Simulation

1 Simulation

How to generate random samples

rand, randn, randi, randperm

Kernel	Description
`rand`	$U(0,1)$
`randn`	$\mathcal{N}(0,1)$
`randi`	Uniform on $\{1,2,\cdots,\mathtt{N}\}$
`randperm`	Random permutation on $\{1,2,\cdots,\mathtt{N}\}$

random

```
x      =  random  ('Dist',  Pa1*ones(n,m),  Pa2*ones(n,m))
↑
```

$n \times m$ samples

Use `>> help random` for detailed distribution name and corresponding parameters.

Fine prints of random number generator

		Description
`rng`	`'default'`	Reset RNG fresh
	`n`	Reset RNG with seed `n`

Example - Coin flip using $U(0,1)$

`rand(n,m)` generates `n-by-m` matrix of uniform samples from $[0,1]$. Generate 5 uniform samples U_i from $[0,1]$ and using this sample with logical indexing generate 5 Bernoulli samples B_i with success rate $p = 0.49$, i.e, $P(B_i = 1) = 0.49$ and $P(B_i = 0) = 1 - 0.49$.

```
%% Generation of n Bernoulli samples using n uniform samples %%%%%%%%%%%%

clear all; close all; clc; rng('default')

n = 5; p = 0.499;

% n uniform samples
U = rand(1,n)

% n Bernoulli samples B(p)
B = U; B(U<=1-p) = 0; B(U>1-p) = 1

%% Output %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

U =

    0.8147    0.9058    0.1270    0.9134    0.6324


B =

     1     1     0     1     1

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

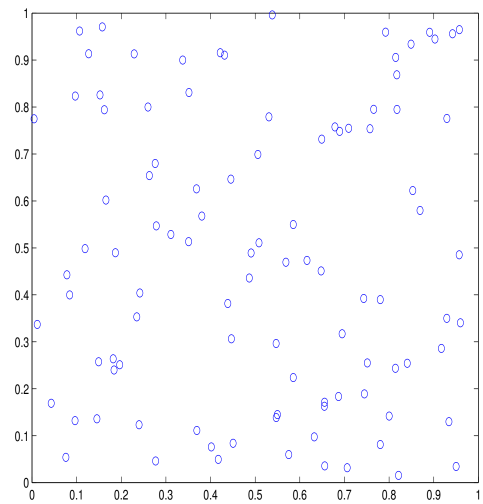


Figure 1: 100 random points in $[0, 1]^2$.

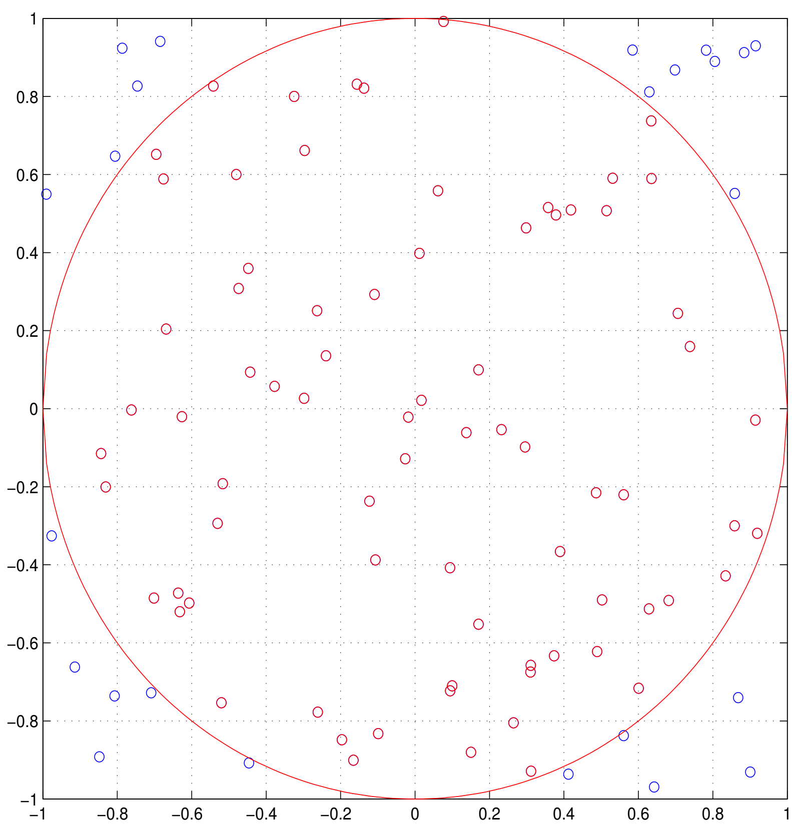
```
clear all; close all; clc; rng('default')

n=100;
x=rand(2,n);

plot(x(1,:),x(2,:),'o')
```

Example - Points inside the unit circle

Generate 100 uniform points on $[-1,1]^2$. Color the points inside the unit circle $x^2+y^2=1$ red and the points outside blue. Draw the unit circle with red color at the same time.




```
clear all; close all; clc; rng('default')

n = 100; x = 2*rand(2,n)-1; plot(x(1,:),x(2,:),'o'); grid on; hold on

r2 = x(1,:).^2+x(2,:).^2; i = find(r2<=1); plot(x(1,i),x(2,i),'or')

xp = -1:0.01:1; yp =sqrt(1-xp.^2); plot(xp,yp,'-r',xp,-yp,'-r')
```

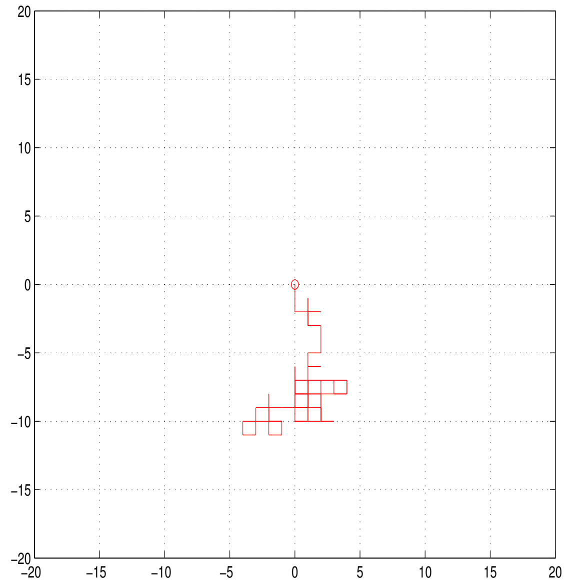


Figure 2: 2D simple random walk.

```
clear all; close all; clc; rng('default')

m=100;
coin=rand(m,1);

increment=zeros(m,2);
increment(coin<0.25,1)=1; increment(coin<0.25,2)=0;
increment(0.25<=coin&coin<0.5,1)=-1; increment(0.25<=coin&coin<0.5,2)=0;
increment(0.5<=coin&coin<0.75,1)=0; increment(0.5<=coin&coin<0.75,2)=1;
increment(0.75<coin,1)=0; increment(0.75<coin,2)=-1;

walk=cumsum(increment);
walk=[0 0; walk];
r=max(max(abs(walk)))+1;

plot(0,0,'or'); grid on; hold on;
axis([-r r -r r])
for k=1:m
    plot([walk(k,1) walk(k+1,1)],[walk(k,2) walk(k+1,2)],'-r')
    pause(0.1)
end
```

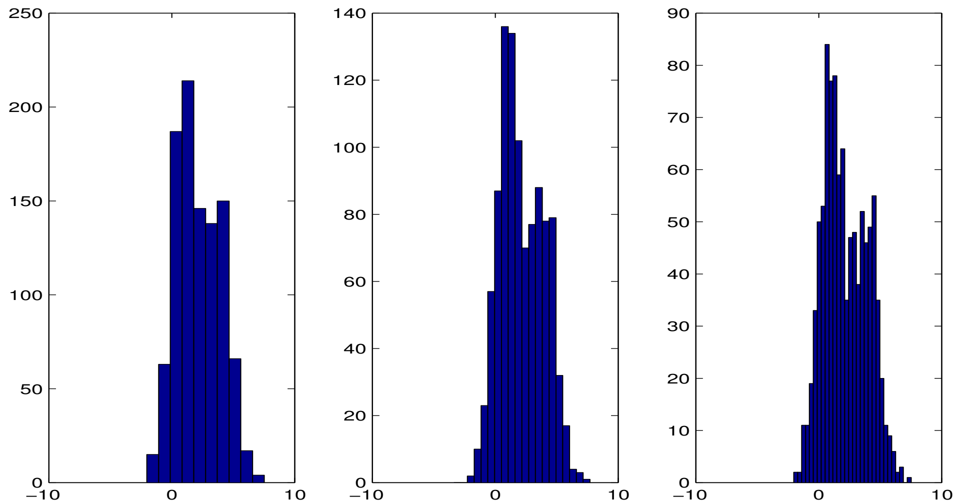


Figure 3: Left is histogram of 600 samples from $\mathcal{N}(1,1)$, Center is histogram of 400 samples from $\mathcal{N}(4,1)$, Right is histogram of combined 1000 samples from two different normal distributions.

```
clear all; close all; clc; rng('default')

% n1 samples from N(mu1,si1)
n1=600;
mu1=1; si1=1;
x1=mu1+si1*randn(1,n1);
subplot(131)
hist(x1)

% n2 samples from N(mu2,si2)
n2=400;
mu2=4; si2=1;
x2=mu2+si1*randn(1,n2);
subplot(132)
hist(x2)

% Samples from two different normal distributions
x=[x1 x2];
subplot(133)
hist(x)
```

Longest run

If one flips a coin n times, what is the probability distribution of the longest "run" (the length of a sequence of consecutive heads or tails) which will occur?

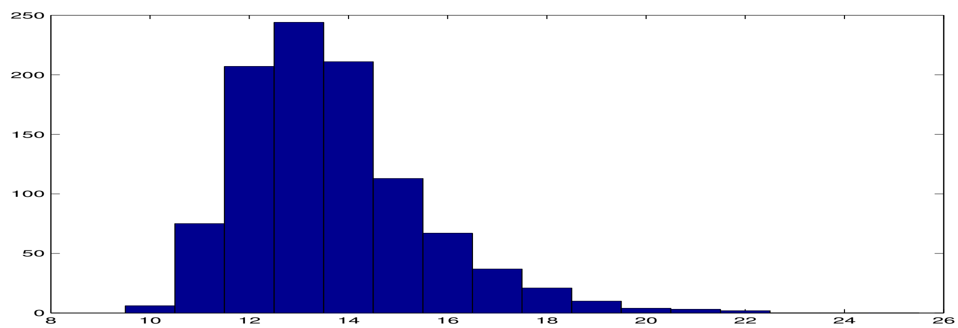


Figure 4: Histogram of 1000 runs of 100000 coin flips.

```
clear all; close all; clc; rng('default')

p=0.5; n=10000; % We flip a fair coin n times
NumSimu=1000; % We do this experiment NumSimu times
x=random('Binomial',1*ones(NumSimu,n),p*ones(NumSimu,n));

Run=zeros(NumSimu,1);
for NumS=1:NumSimu
    Current_Run=1;
    Overall_Run=1;
    for i=2:n
        if x(NumS,i)==x(NumS,i-1)
            Current_Run=Current_Run+1;
            Overall_Run=max(Current_Run,Overall_Run);
        else
            Current_Run=1;
        end
    end
    Run(NumS,1)=Overall_Run;
end

hist(Run)
```

Arcsine law - Last visit time

Starting from the origin run a simple random walk $S_0 = 0, S_1, S_2, \ldots, S_{2n}$ up to time $2n$ on $\mathbb{Z}$. Let L_{2n} be the last visit time to the origin. Then, for $0 \leq a < b \leq 1$

$$\mathbb{P}\left(a \leq \frac{L_{2n}}{2n} \leq b\right) \quad \rightarrow \quad \int_a^b \frac{1}{\pi} \cdot \frac{1}{\sqrt{x(1-x)}} dx = \frac{2}{\pi}\left[\arcsin(\sqrt{b}) - \arcsin(\sqrt{a})\right]$$

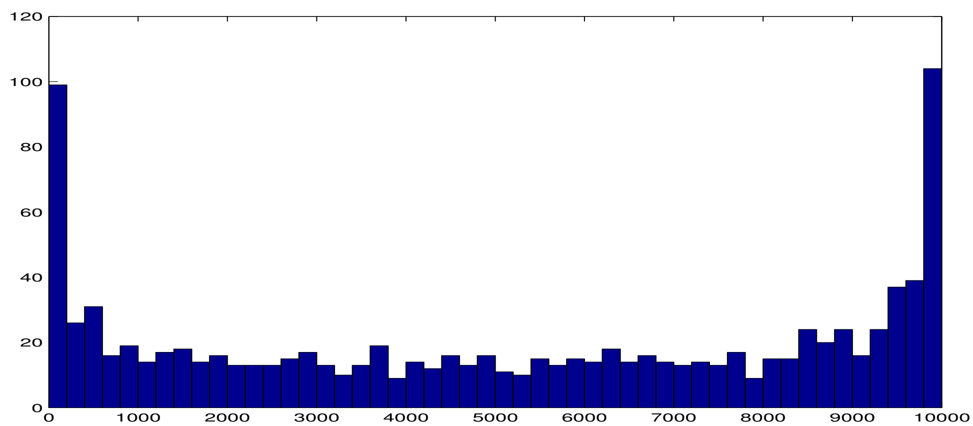


Figure 5: Histogram of 1000 last visit time of 10000 fair coin flips.

```
clear all; close all; clc; rng('default')

% Parameters
p=0.5; n=10000; % We flip a fair coin n times
NumSimu=1000; % We do this experiment NumSimu times
x=random('Binomial',1*ones(NumSimu,n),p*ones(NumSimu,n));
x=2*x-1;

Last_Visit_Time=zeros(NumSimu,1);
for NumS=1:NumSimu
    Sn=cumsum(x(NumS,:));
    Random_Walk=[0 Sn];
    Last_Visit_Time(NumS,1)=find(Random_Walk==0,1,'last')-1;
end

hist(Last_Visit_Time)
```

Arcsine law - Number of positive sticks

Starting from the origin run a simple random walk $S_0 = 0, S_1, S_2, \dots, S_{2n}$ up to time $2n$ on $\mathbb{Z}$. Count the number of positive sticks $(k-1, S_{k-1}) \to (k, S_k)$ whose center lie above the x axis and let N_{2n} be the number of positive sticks. Then, for $0 \le a < b \le 1$

$$\mathbb{P}\left(a \le \frac{N_{2n}}{2n} \le b\right) \quad \to \quad \int_a^b \frac{1}{\pi} \cdot \frac{1}{\sqrt{x(1-x)}} dx = \frac{2}{\pi}\left[\arcsin(\sqrt{b}) - \arcsin(\sqrt{a})\right]$$

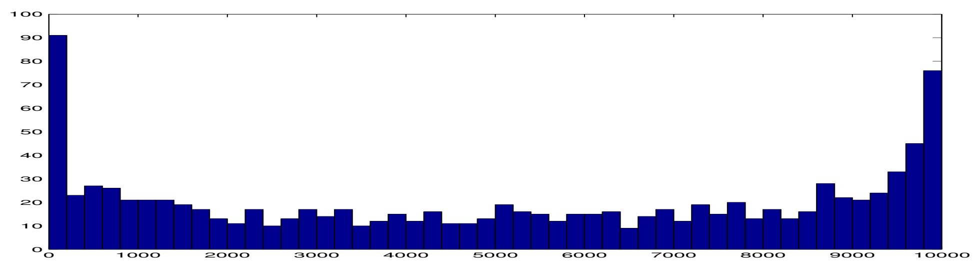


Figure 6: Histogram of 1000 number of positive side sticks of 10000 fair coin flips.

```
%% Number of positive sticks
clear all; close all; clc; rng('default')

% Parameters
p=0.5; n=10000; % We flip a fair coin n times
NumSimu=1000; % We do this experiment NumSimu times
x=random('Binomial',1*ones(NumSimu,n),p*ones(NumSimu,n));
x=2*x-1;

Number_of_Positive_Sticks=zeros(NumSimu,1);
for NumS=1:NumSimu
    Sn=cumsum(x(NumS,:));
    Random_Walk=[0 Sn];
    Center_of_Stick=(Random_Walk(1:end-1)+Random_Walk(2:end))/2;
    Positive_Side_Sticks=find(Center_of_Stick>0);
    Number_of_Positive_Sticks(NumS,1)=length(Positive_Side_Sticks);
end

hist(Number_of_Positive_Sticks)
```