Empirical Methods in Financial Econometrics: Project 1

Jinyuan Wu

September 2019

Exercise 1

A.

- n: The simulation has 80 steps per day.
- T: The number of days to simulate. Here we want to simulate 1.25 years with 252 business days per year.
- μ: The expected drift of the percentage of daily price change .
- σ:The variance of daily log-price change.

The annual return is $\mu_{ann.} = 252 * \mu = 9.53064\%$. The standard deviation for the annual return is $\sigma_{ann.} = \sqrt{n}\sigma = 0.175$

B.

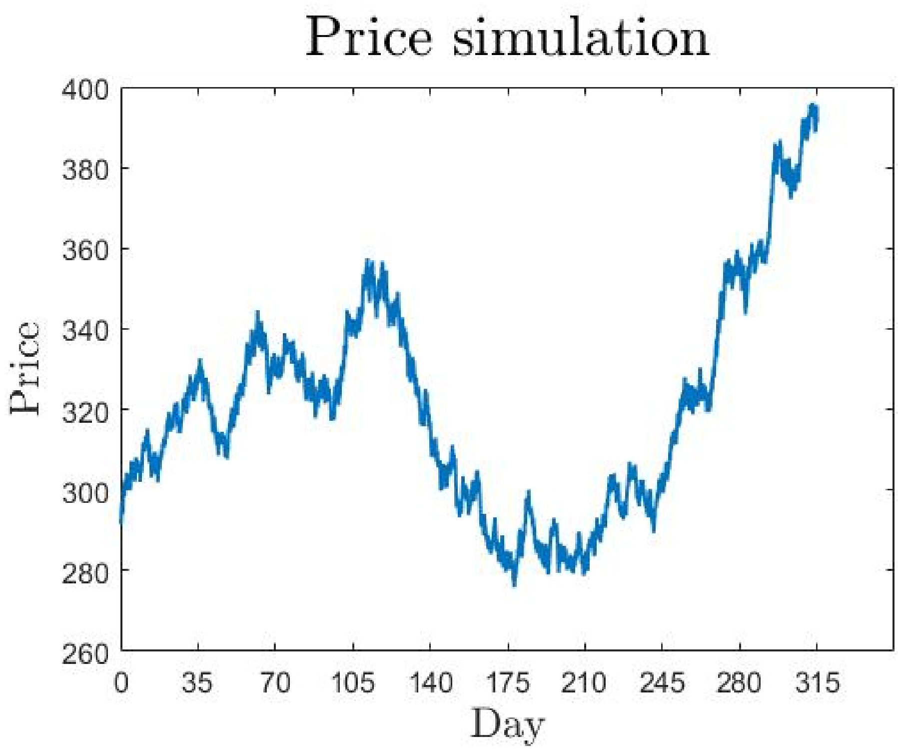

Figure 1: The time series of the prices

As shown in figure 1, the price first shows an increasing trend until reaching a local peak of about 360, then it plummet to lowest price of about 275. After being stable for a period, the price soars to the highest price of about 395.

Exercise 2

A.

- λ: The expected number of jumps per day.
- σ_j:The variance of the jump size.

σ_j is about twice as large as σ. The reason for a $\sqrt{n}$ in the denominator is to make the scale of it at the same level as the sample variance of X_t.

B.

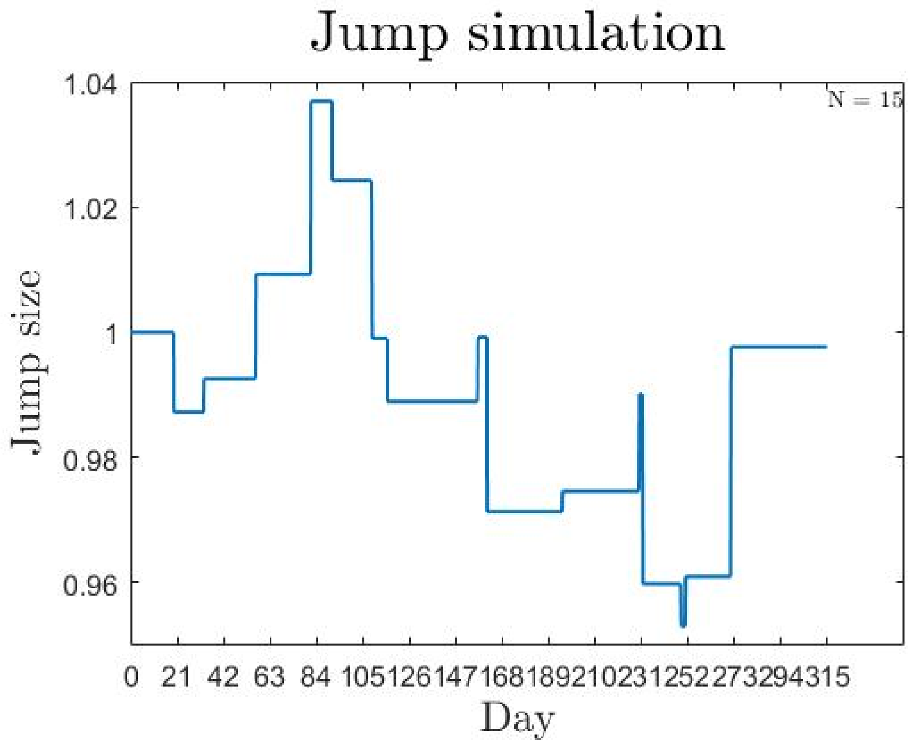

Figure 2: The simulated compound Poisson process

There are 15 jumps in this simulation. The jump sizes and the timing of the jumps are totally random.

Exercise 3

A.

By integrating both sides of jump-diffusion model with constant coefficients, we can get the result from choice 1. After taking the exponent of both sides, we can get choice 4. Therefore, choice 1 and 4 are correct.

B.

After combining the compound Poisson process, the price shows a similar trend as in exercise 1 . In addition, the price shows more volatility.

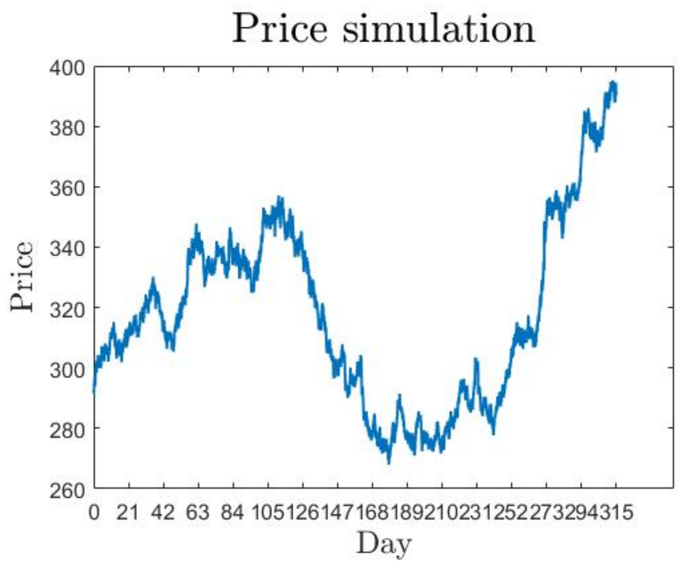

Figure 3: The time series of the prices

Exercise 4

A.

- ρ: The parameter deciding the scale of impact of deviating from the center(μ_c).
- μ_c: The center value of c.
- σ_c:The parameter deciding the variance of c_t.

B.

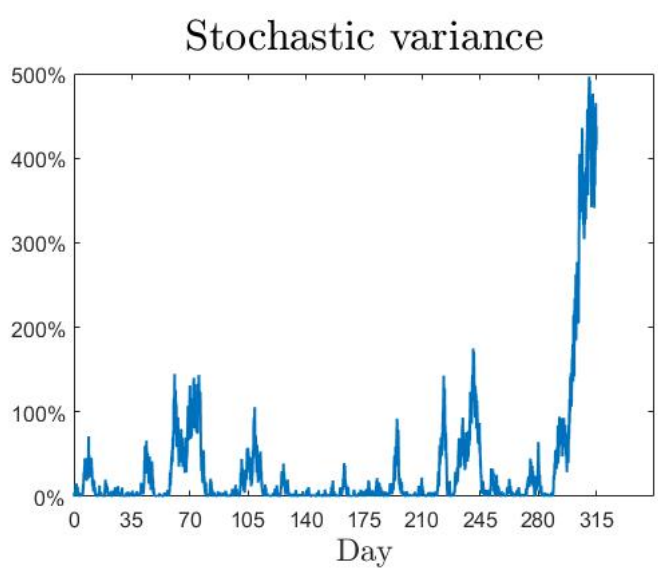

Figure 4: The ultra high-frequency values of c_j

Figure 4 shows a pattern of the concentration of volatility. There will be periods with constant high volatility while the volatility remains stale at other time. The volatility can go extreme as shown in the graph.

C.

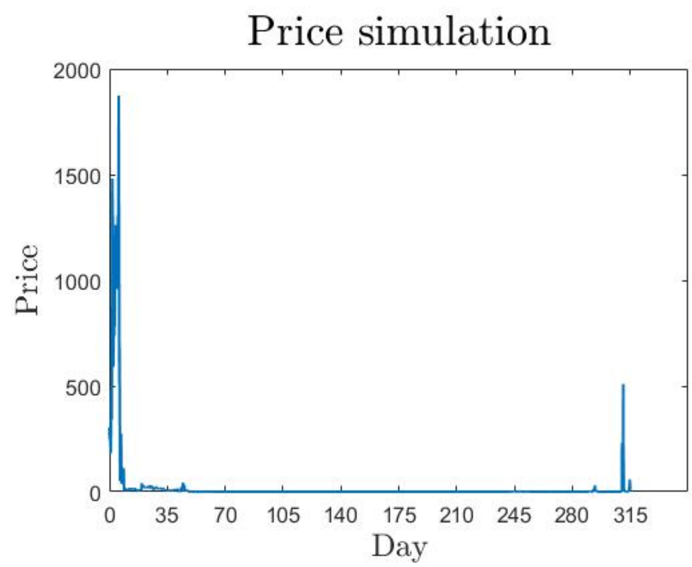

Figure 5: The time series of the prices

The volatility of price mainly shows in several time periods while the price remain stable at most of the time. The price can go extreme at certain period. In this simulation, the price soars up to about 1700 with the initial price of 292.58.

D.

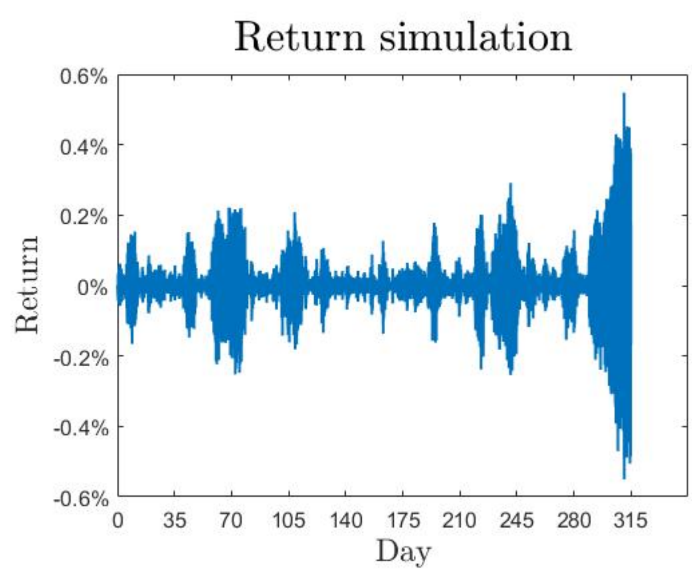

Figure 6: The ultra high frequency returns

Figure 6 shows a pattern of the concentration of volatility in return. There will be periods with constant high volatility while the volatility remains stale at other time.

E.

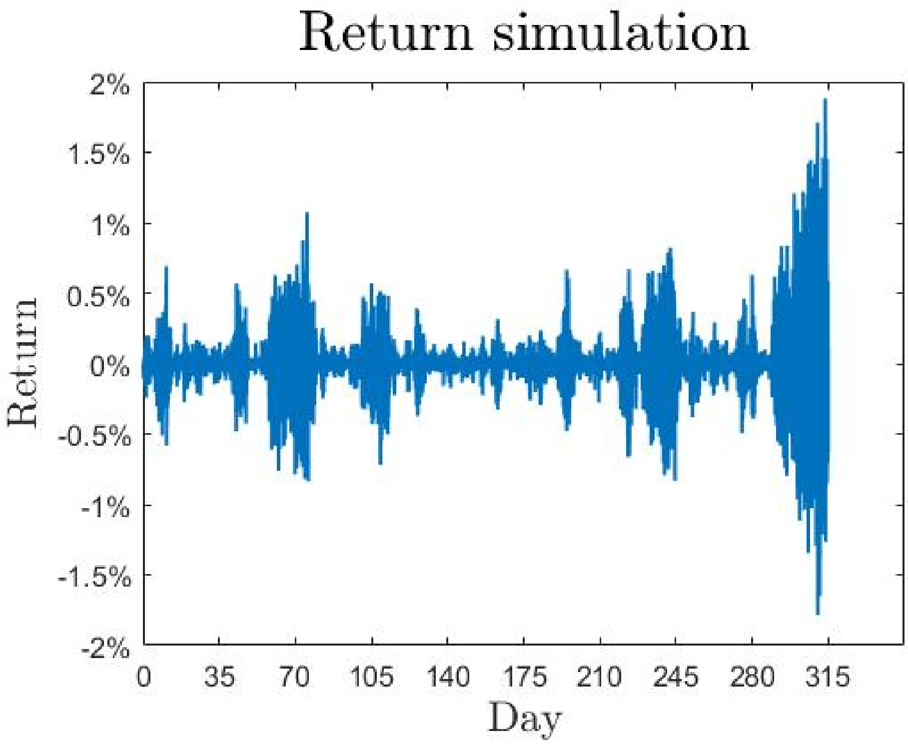

Figure 7: The high frequency returns

The graph shows a similar pattern of the concentration of volatility in return as the ultra high frequency returns.

Appendix A: Matlab code

Listing 1: exercise1.m solves Exercise 1 and creates figure 1 above.

```
clear
% Assign value to parameters
n = 80;
T = 1.25*252;
mu = 0.0003782;
sigma = 0.011;
delta = 1 / n;
steps = T*n;
%log-price matrix. The second column shows the date and only used for graphing.
X = zeros(steps,2);
%Random standard normal
Z = zeros(steps,1);
Z(1) = normrnd(0,1);
```

```

X(1,1) = log(292.58);
X(1,2) = 0;

%Generate price sequense
for i = 2:steps
    Z(i) = normrnd(0,1);
    X(i,1) = X(i-1,1) + mu*delta +  sqrt(sigma)*delta * Z(i);
    X(i,2) = i/80;
end
price = X;
price(:,1) = exp(X(:,1));
%Plot
plot(price(:,2),price(:,1))
title("Price simulation")
ylabel("Price")
xlabel("Day")
set(gca,'XTick',[0:35:315]);
```

Listing 2: exercise2.m solves Exercise 2 and creates figure 2 above.

```
% Assign value to parameters
n = 80;
T = 1.25*252;
sigma = 0.011;
lambda = 15/252;
sigmaj = 18*sigma*sqrt(1/n);
delta = 1 / n;
steps = T*n;
%Number of jumps
N = poissrnd(lambda*T);
%Sizes of jumps the timing of jumps
Y = zeros(N,1);
U = zeros(N,1);
for i = 1:N
    Y(i) = normrnd(0,sigmaj);
    U(i) = unifrnd(0,1);
end
%Compound Poisson process
J = zeros(steps,2);
for t = 1:steps
    for i = 1:N
    if  U(i)<= t/(steps)
      J(t,1) =  J(t,1) + Y(i,1);
    end
    end
    J(t,2) = t/80;
end
jump = J;
jump(:,1) = exp(J(:,1));
%Plot
plot(jump(:,2),jump(:,1))
title("Jump simulation")
ylabel("Jump size")
xlabel("Day")
```

```
set(gca,'XTick',[0:21:315]);
txt = sprintf('N = %i', N);
text(315,max(jump(:,1)),txt)
```

Listing 3: exercise3.m solves Exercise 3 and creates figure 3 above.

```
% after executing exercise1.m and exercise2.m
jprice = X;
jprice(:,1)= exp(X(:,1)+J(:,1));
%Plot
plot(jprice(:,2),jprice(:,1))
title("Price simulation")
ylabel("Price")
xlabel("Day")
set(gca,'XTick',[0:21:315]);
```

Listing 4: exercise4.m solves Exercise 4 and creates figure 4,5,6 and 7 above.

```
clear
% Assign value to parameters
n = 80;
T = 1.25*252;
nE = 20 * n;
ro = 0.03;
muc = 0.11^2;
sigmac = 0.001;
delta = 1 / n;
deltaE = 1 / nE;
steps = T*n;
stepsE = T*nE;
%Random standard normal
Zc = zeros(stepsE,1);
Zc(1) = normrnd(0,1);
% Stochastic variance(B)
C = zeros(stepsE,2);
C(1) = muc;
for j = 2:stepsE
    Zc(j) = normrnd(0,1);
    C(j,1) =  C(j-1,1) + ro*(muc-C(j-1,1))*deltaE + sqrt(C(j-1,1)*deltaE)*
        Zc(j);
    if C(j,1) < muc/2
        C(j,1) = muc/2;
    end
    C(j,2) = j/(80*20);
end
annC = C;
annC(:,1)= C(:,1)* sqrt(252);
%Plot
plot(annC(:,2),annC(:,1))
title("Stochastic variance")
xlabel("Day")
ytickformat(gca, 'percentage')
set(gca,'XTick',[0:35:315]);

%The ultra high frequency log-prices(C)
```

```matlab
Xt = zeros(stepsE,2);
Xt(1,1) = log(292.58);
Z = zeros(stepsE,1);
Z(1) = normrnd(0,1);
for j = 2:stepsE
    Z(j) = normrnd(0,1);
    Xt(j,1) =  Xt(j-1,1) + sqrt(C(j-1,1)*deltaE)*Z(j);
    Xt(j,2) = j/(80*20);
end
price = Xt;
price(:,1) = exp(Xt(:,1));
%Plot
figure
plot(price(:,2),price(:,1))
title("Price simulation")
ylabel("Price")
xlabel("Day")
set(gca,'XTick',[0:35:315]);

%return(D)
deltaXt = zeros(stepsE,2);
for i = 2:stepsE
   deltaXt(i,1) =  Xt(i,1)- Xt(i-1,1);
   deltaXt(i,2) = i/(80*20);
end
figure
plot(deltaXt(:,2),deltaXt(:,1))
title("Return simulation")
ylabel("Return")
xlabel("Day")
ytickformat(gca, 'percentage')
set(gca,'XTick',[0:35:315]);

%the log-returns for coarser frequency(E)
X = zeros(steps,2);
for i = 1:steps
   X(i,1) = Xt(i*nE/n);
end

deltaX = zeros(steps,2);
for i = 2:steps
   deltaX(i,1) =  X(i,1)- X(i-1,1);
   deltaX(i,2) = i/80;
end
figure
plot(deltaX(:,2),deltaX(:,1))
title("Return simulation")
ylabel("Return")
xlabel("Day")
ytickformat(gca, 'percentage')
set(gca,'XTick',[0:35:315]);
```