Marwin B. Alejo 2020-20221 EE214_Module4-LabEx1

Table of Contents

*

I. Simulation of binomial counting process

1.a. Generate a single realization of the binomial sum process and determine n=1000, and observe how the sum process behaves as the probability of success is varied from p=0.1, 0.2, 0.4, and 0.6. Does the simulation agree with the theoretical?

(w) p=0.1

```
bernoulli = (rand(1,1000)<=0.1);
sum(bernoulli(:)==1)
```

```
ans =

    93
```

(x) p=0.2

```
bernoulli = (rand(1,1000)<=0.2);
sum(bernoulli(:)==1)
```

```
ans =

   195
```

(y) p=0.4

```
bernoulli = (rand(1,1000)<=0.4);
sum(bernoulli(:)==1)
```

```
ans =

   400
```

(z) p=0.6

```
bernoulli = (rand(1,1000)<=0.6);
sum(bernoulli(:)==1)
```

```
ans =

   603
```

Theory: The binomial process is a random counting system where the n identical trials (aka. Bernoulli trials), with each having the success probability 'p' and a failure rate of '1-p'. In this case, the `bernoulli` variable contain the Bernoulli trial of `'n'` samples for one realization and return a random 1/0 bits depeding in the given 'p'.

Simulation: The generated sum of success for the simulation of this cases agreed with that of the described theory above. Given case (w) with p=0.1 yielded an ~100 1-bit or 10% of the n samples is '1'. The same observation happened on (x)p=0.2, (y)p=0.4, and (z)p=0.6. Overall, the simulation agree with the theory although the concept of 10%, 20%, 40%, and 60% is not strictly followed by the used simulation algorithm in generating random bits.

1.b. Generate any realization of the binomial sum process starting with k=10, each realization has n=1000 samples. Calculate the mean and variance of the process and compare with the theoretical.

(w) p=0.1

```
for i = 1:10
    bernoulli = (rand(1,1000)<=0.1);
    binomial(i) = sum(bernoulli);
end
```

Mean of p=0.1

```
mean(binomial)-10*0.1
```

```
ans =

  593.6880
```

Variance of p=0.1

```
var(binomial)-10*0.1*(1-0.1)
```

```
ans =

   2.6331e+03
```

(x) p=0.2

```
for i = 1:10
    bernoulli = (rand(1,1000)<=0.2);
    binomial(i) = sum(bernoulli);
end
```

Mean of p=0.2

```
mean(binomial)-10*0.2
```

```
ans =

  593.6570
```

Variance of p=0.2

```
var(binomial)-10*0.2*(1-0.2)
```

```
ans =

   1.7794e+03
```

(y) p=0.4

```
for i = 1:10
    bernoulli = (rand(1,1000)<=0.4);
    binomial(i) = sum(bernoulli);
end
```

Mean of p=0.4

```
mean(binomial)-10*0.4
```

```
ans =

  593.6370
```

Variance of p=0.4

```
var(binomial)-10*0.4*(1-0.4)
```

```
ans =

  609.9516
```

(z) p=0.6

```
for i = 1:10
    bernoulli = (rand(1,1000)<=0.6);
    binomial(i) = sum(bernoulli);
end
```

Mean of p=0.6

```
mean(binomial)-10*0.6
```

```
ans =

  593.6470
```

Variance of p=0.6

```
var(binomial)-10*0.6*(1-0.6)
```

```
ans =

  219.0118
```

Theory: The theory behind these cases are similar to that described in 1a above except that there are 10 realizations in each case. The mean and variance of each case is computed using the standard mean and variance formula and with respect to 'p' and '1-p'.

Simulation: The generated mean and variance through simulation of the 10 realizations differ by a factor of decimal as compared with that of theoretically computed. Nevertheless, if both the mean and varaince computed manually and through simulation are rounded-off to the nearest integer, both yield the same value. Overall, it can be infer from these observations that the simulation still agree with the theory.

1.c. Compare the mean and variance with k=100 and k=1000. Does the simulation agree with the theoretical.

(w) p=0.1 k=100

```
for i = 1:100
    bernoulli = (rand(1,1000)<=0.1);
    binomial(i) = sum(bernoulli);
end
```

Mean of p=0.1

```
mean(binomial)-100*0.1
```

```
ans =

  539.6210
```

Variance of p=0.1

```
var(binomial)-100*0.1*(1-0.1)
```

```
ans =

   2.2750e+04
```

(x) p=0.2 k=100

```
for i = 1:100
    bernoulli = (rand(1,1000)<=0.2);
    binomial(i) = sum(bernoulli);
end
```

Mean of p=0.2

```
mean(binomial)-100*0.2
```

```
ans =

  539.6160
```

Variance of p=0.2

```
var(binomial)-100*0.2*(1-0.2)
```

```
ans =

   1.4636e+04
```

(y) p=0.4 k=100

```
for i = 1:100
    bernoulli = (rand(1,1000)<=0.4);
    binomial(i) = sum(bernoulli);
end
```

Mean of p=0.4

```
mean(binomial)-100*0.4
```

```
ans =

  539.5140
```

Variance of p=0.4

```
var(binomial)-100*0.4*(1-0.4)
```

```
ans =

   3.8530e+03
```

(z) p=0.6 k=100

```
for i = 1:100
    bernoulli = (rand(1,1000)<=0.6);
    binomial(i) = sum(bernoulli);
end
```

Mean of p=0.6

```
mean(binomial)-100*0.6
```

```
ans =
```

```
  539.9440
```

Variance of p=0.6

```
var(binomial)-100*0.6*(1-0.6)
```

```
ans =

  197.8307
```

(w) p=0.1 k=1000

```
for i = 1:1000
    bernoulli = (rand(1,1000)<=0.1);
    binomial(i) = sum(bernoulli);
end
```

Mean of p=0.1

```
mean(binomial)-1000*0.1
```

```
ans =

   -0.5430
```

Variance of p=0.1

```
var(binomial)-1000*0.1*(1-0.1)
```

```
ans =

   12.8810
```

(x) p=0.2 k=1000

```
for i = 1:1000
    bernoulli = (rand(1,1000)<=0.2);
    binomial(i) = sum(bernoulli);
end
```

Mean of p=0.2

```
mean(binomial)-1000*0.2
```

```
ans =

    0.4390
```

Variance of p=0.2

```
var(binomial)-1000*0.2*(1-0.2)
```

```
ans =

   -6.4101
```

(y) p=0.4 k=1000

```
for i = 1:1000
    bernoulli = (rand(1,1000)<=0.4);
    binomial(i) = sum(bernoulli);
end
```

Mean of p=0.4

```
mean(binomial)-1000*0.4
```

```
ans =

    0.2610
```

Variance of p=0.4

```
var(binomial)-1000*0.4*(1-0.4)
```

```
ans =

   -9.9851
```

(z) p=0.6 k=1000

```
for i = 1:1000
    bernoulli = (rand(1,1000)<=0.6);
    binomial(i) = sum(bernoulli);
end
```

Mean of p=0.6

```
mean(binomial)-1000*0.6
```

```
ans =

   -0.1000
```

Variance of p=0.6

```
var(binomial)-1000*0.6*(1-0.6)
```

```
ans =

   -8.4925
```

Given k=100, it is observed that the means of all cases (w:z) are closely related to that k=10 in terms of units while the means of k=1000 shrank largely considering that the sample size is 1000 and the process have to fit 1000 realizations in it. It is also observable that the variance shrank inverse-exponentially as the number of realization increases per case to the point that it yield negative variances on k=1000. Moreover, mean/realization overlap each other as realization increases in size.

Discussion: Given the above observations about Binomial counting or distribution, it can be infer that the probability of success shrink as realization k increases in size and the possibility of happening the other realizations with their respective mean and outcome become hypothetical/imaginary (negative).

II. Simulate a symmetric random walk process

a. What happens when n=[100 1000 10000] becomes larger?

```
N=[100 1000 10000];

% N=100
randn('state',N(1));          % set the state of randn
T = 1; dt = T/N(1);
dW = sqrt(dt)*randn(1,N(1));   % increments
W1 = cumsum(dW);              % cumulative sum
figure; plot([0:dt:T],[0,W1],'r-'); title('Fig.1: Wiener Process N=100');  %
 plot W against t
xlabel('t','FontSize',16)
ylabel('W(t)','FontSize',16,'Rotation',0)

% N=1000
randn('state',N(2));          % set the state of randn
T = 1; dt = T/N(2);
dW = sqrt(dt)*randn(1,N(2));   % increments
W2 = cumsum(dW);              % cumulative sum
figure; plot([0:dt:T],[0,W2],'r-'); title('Fig.2: Wiener Process N=1k');   %
 plot W against t
xlabel('t','FontSize',16)
ylabel('W(t)','FontSize',16,'Rotation',0)

% N=10000
randn('state',N(3));          % set the state of randn
T = 1; dt = T/N(3);
dW = sqrt(dt)*randn(1,N(3));   % increments
W3 = cumsum(dW);              % cumulative sum
figure; plot([0:dt:T],[0,W3],'r-'); title('Fig.3: Wiener Process N=10k');   %
 plot W against t
xlabel('t','FontSize',16)
ylabel('W(t)','FontSize',16,'Rotation',0)
```

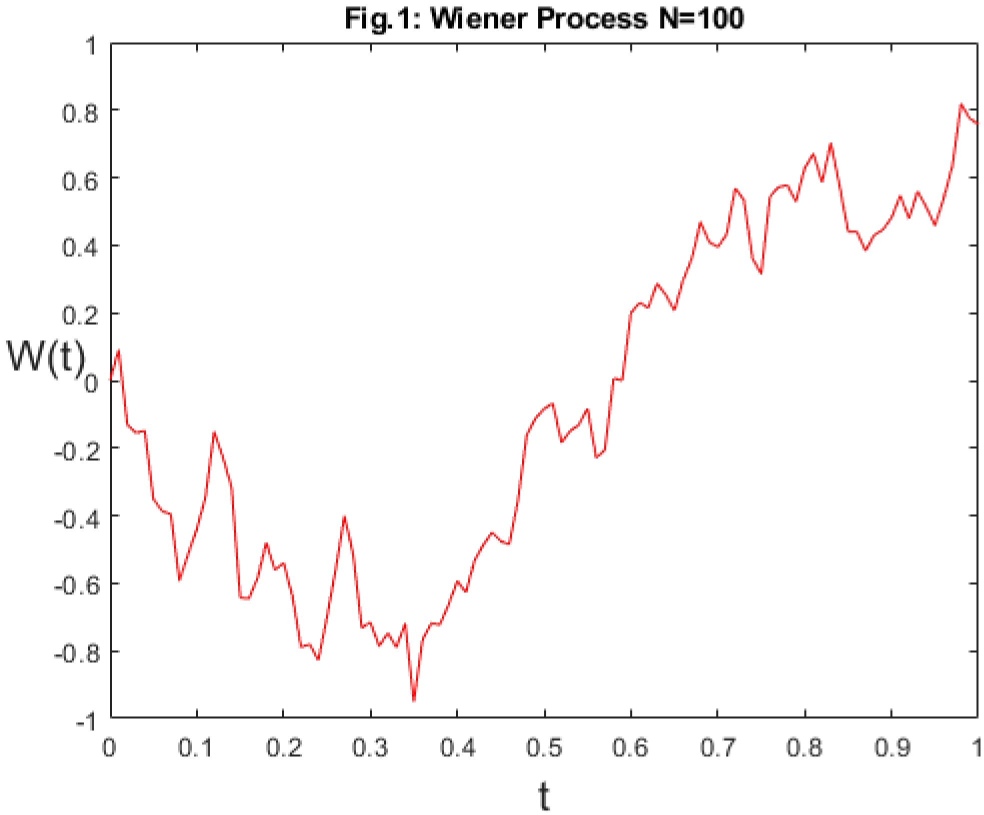


Fig.1: Wiener Process N=100
W(t)
t
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

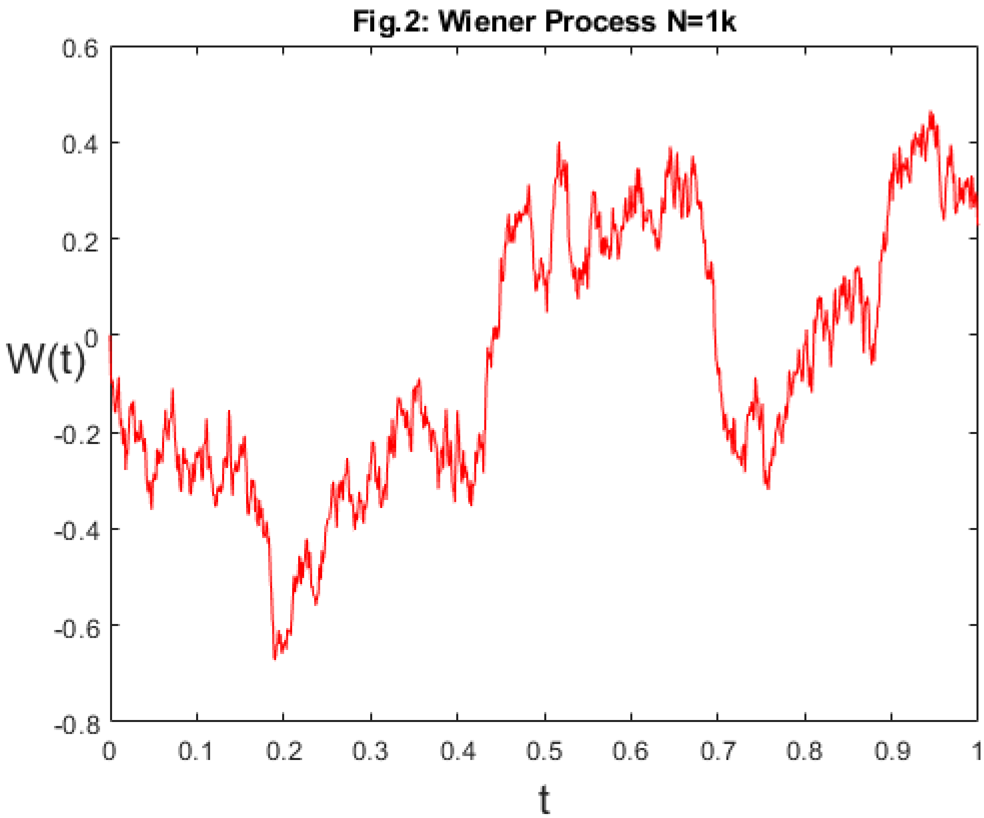
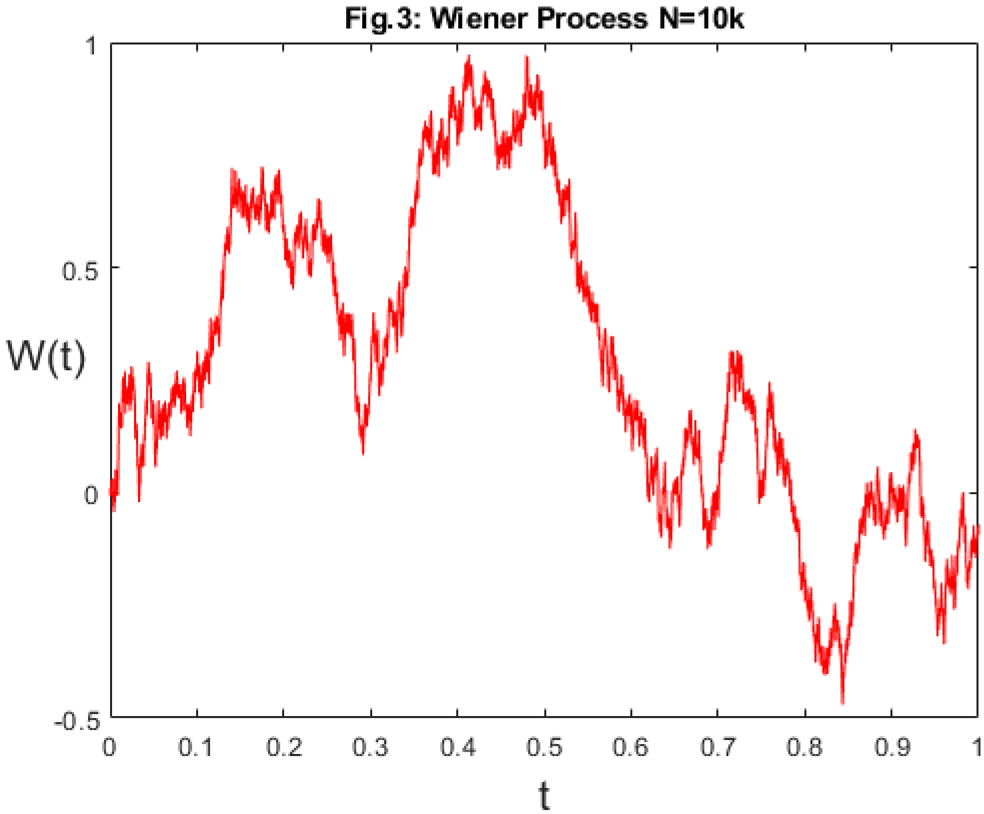


Fig.2: Wiener Process N=1k
W(t)
t
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1



Figures 1-3 above shown the generated random walks given n=[100 1000 10000]. Also, when n becomes larger, random walk becomes denser and finer as well.

b. Calculate the mean and variance of the process, for each value n. Does the simulated mean and varaince agree with the theoretical?

Mean of W1

```
mean(W1)
% Uncomment the code below to see proof that W1 agree with theoretical.
% W1-mean(W1,1)
```

```
ans =

   -0.0680
```

Variance of W1

```
var(W1)
```

```
ans =

    0.2593
```

Mean of W2

```
mean(W2)
% Uncomment the code below to see proof that W2 agree with theoretical.
% W2-mean(W2,1)
```

```
ans =

   -0.0515
```

Variance of W2

```
var(W2)
```

```
ans =

    0.0748
```

Mean of W3

```
mean(W3)
% Uncomment the code below to see proof that W3 agree with theoretical.
% W3-mean(W3,1)
```

```
ans =

    0.2881
```

Variance of W3

```
var(W3)
```

```
ans =

    0.1199
```

Observation: The mean of the simulated random walk with n value [100 1000 10000] agreed with that of the theoretical with mean is defined be equal to '0'. The variance on the other hand does not agree with that of theoretical with it being defined to be equal to '1' which is in contrary of the generated variances of the random walks.

c. What is the mean E[X[n]] as a function of n?

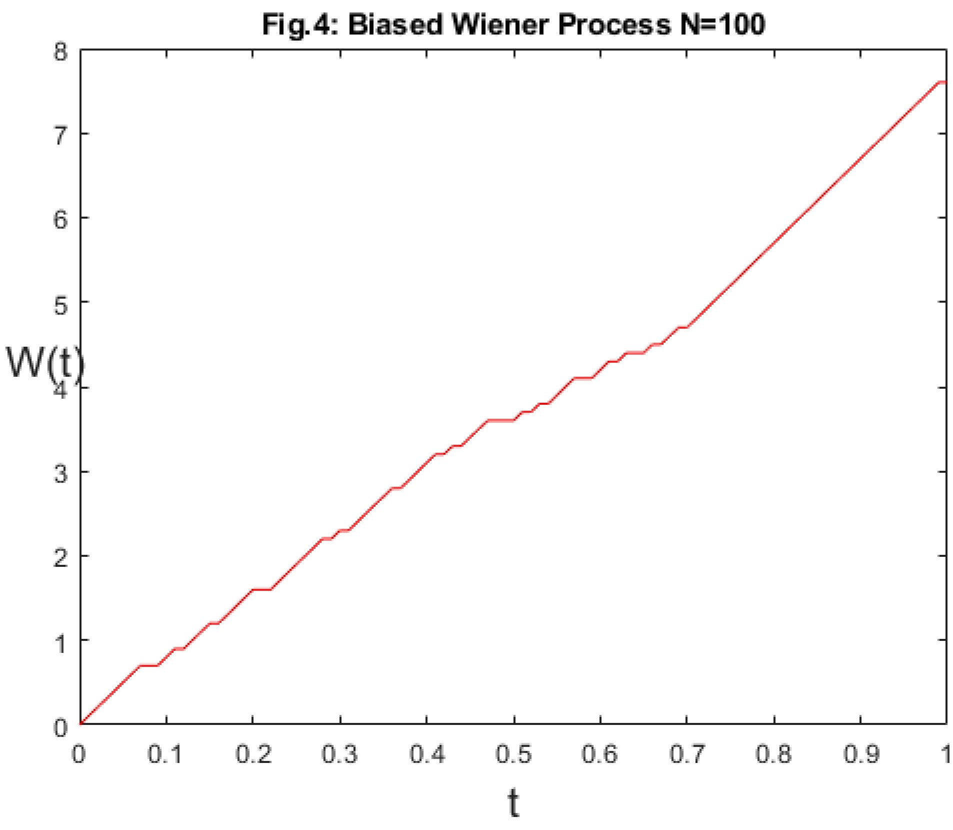
```
N=[100 1000 10000];

% N=100
randn('state',N(1));          % set the state of randn
T = 1; dt = T/N(1);
```

```
dW = sqrt(dt)*(rand(1,N(1))<=0.75);   % increments
W1 = cumsum(dW);                      % cumulative sum
figure; plot([0:dt:T],[0,W1],'r-'); title('Fig.4: Biased Wiener Process
 N=100');  % plot W against t
xlabel('t','FontSize',16)
ylabel('W(t)','FontSize',16,'Rotation',0)

% N=1000
randn('state',N(2));          % set the state of randn
T = 1; dt = T/N(2);
dW = sqrt(dt)*(rand(1,N(2))<=0.75);   % increments
W2 = cumsum(dW);                      % cumulative sum
figure; plot([0:dt:T],[0,W2],'r-'); title('Fig.5: Biased Wiener Process
 N=1k');   % plot W against t
xlabel('t','FontSize',16)
ylabel('W(t)','FontSize',16,'Rotation',0)

% N=10000
randn('state',N(3));          % set the state of randn
T = 1; dt = T/N(3);
dW = sqrt(dt)*(rand(1,N(3))<=0.75);   % increments
W3 = cumsum(dW);                      % cumulative sum
figure; plot([0:dt:T],[0,W3],'r-'); title('Fig.6: Biased Wiener Process
 N=10k');   % plot W against t
xlabel('t','FontSize',16)
ylabel('W(t)','FontSize',16,'Rotation',0)
```



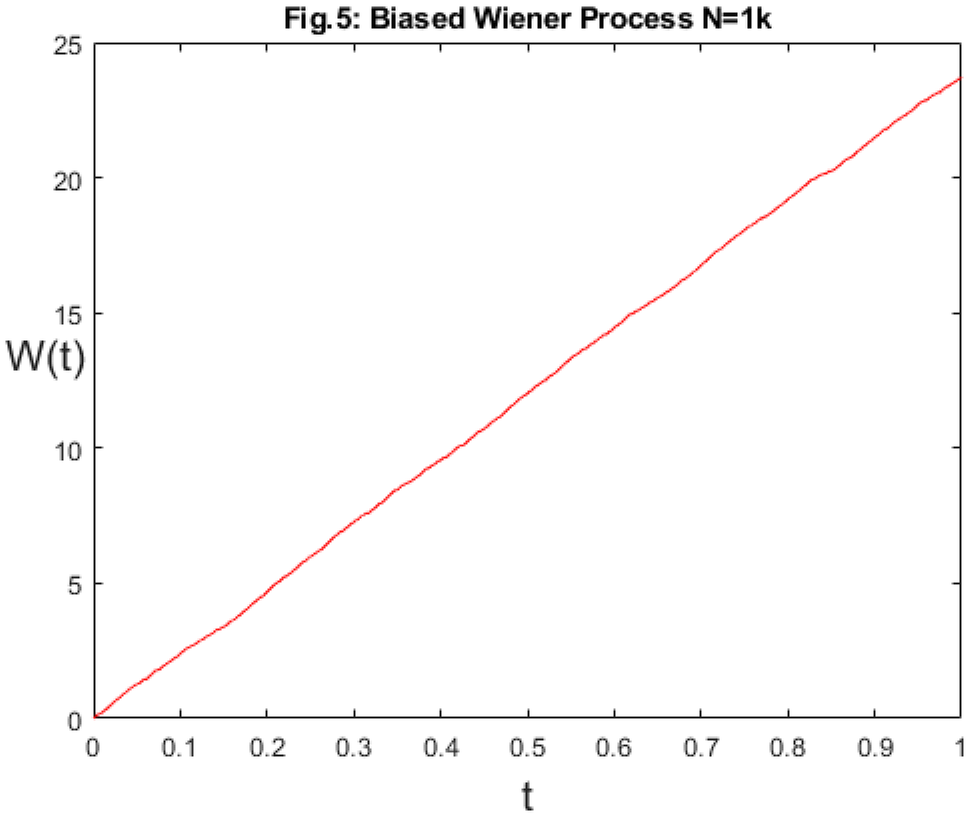


Fig.5: Biased Wiener Process N=1k
W(t)
t
0
5
10
15
20
25
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

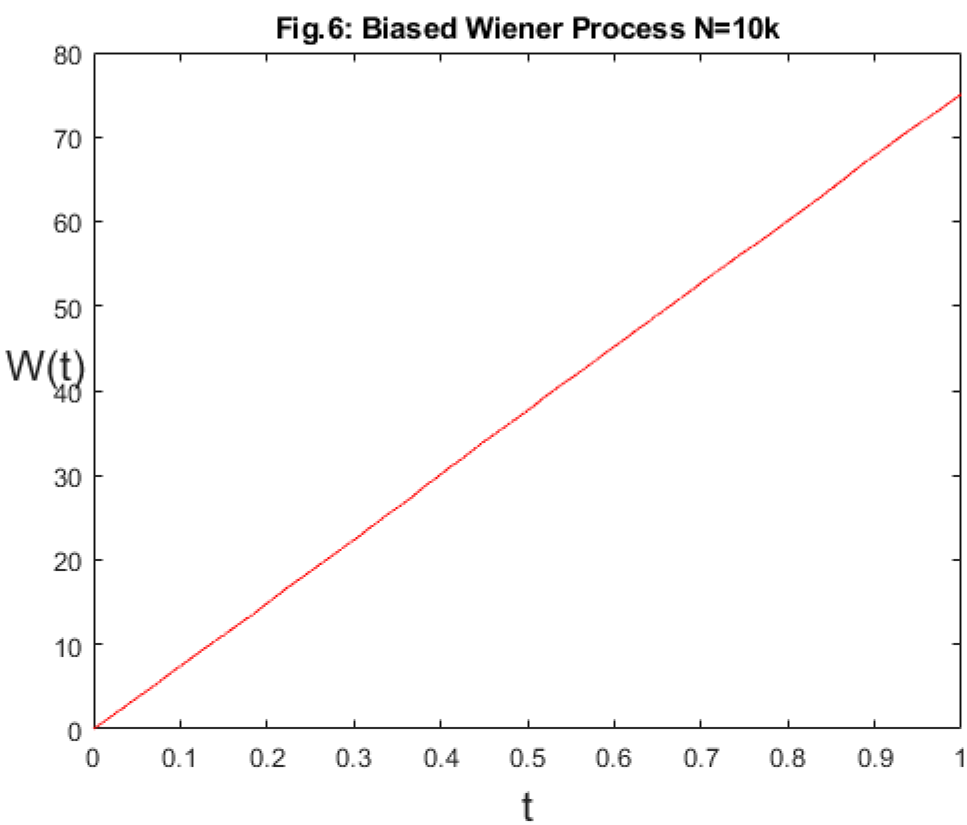


Fig.6: Biased Wiener Process N=10k
W(t)
t
0
10
20
30
40
50
60
70
80
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

Mean of biased W1

```
mean(W1)
% Uncomment the code below to see proof that W1 agree with theoretical.
% W1-mean(W1,1)
```

```
ans =

    3.6950
```

Variance of biased W1

```
var(W1)
var(var(W1))
```

```
ans =

    4.3671


ans =

     0
```

Mean of biased W2

```
mean(W2)
% Uncomment the code below to see proof that W2 agree with theoretical.
% W2-mean(W2,1)
```

```
ans =

   11.9796
```

Variance of biased W2

```
var(W2)
var(var(W2))
```

```
ans =

   47.7432


ans =

     0
```

Mean of biased W3

```
mean(W3)
% Uncomment the code below to see proof that W3 agree with theoretical.
% W3-mean(W3,1)
```

```
ans =

   37.5856
```

Variance of biased W1

```
var(W3)
var(var(W3))
```

```
ans =

  473.4633


ans =

     0
```

Observation: The mean of the biased Wiener process still is similar to the above discussed mean od standard random walk -- which is defined by '0'. Moreover, the varaince of the second moment of it yield a value of 0.

d. What happens as n approaches infinity? Why? As n approaches infinity, the random walk becomes finer but in contrast to the standard Wiener process, its density is not observable -- which is observable in figures 4-6.

Discussion: Based from the above results and quick observations, a random walk is said to be **symmetric if (1)** $X_0 = 0$**, (2) the random varaibles are independent, and (3) each Sn has values [-1 1] given p=0.5** . A random walk is said to be **biased with parameters (0,1) if (1)** $X_0 = 0$**, (2) the random varaibles are independent, and (3) Dn has a distribution [-1 1] for 1-p and p** .

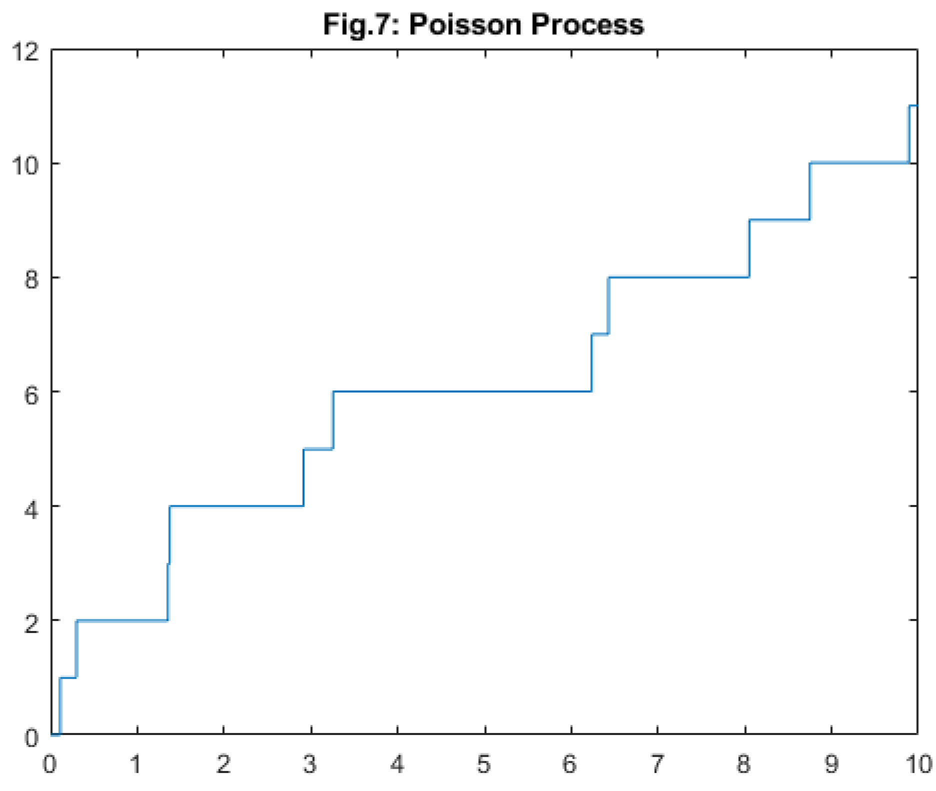
III. Poisson Process

a. Experimentally verify the mean and variance of the Poisson process.

```
Tmax = 10;
lambda = 1;
n=0; % Number of points
Tlast = 0; % Time of last arrival
while Tlast <= Tmax
n = n + 1;
X = -log(rand(1))/lambda; % Generate exp(lambda) RV
Tlast = Tlast + X;
```

```
T(n) = Tlast;
E(n) = n*mean(T(n));
V(n) = n*var(T(n));
end
n = n-1; % Remove last arrival,
T = T(1:n); % which is after Tmax.
fprintf('There were %g arrivals in [0,%g].\n',n,Tmax)
tt = kron(T,[1 1]); % Convert [x y z] to [x x y y z z]
tt = [ 0 tt Tmax ];
N = [ 0:n ]; % Values of the Poisson process
NN = kron(N,[1 1]);
figure; plot(tt,NN); title('Fig.7: Poisson Process');
axis([0 Tmax 0 n+1]);
```

There were 11 arrivals in [0,10].



The mean and variance of the Poisson process as simulated above are given by the variables E for mean and V for variance. It is noticeable that the variance of each element of the T yields '0' which is quite inaccordance to the theoretical.

Mean of 3a

```
E

mean(E)
```

E =

```
  Columns 1 through 7

    0.1114    0.5969    4.0641    5.4932   14.5737   19.5283   43.6473

  Columns 8 through 14

   51.4359   72.4512   87.4912  108.8051  121.9904  127.7614  145.2649


ans =

   57.3725
```

Variance of 3a

```
V

var(V)


V =

  Columns 1 through 13

     0     0     0     0     0     0     0     0     0     0     0     0     0

  Column 14

     0


ans =

     0
```

b. Modify the code to simulate the kth arrival time of a Poisson process.

```
lambda=1;      % arrival rate
Tmax=10;         % maximum time
clear T;
T(1)=random('Exponential',1/lambda);
Y(1)=mean(T(1));
V(1)=var(T(1));
i=1;
while T(i) < Tmax
  T(i+1)=T(i)+random('Exponential',1/lambda);
  Y(i+1)=mean(T(i+1));
  V(i+1)=var(T(i+1));
  i=i+1;
end
T(i)=Tmax;
figure; stairs(T(1:i), 0:(i-1)); title('Fig.8: Modified Poisson Process');
```

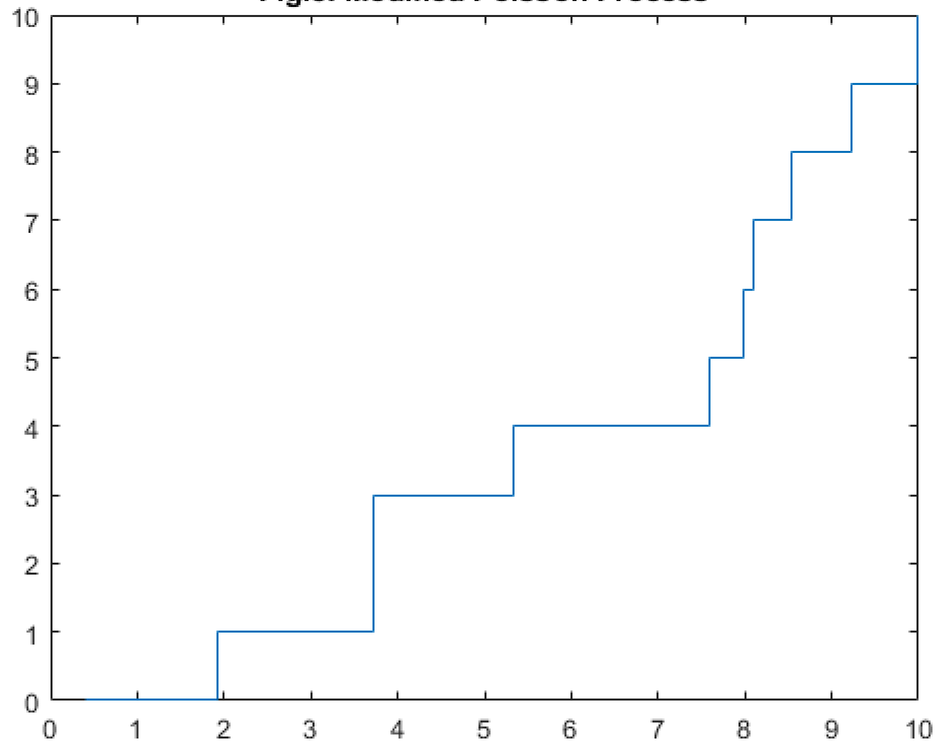


Fig.8: Modified Poisson Process

The modified code above demonstrate the exponential distrib. which generate the both the mean (Y variable) and variance (V variable) of each kth arrival.

c. Experimentally verify the mean and variance of the kth arrival time.

Similar to the codes in 3a, the mean of each kth arrival is in variable Y while variance is in variable V. Moreover, it is noticeable that the variance yield a vector of 0. The following below show these in detail.

Mean of 3c

```
Y

mean(Y)
```

```
Y =

  Columns 1 through 7

    0.4118    1.9313    3.7243    3.7300    5.3412    7.5821    7.9755

  Columns 8 through 11

    8.1057    8.5303    9.2218   10.1281


ans =
```

```
    6.0620
```

Variance of 3c

```
V

var(V)
```

```
V =

  Columns 1 through 13

     0     0     0     0     0     0     0     0     0     0     0     0     0

  Column 14

     0


ans =

     0
```

Published with MATLAB® R2021b