STAT 753: Stochastic Models and Simulations
Jaleesa Houle
University of Nevada, Reno - Spring 2024

Homework 10

Theory

1. Consider the standard Brownian motion $W = (W(t), t \geq 0)$. Find:

   A. $\mathbb{P}(1 < W(4) < 3)$
   B. $\mathbb{P}(W(3) > W(1) + 1)$
   C. $\mathbb{P}(W(1) < W(2) < W(4))$
   D. $\mathbb{P}(-5 < W(5) < 0 | W(1.4) = -2)$
   E. $\mathbb{E}[W^3(t)]$ for $t > 0$
   F. $\mathbb{E}[W^4(t)]$ for $t > 0$
   G. $\mathbb{E}[W^2(t) | W(s) = x]$ for 0<s<t and x $\in \mathbb{R}$

Solution

A. $\mathbb{P}(1 < W(4) < 3)$

For standard Brownian motion $W = (W(t), t \geq 0)$, $W(t) - W(s) \sim N(0, t - s)$. $W(4)$ is then $W(4) - W(0) \sim N(0, 4-0) \sim N(0, 4)$. We can convert this distribution to standard normal and solve:

$$\begin{aligned}
\mathbb{P}(1 < W(4) < 3) &= \mathbb{P}(\frac{1-0}{\sqrt{4}} < \frac{W(4)-0}{\sqrt{4}} < \frac{3-0}{\sqrt{4}}) \\
&= \mathbb{P}(\frac{1}{2} < Z < \frac{3}{2}) \\
&= \mathbb{P}(Z < \frac{3}{2}) - P(Z < \frac{1}{2}) \\
&= \Phi(\frac{3}{2}) - \Phi(\frac{1}{2}) \\
&= 0.9332 - 0.6915 \\
&= 0.2417.
\end{aligned} \tag{1}$$

B. $\mathbb{P}(W(3) > W(1) + 1)$

The probability $\mathbb{P}(W(3) > W(1) + 1)$ can be rewritten as $\mathbb{P}(W(3) - W(1) > 1)$. For standard Brownian motion, $W(3) - W(1) \sim N(0, 3-1) \sim N(0, 2)$. We can define $\zeta = W(3) - W(1)$, convert to standard normal and then solve:

$$\begin{aligned}
\mathbb{P}(W(3) > W(1)+1) &= \mathbb{P}(W(3) - W(1) > 1) \\
&= \mathbb{P}(\zeta > 1) \\
&= \mathbb{P}(\frac{\zeta - 0}{\sqrt{2}} > \frac{1}{\sqrt{2}}) \\
&= 1 - \Phi(\frac{1}{\sqrt{2}}) \\
&\approx 1 - 0.7611 \\
&= 0.2389.
\end{aligned} \tag{2}$$

C. $\mathbb{P}(W(1) < W(2) < W(4))$

We can rewrite $\mathbb{P}(W(1) < W(2) < W(4))$ as $\mathbb{P}(W(1) < W(2) \cap W(2) < W(4))$. Since Brownian motion has independent increments, this becomes $\mathbb{P}(W(1) < W(2)) \cdot \mathbb{P}(W(2) < W(4))$. We can rearrange this expression as $\mathbb{P}(0 < W(2) - W(1)) \cdot \mathbb{P}(0 < W(4) - W(2))$. Then, by defining $\zeta_1 = W(2) - W(1) \sim N(0,1)$ and $\zeta_2 = W(4) - W(2) \sim N(0,2)$, we can solve:

$$\begin{aligned}
\mathbb{P}(0 < W(2) - W(1)) \cdot \mathbb{P}(0 < W(4) - W(2)) &= \mathbb{P}(0 < \zeta_1) \cdot \mathbb{P}(0 < \frac{\zeta_2 - 0}{\sqrt{2}}) \\
&= (1 - \Phi(0)) \cdot (1 - \Phi(0)) \\
&= 0.5 \cdot 0.5 \\
&= 0.25.
\end{aligned} \tag{3}$$

D. $\mathbb{P}(-5 < W(5) < 0 | W(1.4) = -2)$

The probability $\mathbb{P}(-5 < W(5) < 0 | W(1.4) = -2)$ can be rewritten as $\mathbb{P}(-5-(-2) < W(5) - W(1.4) < 0 - (-2)) = \mathbb{P}(-3 < W(5) - W(1.4) < 2)$. We can let $\zeta = W(5) - W(1.4) \sim N(0, 3.6)$ and then solve:

$$\begin{aligned}
\mathbb{P}(-3 < W(5) - W(1.4) < 2) &= \mathbb{P}(-3 < \zeta < 2) \\
&= \mathbb{P}(\frac{-3}{\sqrt{3.6}} < \frac{\zeta}{\sqrt{3.6}} < \frac{2}{\sqrt{3.6}}) \\
&= \mathbb{P}(-1.5811 < Z < 1.0541) \\
&= \Phi(1.0541) - \Phi(-1.5811) \\
&\approx .8531 - .05705 \\
&= 0.7961.
\end{aligned} \tag{4}$$

E. $\mathbb{E}[W^3(t)]$ **for** $t > 0$

In Homework 2, $E[X^3]$ for a Gaussian random variable $\sim N(\mu, \sigma^2)$ was derived and found to be $E[X^3] = \mu^3 + 3\sigma^2\mu$. For standard Brownian motion, $W(t) \sim N(0,t)$ for s=0. If we plug in the variance and mean of $W(t)$, we find that $E[W^3(t)] = 0$ and $E[W^4(t)] = t^2$.

F. $\mathbb{E}[W^4(t)]$ **for** $t > 0$

In Homework 2, $E[X^4]$ for a Gaussian random variable $\sim N(\mu, \sigma^2)$ was derived and found to be $E[X^4] = \mu^4 + 6\sigma^2\mu^2 + 3\sigma^4$. If we again plug in the variance and mean of $W(t)$, we find that $E[W^4(t)] = t^2$.

G. $\mathbb{E}[W^2(t)|W(s) = x]$ **for 0<s<t and x** $\in \mathbb{R}$

To find $\mathbb{E}[W^2(t)|W(s) = x]$, we can first define $\zeta = W(t) - W(s)$, then substitute and solve:

$$\begin{aligned}\mathbb{E}[W^2(t)|W(s) = x] &= E[(\zeta + x)^2] \\ &= E[\zeta^2] + E[2x\zeta] + E[x^2] \\ &= (Var(\zeta) - E[\zeta]^2) + 0 + x^2 \\ &= (Var(W(t) - W(s)) - E[W(t) - W(s)]^2) + x^2 \\ &= t - s + x^2.\end{aligned} \tag{5}$$

Alternatively, we could use the results for $\mathbb{E}[W(t)|W(s) = x)]$ and $Var[W(t)|W(s) = 0]$ derived in class:

$$\begin{aligned}\mathbb{E}[W^2(t)|W(s) = x] &= Var(W(t)|W(s) = 0) + \mathbb{E}[W(t)|W(s) = x)]^2 \\ &= (t - s) + x^2.\end{aligned} \tag{6}$$

2. Take a Brownian motion $X = (X(t), t \geq 0)$ with drift $\mu = 1.5$ and diffusion $\sigma^2 = 0.25$. Assume it starts from $X(0) = -2.4$. Find:

A. $\mathbb{P}(X(3) > 0)$
B. $\mathbb{P}(X(5) > -2|X(3) = -1)$
C. The density of $X(5)$
D. $\mathbb{E}[X^2(5)]$
E. $\mathbb{E}[X^3(5)]$
F. $\mathbb{P}(1 + X(1) < X(3))$
G. $\mathbb{P}(X(1) < X(2) < X(4))$

Solution
A. $\mathbb{P}(X(3) > 0)$

To solve $\mathbb{P}(X(3) > 0)$, we should first find the probability distribution for $X(5)$. We can determine this using the independent increments property of general Brownian motion with drift and diffusion, where $X(t) - X(s) \sim N(\mu(t-s), \sigma^2(t-s))$. For this case, $X(3) - X(0) \sim N(1.5(3-0), 0.25(3-0))$. Since we are given that $X(0) = -2.4$, this becomes $X(3) \sim N(1.5(3-0), 0.25(3-0)) + X(0)$, which can be further simplified as $X(5) \sim N(4.5 - 2.4, 0.75)$. Thus, $\mathbb{P}(X(3) > 0)$ for $X(5) \sim (2.1, 0.75)$ can be calculated as:

$$
\begin{aligned}
\mathbb{P}(X(3) > 0) &= \mathbb{P}(\frac{X(3) - 2.1}{\sqrt{0.75}} > \frac{0 - 2.1}{\sqrt{0.75}}) \\
&= \mathbb{P}(Z > -2.4248) \\
&= 1 - \mathbb{P}(Z < -2.4248) \\
&= 1 - \Phi(-2.4248) \\
&\approx 1 - .00776 \\
&= 0.992.
\end{aligned} \tag{7}
$$

B. $\mathbb{P}(X(5) > -2|X(3) = -1)$

By independence, the probability $\mathbb{P}(X(5) > -2|X(3) = -1)$ can be rewritten as $\mathbb{P}(X(5) - X(3) > -2 - (-1))$. Then, we can see that $X(5) - X(3) \sim N(1.5(5 - 3), .25(5 - 3) \sim N(3, 0.5)$. For simplicity, we can define $X(5) - X(3) = \zeta$ and then solve by converting to standard normal using a z-table:

$$
\begin{aligned}
\mathbb{P}(X(5) > -2|X(3) = -1) &= \mathbb{P}(X(5) - X(3) > -1) \\
&= \mathbb{P}(\frac{\zeta - 3}{\sqrt{0.5}} > \frac{-1 - 3}{\sqrt{0.5}}) \\
&= \mathbb{P}(Z > -5.6568) \\
&= 1 - \Phi(-5.6568) \\
&\approx 1.
\end{aligned} \tag{8}
$$

C. The density of $X(5)$

The parameters μ and σ^2 of $X(5)$ can be found in the same way as above:

$$\begin{aligned} X(5) - X(0) &\sim N(1.5(5-0), 0.25(5-0)) \\ &\sim N(1.5(5-0), 0.25(5-0)) + X(0) \\ &\sim N(1.5(5-0) + X(0), 0.25(5-0)) \\ &\sim N(5.1, 1.25). \end{aligned} \tag{9}$$

The density of $X(5)$ is then

$$X(5) = \frac{1}{\sqrt{2\pi}\sqrt{1.25}} exp\left(\frac{-(x-5.1)^2}{2.5}\right). \tag{10}$$

D. $\mathbb{E}[X^2(5)]$

We can compute $\mathbb{E}[X^2(5)]$ using the relationship $E[X^2] = Var(X) + E[X]^2$, where $X(5) \sim N(5.1, 1.25)$. We can see that the variance is 1.25 and $\mathbb{E}[X(5)]^2 = 5.1^2 = 26.01$. Thus, $\mathbb{E}[X^2(5)] = 1.25 + 26.01 = 27.26$.

E. $\mathbb{E}[X^3(5)]$

In homework 2, we derived that $\mathbb{E}[X^3]$ for $X \sim N(\mu, \sigma^2) = \mu^3 + 3\sigma^2\mu$. For $X(5) \sim N(5.1, 1.25)$, $\mathbb{E}[X^3]$ is then $5.1^3 + 3(1.25)^2 \cdot 5.1 = 156.56$.

F. $\mathbb{P}(1 + X(1) < X(3))$

The probability $\mathbb{P}(1 + X(1) < X(3))$ can be rewritten as $\mathbb{P}(1 < X(3) - X(1))$, where $X(3) - X(1) \sim N(1.5(3-1), .25(3-1)) \sim N(3, 0.5)$. The solution can be found to be

$$\begin{aligned} \mathbb{P}(1 < X(3) - X(1)) &= \mathbb{P}(\frac{1-3}{\sqrt{0.5}} < Z) \\ &= 1 - \Phi(-2.8284) \\ &\approx .99767. \end{aligned} \tag{11}$$

G. $\mathbb{P}(X(1) < X(2) < X(4))$

Similar to Theory 1c, We can rewrite $\mathbb{P}(X(1) < X(2) < X(4))$ as $\mathbb{P}(X(1) < X(2) \cap X(2) < X(4))$, which, by independence, becomes $\mathbb{P}(X(1) < X(2)) \cdot \mathbb{P}(X(2) < X(4))$. We can rearrange this expression as $\mathbb{P}(0 < X(2) - X(1)) \cdot \mathbb{P}(0 < X(4) - X(2))$. Then, by defining $\zeta_1 = X(2) - X(1) \sim N(1.5, 0.25)$ and $\zeta_2 = X(4) - X(2) \sim N(3, 0.5)$, we can solve:

$$
\begin{aligned}
\mathbb{P}(0 < X(2) - X(1)) \cdot \mathbb{P}(0 < X(4) - X(2)) &= \mathbb{P}\left(\frac{0 - 1.5}{\sqrt{0.25}} < \frac{\zeta_1 - 1.5}{\sqrt{0.25}}\right) \cdot \mathbb{P}\left(\frac{0 - 3}{\sqrt{0.5}} < \frac{\zeta_2 - 3}{\sqrt{0.5}}\right) \\
&= (1 - \Phi(-3)) \cdot (1 - \Phi(-4.2426)) \\
&\approx 0.99865 \cdot 1 \\
&= 0.99865.
\end{aligned} \tag{12}
$$

3. Take a Levy process $L = (L(t), t \geq 0)$ which is a sum of independent Brownian motion from Theory 2 and a compound Poisson process with intensity $\lambda = 0.4$ and jumps with Laplace distribution with mean 0.3 and standard deviation 1.2. Find the mean, variance, and the MGF of $L(t)$.

Solution

We can write the equation for the Levy process $L(t)$ as $L(t) = X(t) + Y(t)$ where $X(t) = -2.4 + 1.5t + 0.5W(t)$ and $Y(t) = \sum_{k=1}^{N(t)} Y_k$ for Y_k IID Laplace variables with $\mu = 0.3$, $\sigma^2 = 1.44$, and $N(t) \sim Exp(\lambda = 0.4)$. We know for a compound Poisson process with mean μ, variance σ^2, and intensity λ, $E[Y(t)] = \mu\lambda t$ and $Var(Y(t)) = \lambda t(\mu^2 + \sigma^2)$. The expected value can then be found to be

$$\begin{aligned}
\mathbb{E}[L(t)] &= \mathbb{E}[X(t) + Y(t)] \\
&= \mathbb{E}[-2.4 + 1.5t + 0.5W(t) + \sum_{k=1}^{N(t)} Y_k] \\
&= \mathbb{E}[-2.4] + \mathbb{E}[1.5t] + \mathbb{E}[0.5W(t)] + \mathbb{E}[\sum_{k=1}^{N(t)} Y_k] \\
&= -2.4 + 1.5t + 0 + \mu\lambda t \\
&= -2.4 + 1.5t + (0.3)(0.4)t \\
&= 1.62t - 2.4.
\end{aligned} \tag{13}$$

Similarly, we can compute the variance as

$$\begin{aligned}
Var(L(t)) &= Var(X(t) + Y(t)) \\
&= Var(-2.4 + 1.5t + 0.5W(t) + \sum_{k=1}^{N(t)} Y_k) \\
&= 0 + 0 + 0.25Var(W(t)) + Var(\sum_{k=1}^{N(t)} Y_k)) \\
&= 0.25t + \lambda t(\mu^2 + \sigma^2) \\
&= 0.25t + 0.4t(0.09 + 1.44) \\
&= 0.862t.
\end{aligned} \tag{14}$$

Thus, the mean for $L(t)$ is $1.62t - 2.4$ and variance is $0.862t$. Lastly, we can compute the MGF of $L(t)$ by first noting that for a compound Poisson process

$$\begin{aligned}
\mathbb{E}[e^{uY(t)}] &= e^{(\lambda t(M_z(u)-1))} \\
\text{where } M_z(u) &= \mathbb{E}[e^{(uZ_k)}]
\end{aligned} \tag{15}$$

For a Laplace random variable,

$$M_Z(u) = \mathbb{E}[e^{uZ_k}] = e^{\left(\frac{\mu u}{(1-0.5\sigma^2 u^2)}\right)}. \tag{16}$$

We can also observe that since $X(t) \sim N(1.5t - 2.4, 0.25t)$ for $t > 0$, the MGF for $X(t)$ is the normal distribution MGF

$$\mathbb{E}[e^{uX(t)}] = e^{(u\mu + 0.5\sigma^2 u^2)}. \tag{17}$$

For this problem, the parameters for $M_z(u)$ are $\mu = 0.3$ and $\sigma^2 = 1.44$, the parameter for $\mathbb{E}[e^{uY(t)}]$ is $\lambda = 0.4$, and the parameters for $\mathbb{E}[e^{uX(t)}]$ are $\mu = 1.5t - 2.4$ and $\sigma^2 = 0.25t$. Now, we can use this information to find the MGF of $L(t)$:

$$\begin{aligned} M_z(u) &= e^{\left(\frac{\mu u}{(1-0.5\sigma^2 u^2)}\right)} \\ &= e^{\left(\frac{0.3u}{(1-0.72u^2)}\right)} \end{aligned} \tag{18}$$

$$\begin{aligned} \mathbb{E}[e^{uY(t)}] &= e^{(\lambda t(M_z(u)-1))} \\ &= exp\left(0.4t(e^{\left(\frac{0.3u}{(1-0.72u^2)}\right)} - 1)\right) \end{aligned} \tag{19}$$

$$\begin{aligned} \mathbb{E}[e^{uX(t)}] &= e^{(u\mu + 0.5\sigma^2 u^2)} \\ &= e^{(u(1.5t-2.4)+0.5(0.25)u^2)} \end{aligned} \tag{20}$$

$$\begin{aligned} \mathbb{E}[e^{L(t)}] &= \mathbb{E}[e^{X(t)}]\mathbb{E}[e^{Y(t)}] \\ &= e^{(u(1.5t-2.4)+0.5(0.25)u^2)} \cdot e^{(0.4t(e^{\left(\frac{0.3u}{(1-0.72u^2)}\right)}-1))} \\ &= e^{(tf(u)-2.4u)} \\ &\text{where } f(u) = 0.125u^2 + 1.5u + 0.4\left(e^{\left(\frac{0.3u}{(1-0.72u^2)}\right)} - 1\right). \end{aligned} \tag{21}$$

The MGF for $L(t)$ is then $e^{(tf(u)-2.4u)}$, where $f(u) = 0.125u^2 + 1.5u + 0.4e^{\left(\frac{0.3u}{(1-0.72u^2)}\right)} - 0.4$.

Code

1. Simulate the standard Brownian motion from Theory 1. Plot 4 simulation graphs for time $t \leq 4$. Using Monte Carlo approach, compute A, B, C, and write functions with input t to compute E, F.

Solution

```python

def brownian_motion_sim( dt, T, mu=0, sigma=1, x0=None):
    N = int(T/dt)
    BM = np.append(np.zeros(1),np.cumsum(np.random.normal(mu*dt, sigma*np.sqrt(dt), N)))
    if x0 is not None:
        BM= np.cumsum(np.append(x0, np.random.normal(mu*dt, sigma*np.sqrt(dt), N)))
    time = np.linspace(0,T, N+1)

    return BM, np.round(time,2)

def find_value_at_T(values,jumptimes, T):
    idx = np.where(jumptimes == T)[-1][-1]
    return values[idx]

dt = 0.01
N = 10000
T=5
sims=[]
np.random.seed(1234)
for i in range(N):
    a,b = brownian_motion_sim(dt, T)
    sims.append(a)
    time = b

np.random.seed(10000)
for i in np.random.choice(N, 4):
    plt.plot(time[0:401], sims[i][0:401], lw=1)

plt.xlabel('time')
plt.ylabel('values')
plt.title('Brownian Motion - Theory 1')
```

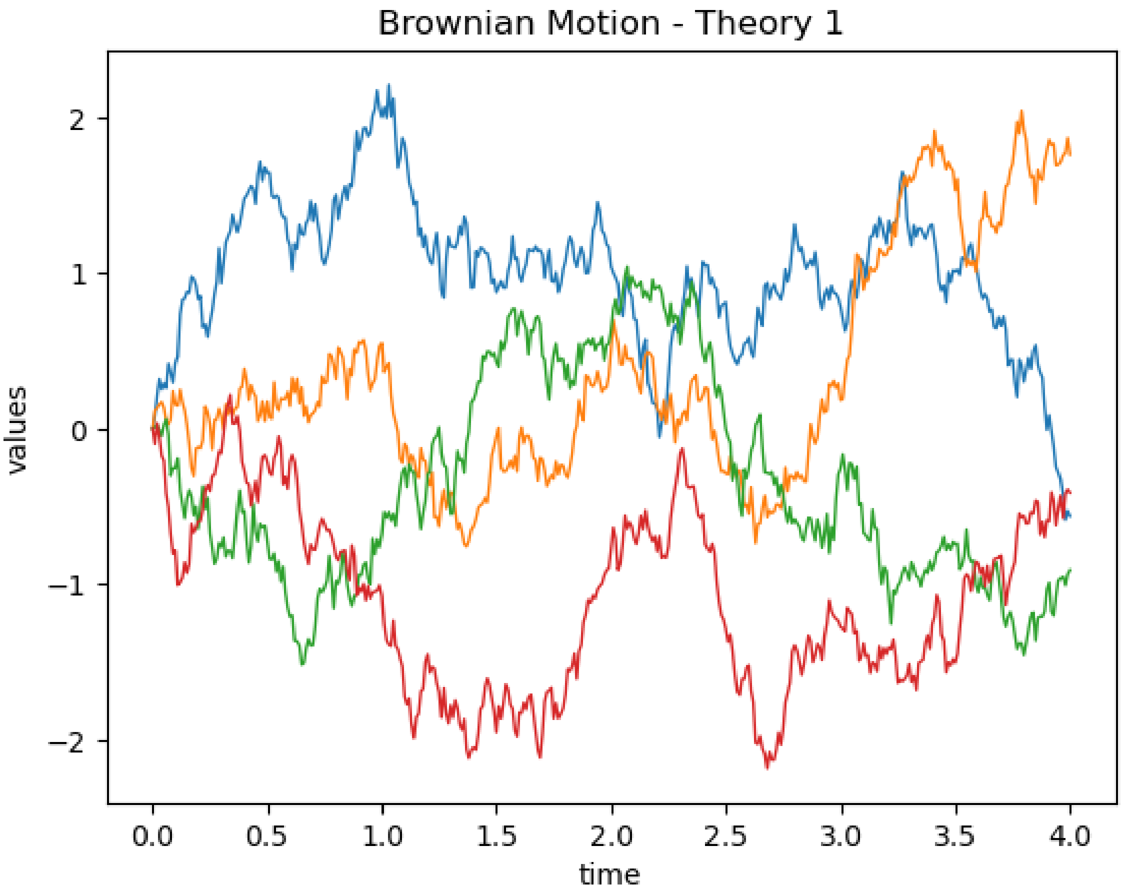


Figure 1: Four randomly selected simulations of general Brownian motion from Theory 1.

A. $\mathbb{P}(1 < W(4) < 3)$

```
count=0
for i in range(N):
    w_4 = find_value_at_T(sims[i], time, 4)
    if (w_4 >1) and (w_4<3):
        count+=1

print('P(1<W(4)<3)=', count/N)
>>
>> P(1<W(4)<3)= 0.243

```

B. $\mathbb{P}(W(3) > W(1) + 1)$

```
count=0
for i in range(N):
    w_1 = find_value_at_T(sims[i], time, 1)
    w_3 = find_value_at_T(sims[i], time, 3)
    if w_3 > (w_1+ 1):
```

```
        count+=1

print('P(W(3) > W(1) + 1)=', count/N)
>>
>> P(W(3) > W(1) + 1)= 0.2345

```

C. $\mathbb{P}(W(1) < W(2) < W(4))$

```
count=0
for i in range(N):
    w_1 = find_value_at_T(sims[i], time, 1)
    w_2 = find_value_at_T(sims[i], time, 2)
    w_4 = find_value_at_T(sims[i], time, 4)
    if (w_1 < w_2) & (w_2 < w_4):
        count+=1

print('P(W(1) < W(2) < W(4))=', count/N)
>>
>> P(W(1) < W(2) < W(4))= 0.2518

```

E. $\mathbb{E}[W^3(t)]$ **for** $t > 0$

```
def get_BM_skew(t):
    return 0

```

F. $\mathbb{E}[W^4(t)]$ **for** $t > 0$

```
def get_BM_kurtosis(t):
    return 3*t**2

```

2. Simulate the Brownian motion from Theory 2. Plot 4 simulation graphs for time $t \leq 5$. Empirically compute A, D, E, F, G.

Solution

```
dt = 0.01
T = 5
N = 10000
sims=[]
np.random.seed(12345)
for i in range(N):
    a,b = brownian_motion_sim(dt, T, mu=1.5, sigma=0.5, x0=-2.4)
    sims.append(a)
    time = np.round(b,2)

np.random.seed(10000)
for i in np.random.choice(N,4):
    plt.plot(time, sims[i], lw=1)

plt.xlabel('time')
plt.ylabel('values')
plt.title('Brownian Motion - Theory 2')
```

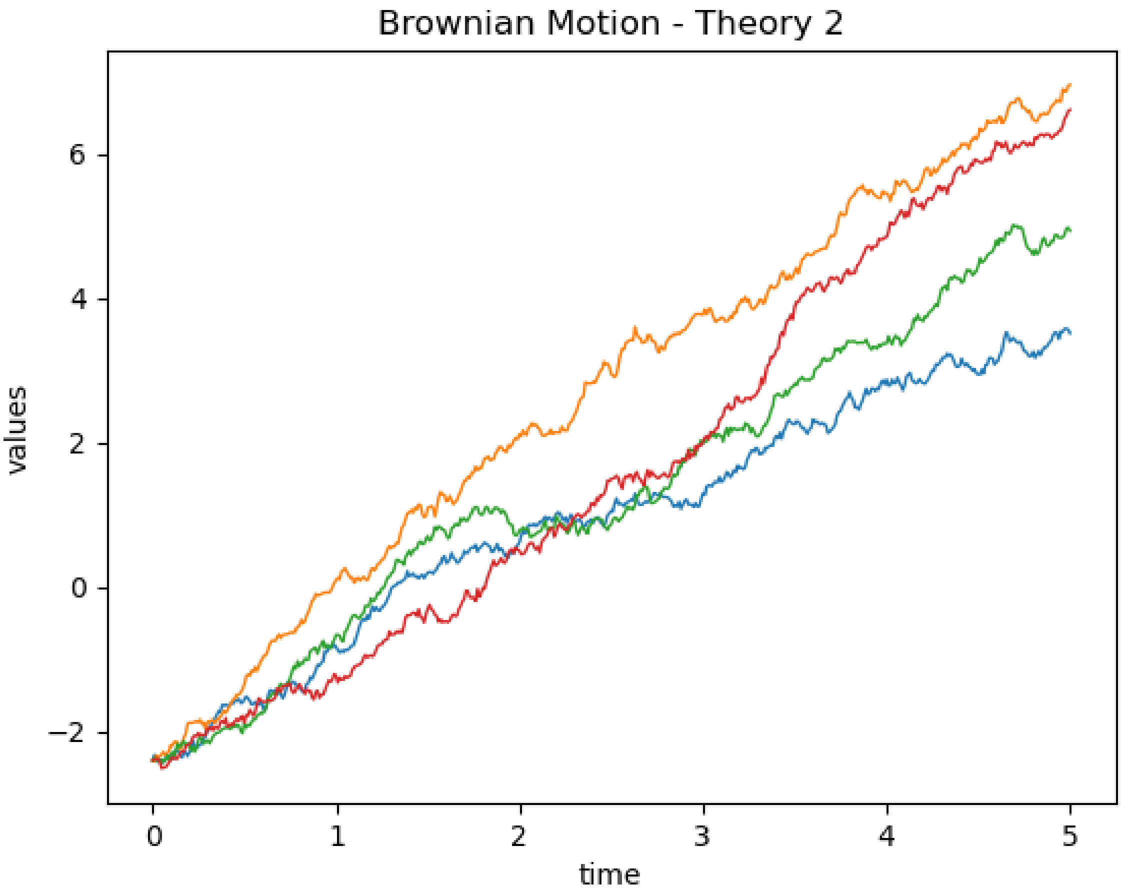


Figure 2: Four randomly selected simulations of Brownian motion with drift and diffusion from Theory 2.

A. $\mathbb{P}(X(3) > 0)$

```
count=0
for i in range(N):
    x_3 = find_value_at_T(sims[i], time, 3)
    if x_3>0:
        count+=1

print('P(X(3)>0) = ', count/N)
>>
>>P(X(3)>0) =  0.9923

```

D. $\mathbb{E}[X^2(5)]$

```

x_5 = []
for i in range(N):
```

```
    x_5.append(find_value_at_T(sims[i], time, 5))

print('E[X^2(5)]=', np.mean((np.array(x_5))**2))
>>
>> E[X^2(5)]= 27.257475061609387

```

E. $\mathbb{E}[X^3(5)]$

```
print('E[X^3(5)]=',np.mean((np.array(x_5))**3))
>>
>> E[X^3(5)]= 151.80741540957388

```

F. $\mathbb{P}(1 + X(1) < X(3))$

```
count=0
for i in range(N):
    x_1= find_value_at_T(sims[i], time, 1)
    x_3 = find_value_at_T(sims[i], time, 3)
    if (x_1+1)< x_3:
        count+=1

print('P(1+ X(1) < X(3))=', count/N)
>>
>> P(1+ X(1) < X(3))= 0.9977

```

G. $\mathbb{P}(X(1) < X(2) < X(4))$

```
count=0
for i in range(N):
    x_1= find_value_at_T(sims[i], time, 1)
    x_2 = find_value_at_T(sims[i], time, 2)
    x_4 = find_value_at_T(sims[i], time, 4)
    if (x_1<x_2) and (x_2<x_4):
        count+=1

print('P(X(1) < X(2) < X(4))=', count/N)
>>
>> P(X(1) < X(2) < X(4))= 0.9989
```

3. Simulate the Levy process from Theory 3. Plot 4 simulation graphs for time $t \leq 5$

```
def simulate_compound_poisson_process(CPP_mu, CPP_sigma, intensity, N):
    jumpTimes= np.append(np.zeros(1),np.cumsum(np.random.exponential(scale=1/intensity, size=N)))
    compound_values=  np.append(np.zeros(1), np.random.laplace(loc=CPP_mu, scale=CPP_sigma/np.sqrt(2),  size=N))
    return compound_values, np.round(jumpTimes,2)

def Levy_sim(dt, T, BM_mu, BM_sigma, CPP_mu,CPP_sigma,intensity, x0):
    N = int(T/dt)
    BM_increments = np.append(x0,np.random.normal(BM_mu*dt, BM_sigma*np.sqrt(dt), N))
    CPP_increments, CPP_jumptimes = simulate_compound_poisson_process(CPP_mu,CPP_sigma,intensity,N)

    time = np.round(np.linspace(0,T, N+1),2)
    for i,count in enumerate(CPP_jumptimes):
        if count> np.max(time):
            break
        try:
            idx = np.where(time==count)[-1][-1]
            BM_increments[idx] += CPP_increments[i]
        except (IndexError):
            print('No matching time jump for', count)
    L= np.cumsum(BM_increments)
    return L, time


dt = 0.01
N = 10000
T=5
sims=[]
np.random.seed(12345)
for i in range(N):
    a,b = Levy_sim(dt, T, BM_mu=1.5, BM_sigma=0.5, CPP_mu=0.3, CPP_sigma=1.2, intensity=0.4, x0=-2.4)
    sims.append(a)
    time = b

np.random.seed(10000)
for i in np.random.choice(N, 4):
    plt.plot(time, sims[i], lw=1)

plt.xlabel('time')
plt.ylabel('values')
plt.title('Levy Process - Theory 3')
```

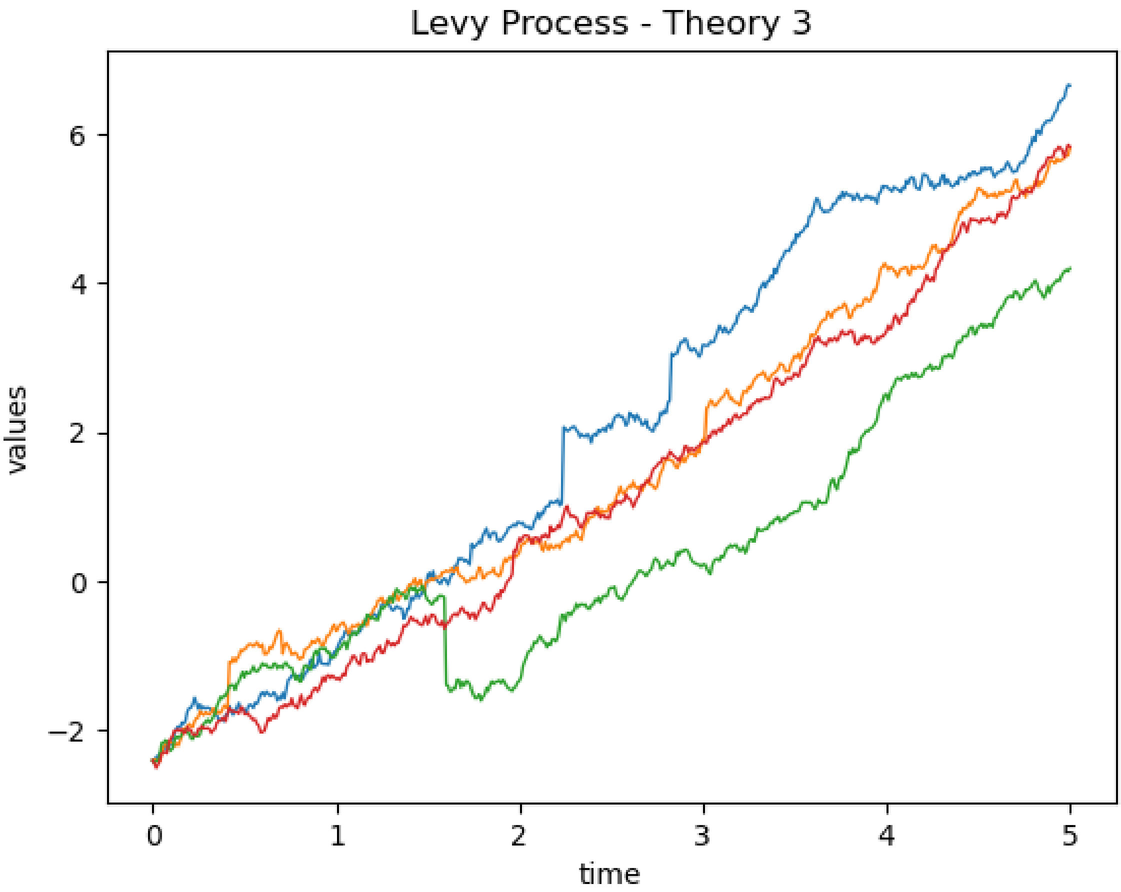


Figure 3: Four randomly selected Levy process simulations from Theory 3.

For completeness, the full pdf of my Jupyter Notebook workspace is included below.

In [1]:
```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

Simulate the standard Brownian motion from Theory 1. Plot 4 simulation graphs for time $t \leq 4$. Using Monte Carlo approach, compute A, B, C, and write functions with input t to compute E, F.

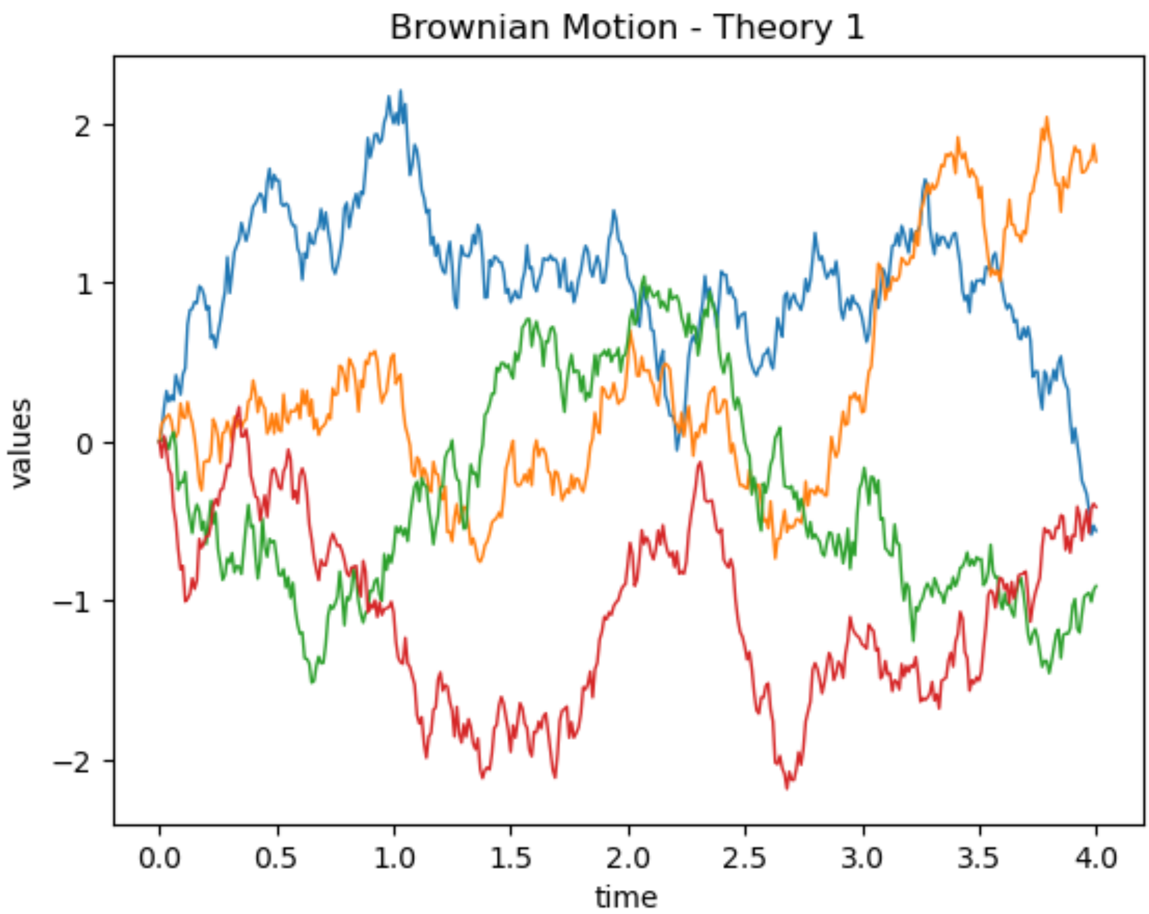
Theory 1: Consider the standard Brownian motion $W = (W(t), t \geq 0)$. Find:

In [2]:
```
def brownian_motion_sim( dt, T, mu=0, sigma=1, x0=None):
    N = int(T/dt)
    BM = np.append(np.zeros(1),np.cumsum(np.random.normal(mu*dt, sigma*np.sq
    if x0 is not None:
        BM= np.cumsum(np.append(x0, np.random.normal(mu*dt, sigma*np.sqrt(dt
    time = np.linspace(0,T, N+1)

    return BM, np.round(time,2)

def find_value_at_T(values,jumptimes, T):
    idx = np.where(jumptimes == T)[-1][-1]
    return values[idx]
```

In [3]:
```
dt = 0.01
N = 10000
T=5
sims=[]
np.random.seed(1234)
for i in range(N):
    a,b = brownian_motion_sim(dt, T)
    sims.append(a)
    time = b
```

In [4]:
```
np.random.seed(10000)
for i in np.random.choice(N, 4):
    plt.plot(time[0:401], sims[i][0:401], lw=1)

plt.xlabel('time')
plt.ylabel('values')
plt.title('Brownian Motion - Theory 1')
```

Out[4]:
```
Text(0.5, 1.0, 'Brownian Motion - Theory 1')
```

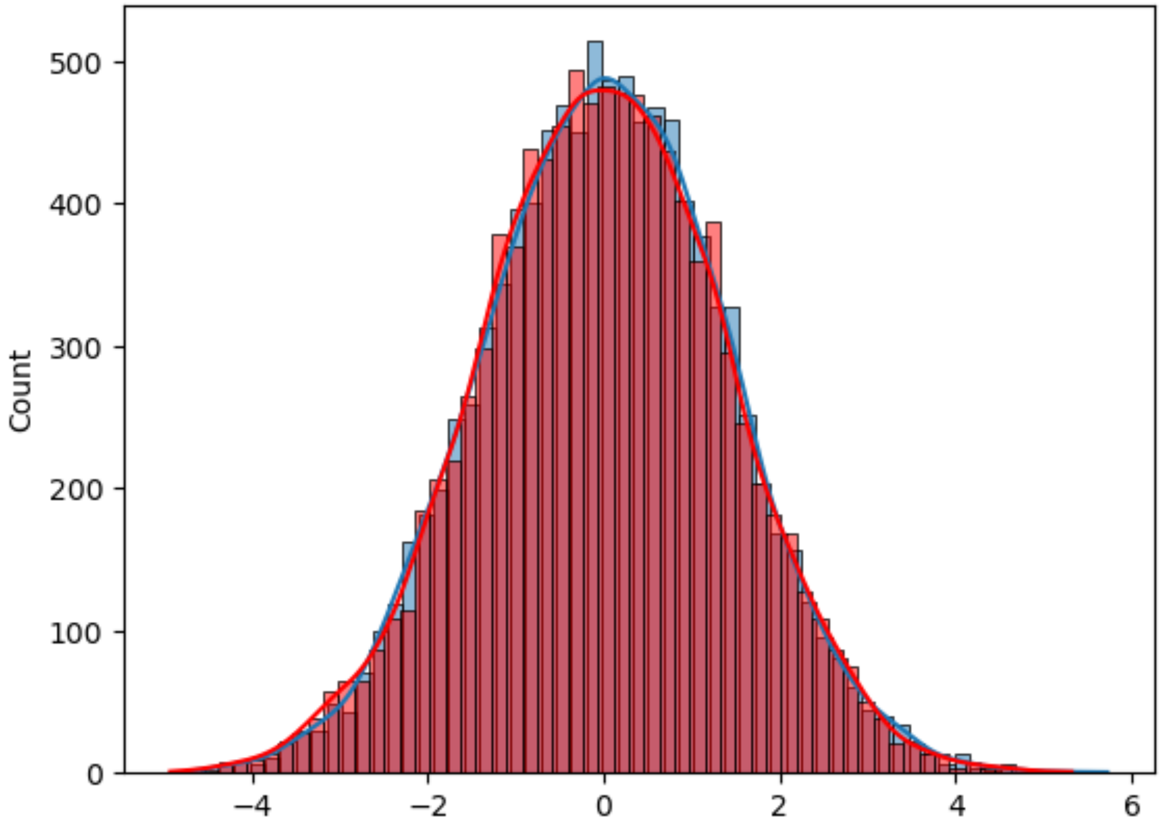


In [5]:

```
#verify pdf matches expected pdf
x_2 = []
for i in range(N):
    x_2.append(find_value_at_T(sims[i], time, 2))

x_2_rand = np.random.normal(loc=0, scale=np.sqrt(2), size=N)

sns.histplot(x_2, kde=True)
sns.histplot(x_2_rand, kde=True, color='red')
```

Out[5]:

```
<AxesSubplot:ylabel='Count'>
```



A. $\mathbb{P}(1 < W(4) < 3)$

In [6]:
```
count=0
for i in range(N):
    w_4 = find_value_at_T(sims[i], time, 4)
    if (w_4 >1) and (w_4<3):
        count+=1
```

In [7]:
```
print('P(1<W(4)<3)=', count/N)
```

```
P(1<W(4)<3)= 0.243
```

B. $\mathbb{P}(W(3) > W(1) + 1)$

In [8]:
```
count=0
for i in range(N):
    w_1 = find_value_at_T(sims[i], time, 1)
    w_3 = find_value_at_T(sims[i], time, 3)
    if w_3 > (w_1+ 1):
        count+=1
```

In [9]:
```
print('P(W(3) > W(1) + 1)=', count/N)
```

```
P(W(3) > W(1) + 1)= 0.2345
```

C. $\mathbb{P}(W(1) < W(2) < W(4))$

In [10]:
```
count=0
for i in range(N):
    w_1 = find_value_at_T(sims[i], time, 1)
    w_2 = find_value_at_T(sims[i], time, 2)
    w_4 = find_value_at_T(sims[i], time, 4)
    if (w_1 < w_2) & (w_2 < w_4):
        count+=1
```

In [11]:
```
print('P(W(1) < W(2) < W(4))=', count/N)
```

```
P(W(1) < W(2) < W(4))= 0.2518
```

D. $\mathbb{P}(-5 < W(5) < 0 | W(1.4) = -2)$

In [12]:
```
count1=0
count2= 0
for i in range(N):
    w = find_value_at_T(sims[i], time, 1.4)
    w_5 = find_value_at_T(sims[i], time, 5)
    if np.round(w,2) == -2.0:
        count1+=1
        if w_5 > -5 and w_5 < 0:
            count2+=1
print('P(-5 < W(5) < 0 | W(1.4) = -2)=', count2/count1)
```

```
P(-5 < W(5) < 0 | W(1.4) = -2)= 0.8888888888888888
```

E. $\mathbb{E}[W^3(t)]$ for $t > 0$

In [13]:
```
def get_BM_skew(t):
    return 0
```

F. $\mathbb{E}[W^4(t)]$ for $t > 0$

In [14]:
```
def get_BM_kurtosis(t):
    return 3*t**2
```

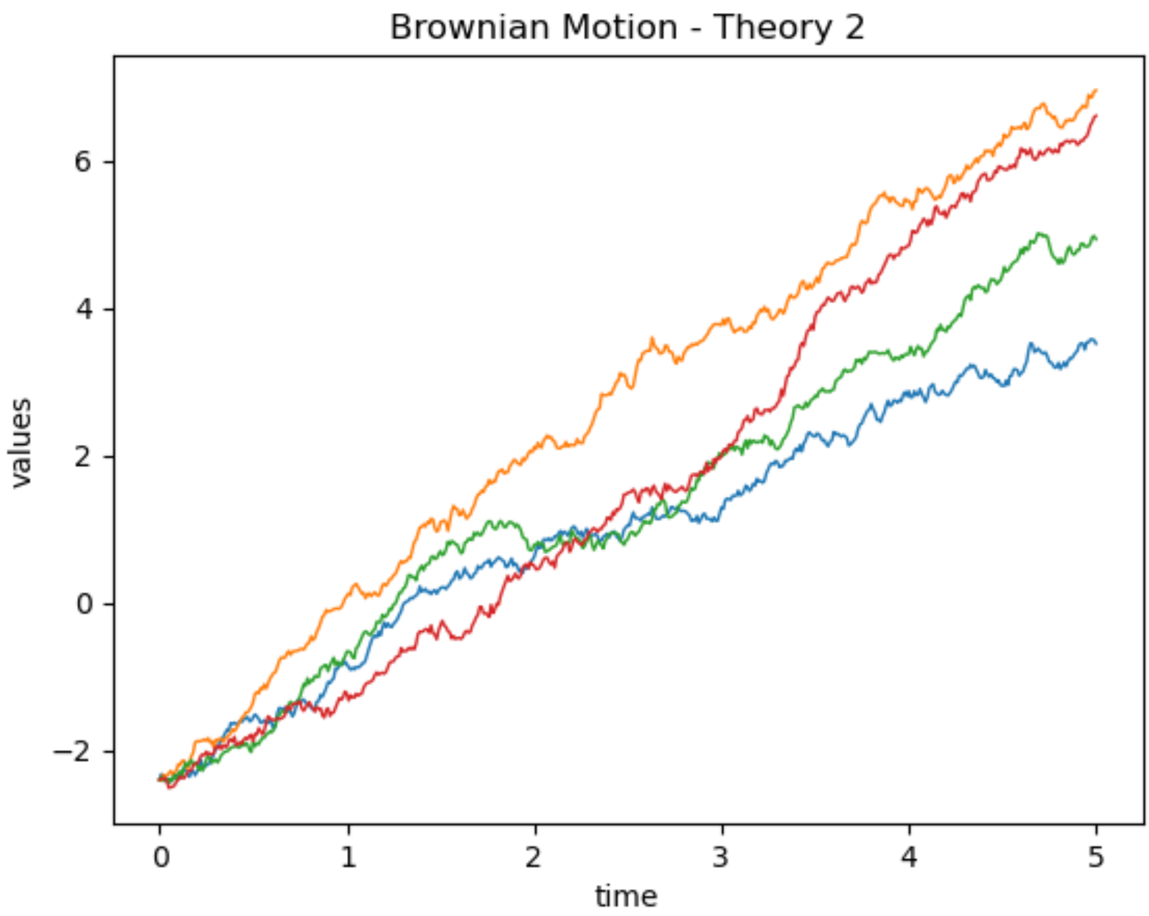
Simulate the Brownian motion from Theory 2. Plot 4 simulation graphs for time $t \leq 5$. Empirically compute A, D, E, F, G.

Theory 2: Take a Brownian motion $X = (X(t), t \geq 0)$ with drift $\mu = 1.5$ and diffusion $\sigma^2 = 0.25$. Assume it starts from $X(0) = -2.4$. Find:

In [15]:
```
dt = 0.01
T = 5
N = 10000
sims=[]
np.random.seed(12345)
for i in range(N):
    a,b = brownian_motion_sim(dt, T, mu=1.5, sigma=0.5, x0=-2.4)
    sims.append(a)
    time = np.round(b,2)
```

In [16]:
```
np.random.seed(10000)
for i in np.random.choice(N,4):
    plt.plot(time, sims[i], lw=1)

plt.xlabel('time')
plt.ylabel('values')
plt.title('Brownian Motion - Theory 2')
```

Out[16]:
```
Text(0.5, 1.0, 'Brownian Motion - Theory 2')
```



A. $\mathbb{P}(X(3) > 0)$

In [17]:
```
count=0
for i in range(N):
    x_3 = find_value_at_T(sims[i], time, 3)
    if x_3>0:
        count+=1
```

In [18]:
```
print('P(X(3)>0) = ', count/N)
```

```
P(X(3)>0) =  0.9923
```

B. $\mathbb{P}(X(5) > -2 | X(3) = -1)$

In [19]:
```
count1=0
count2= 0
for i in range(N):
    x_3 = find_value_at_T(sims[i], time, 3)
    x_5 = find_value_at_T(sims[i], time, 5)
    if np.round(x_3,2) == -1:
        count1+=1
        if x_5 > -2:
            count2+=1
```

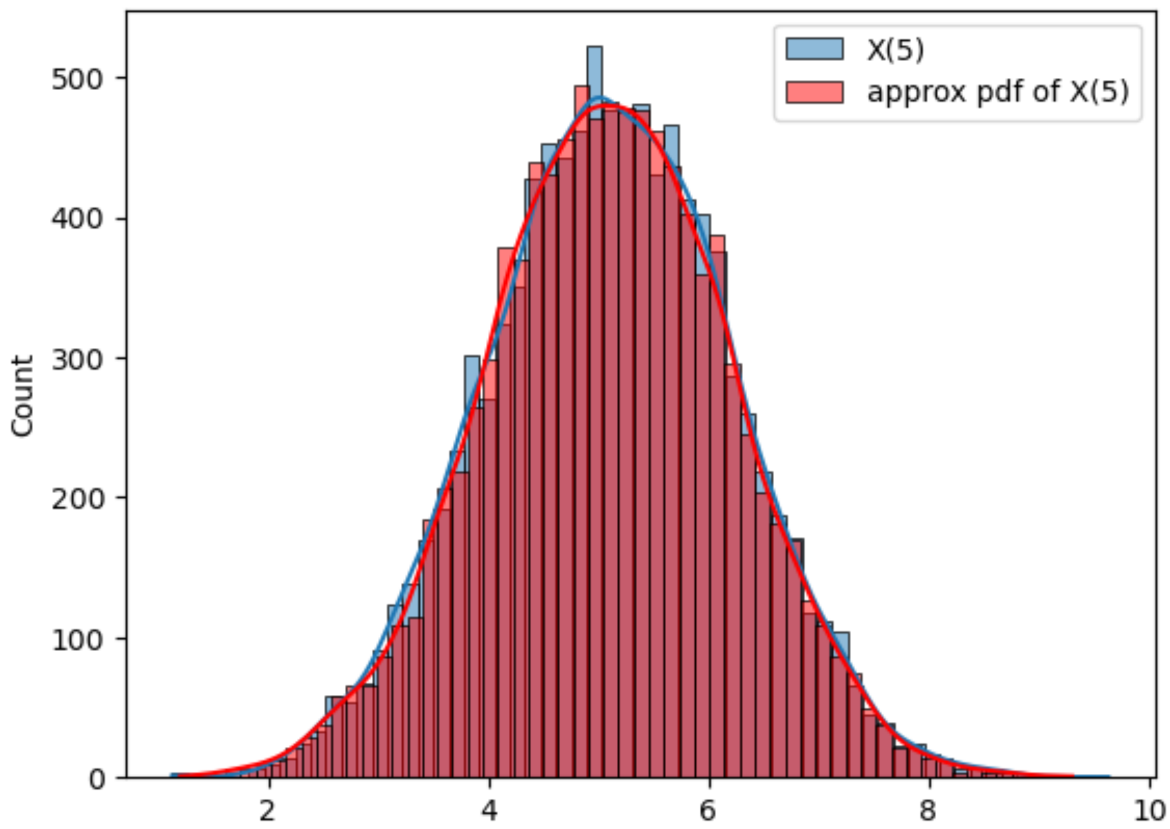
In [20]:
```
print('X(5)>-2 | X(3)=-1)=', count2/count1)
```

```
X(5)>-2 | X(3)=-1)= 1.0
```

C. The density of $X(5)$

In [21]:
```
x_5 = []
for i in range(N):
    x_5.append(find_value_at_T(sims[i], time, 5))

x_5_rand = np.random.normal(loc=7.5, scale=np.sqrt(1.25), size=N) + -2.4

sns.histplot(x_5, kde=True, label='X(5)')
sns.histplot(x_5_rand, kde=True, color='red', label='approx pdf of X(5)')

plt.legend()
```

Out[21]:
```
<matplotlib.legend.Legend at 0x7f8a2f8e9b80>
```



D. $\mathbb{E}[X^2(5)]$

In [22]:
```
print('E[X^2(5)]=', np.mean((np.array(x_5))**2))
```

```
E[X^2(5)]= 27.257475061609387
```

E. $\mathbb{E}[X^3(5)]$

In [23]:
```
print('E[X^3(5)]=',np.mean((np.array(x_5))**3))
```

```
E[X^3(5)]= 151.80741540957388
```

F. $\mathbb{P}(1 + X(1) < X(3))$

In [24]:
```
count=0
for i in range(N):
    x_1= find_value_at_T(sims[i], time, 1)
    x_3 = find_value_at_T(sims[i], time, 3)
    if (x_1+1)< x_3:
        count+=1
```

In [25]:
```
print('P(1+ X(1) < X(3))=', count/N)
```

```
P(1+ X(1) < X(3))= 0.9977
```

G. $\mathbb{P}(X(1) < X(2) < X(4))$

In [26]:
```
count=0
for i in range(N):
    x_1= find_value_at_T(sims[i], time, 1)
    x_2 = find_value_at_T(sims[i], time, 2)
    x_4 = find_value_at_T(sims[i], time, 4)
    if (x_1<x_2) and (x_2<x_4):
        count+=1
```

In [27]:
```
print('P(X(1) < X(2) < X(4))=', count/N)
```

```
P(X(1) < X(2) < X(4))= 0.9989
```

Simulate the Levy process from Theory 3. Plot 4 simulation graphs for time $t \le 5$

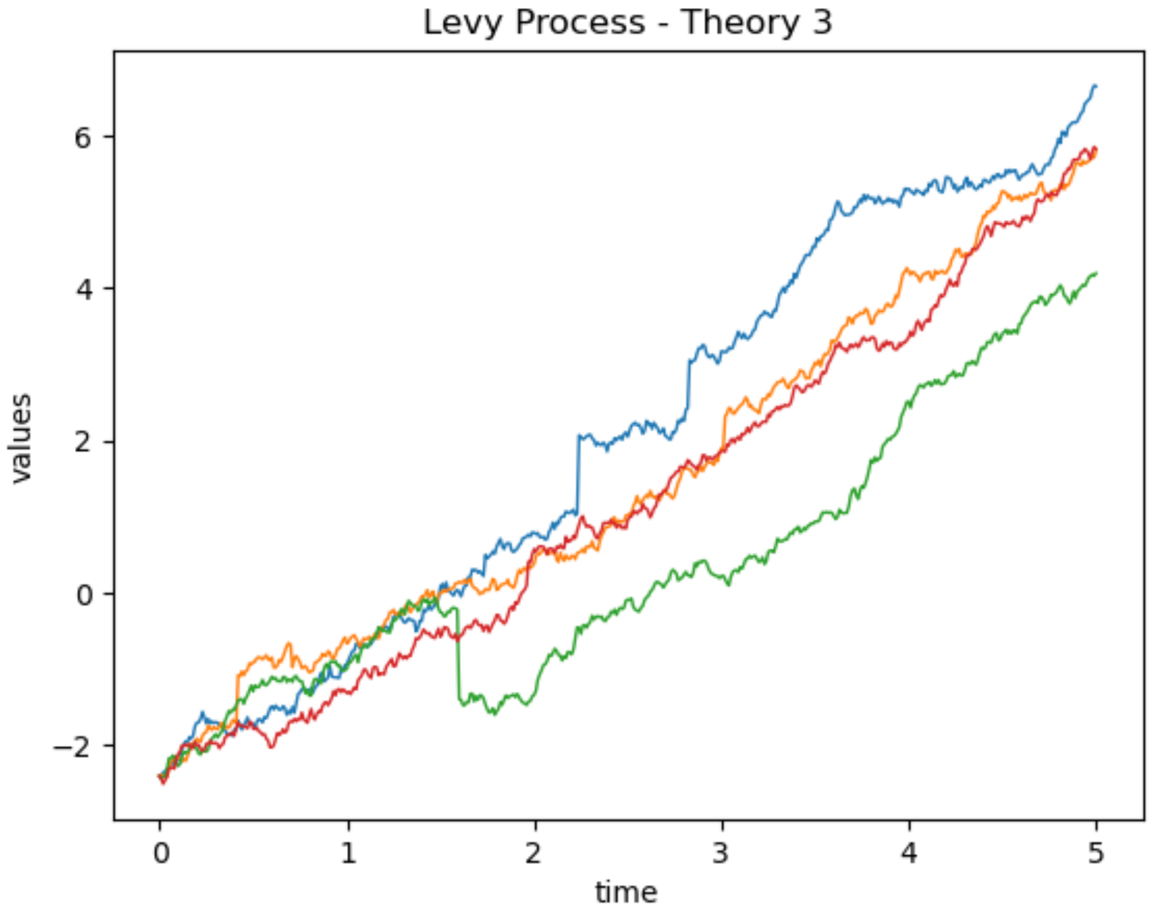
Theory 3: Take a Levy process $L = (L(t), t \ge 0)$ which is a sum of independent Brownian motion from Theory 2 and a compound Poisson process with intensity $\lambda = 0.4$ and jumps with Laplace distribution with mean 0.3 and standard deviation 1.2. Find the mean, variance, and the MGF of $L(t)$.

In [28]:
```
def simulate_compound_poisson_process(CPP_mu, CPP_sigma, intensity, N):
    jumpTimes= np.append(np.zeros(1),np.cumsum(np.random.exponential(scale=1
    compound_values=  np.append(np.zeros(1), np.random.laplace(loc=CPP_mu, s
    return compound_values, np.round(jumpTimes,2)

def Levy_sim(dt, T, BM_mu, BM_sigma, CPP_mu,CPP_sigma,intensity, x0):
    N = int(T/dt)
    BM_increments = np.append(x0,np.random.normal(BM_mu*dt, BM_sigma*np.sqrt
    CPP_increments, CPP_jumptimes = simulate_compound_poisson_process(CPP_mu

    time = np.round(np.linspace(0,T, N+1),2)
    for i,count in enumerate(CPP_jumptimes):
        if count> np.max(time):
            break
        try:
            idx = np.where(time==count)[-1][-1]
            BM_increments[idx] += CPP_increments[i]
        except (IndexError):
            print('No matching time jump for', count)
    L= np.cumsum(BM_increments)
    return L, time
```

In [29]:
```
dt = 0.01
N = 10000
T=5
sims=[]
np.random.seed(12345)
for i in range(N):
    a,b = Levy_sim(dt, T, BM_mu=1.5, BM_sigma=0.5, CPP_mu=0.3, CPP_sigma=1.2
    sims.append(a)
    time = b
```

In [30]:
```
np.random.seed(10000)
for i in np.random.choice(N, 4):
    plt.plot(time, sims[i], lw=1)

plt.xlabel('time')
plt.ylabel('values')
plt.title('Levy Process - Theory 3')
```

Out[30]:
```
Text(0.5, 1.0, 'Levy Process - Theory 3')
```



In [31]:
```
x= []
for i in range(N):
    x.append(find_value_at_T(sims[i],time,5))

print('E[X(5)]=', np.mean(x))
print('Var[X(5)]=', np.var(x))
```

```
E[X(5)]= 5.698926645489278
Var[X(5)]= 4.26190550193573
```

In [ ]: