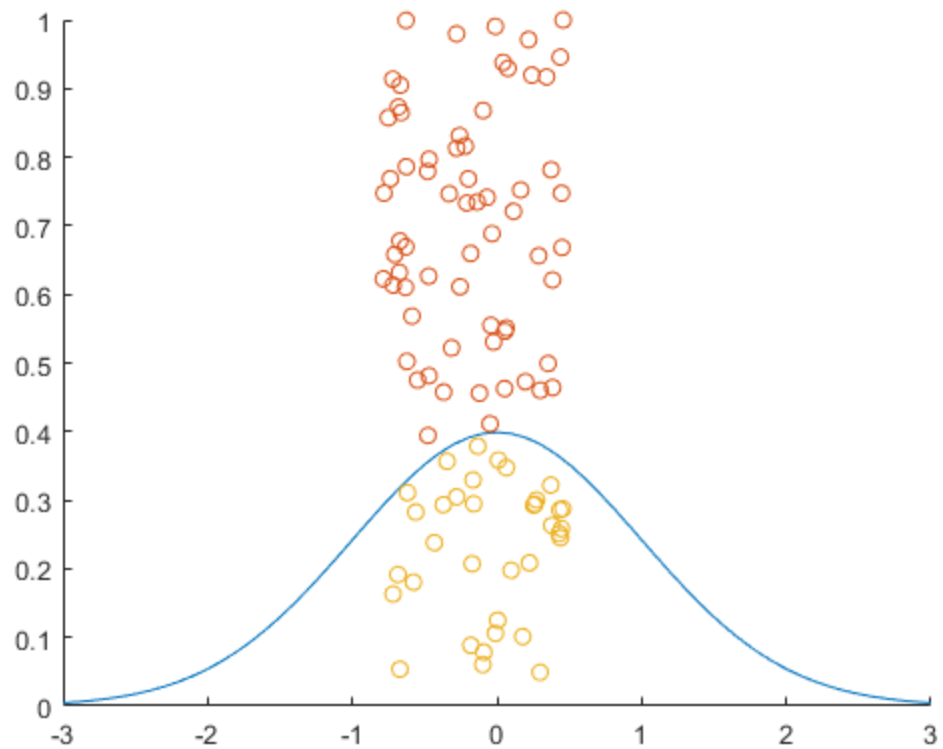
```

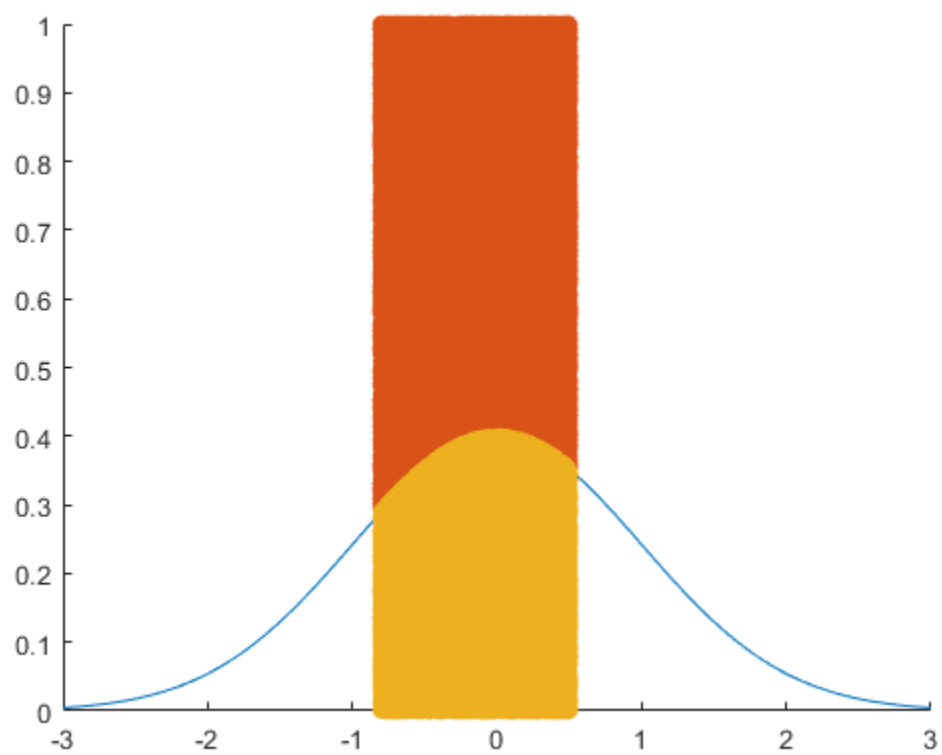
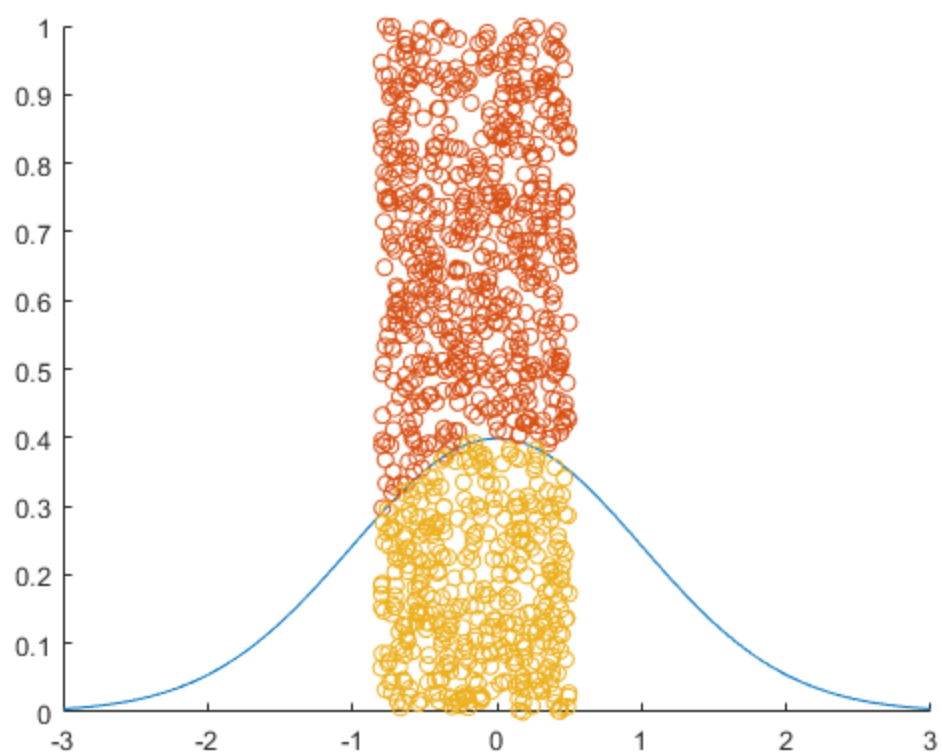
grid = 1*1.3;

```

```
n = [100 1000 100000];
G = zeros(1,3);

for i = 1:length(n)
    figure
    x = deal(coords(n(i)));
    y = x(2,:);
    x = x(1,:);
    G(i) = P(x,y,n(i))*grid;
    yU = y.*underG(x,y);
    yU(yU==0)=NaN;
    hold on
    plot(-3:.1:3,normpdf(-3:.1:3,0,1))
    plot(x,y,'o')
    plot(x,yU,'o')
    hold off
end
```





e) Check with CDF

```
realG = cdf('Normal',0.5,0,1)-cdf('Normal',-0.8,0,1);  
accuracy = G./realG;
```

Published with MATLAB® R2016a