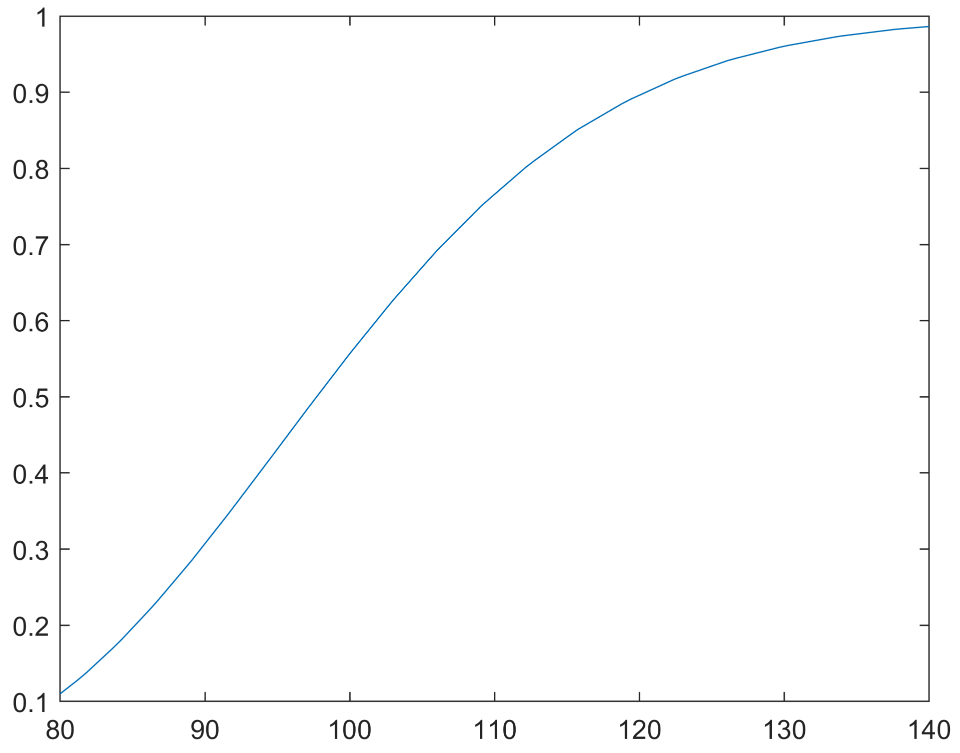
3a-c

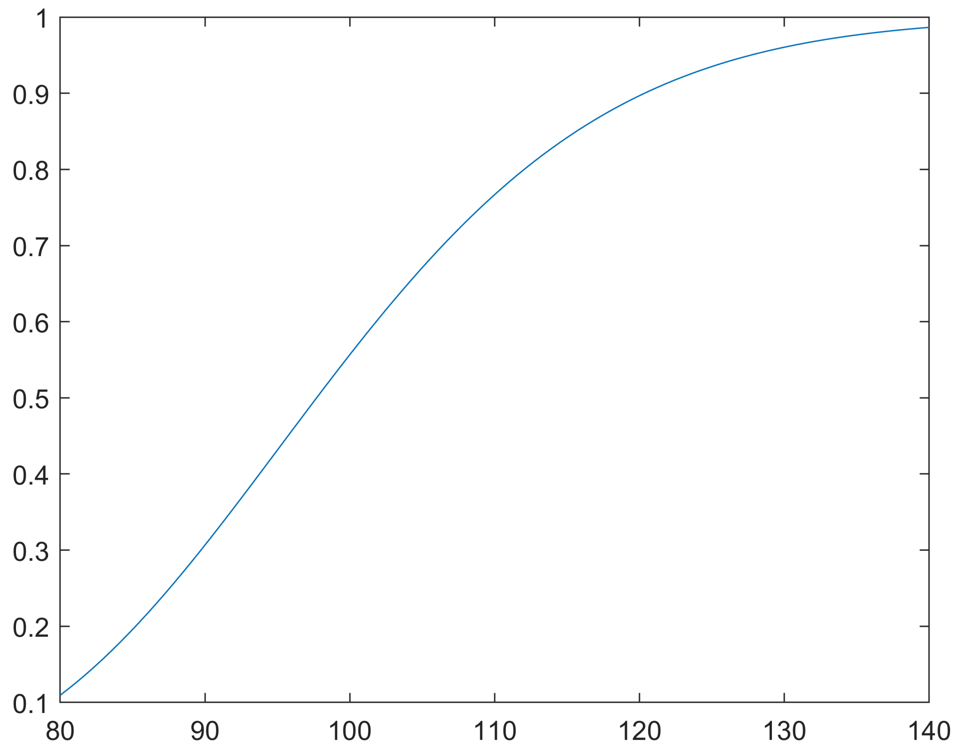
```
S0=100;
r=0.02;
sigma=0.23;
T=1;
N=250;
m=1;
K=100;

[~,s,delta]=binomialDeltaPowerCall(S0,r,sigma,T,N,m,K);

stocknew=linspace(80,140,100);

y=interpDelta(delta(1:N/2,N/2), s(1:N/2,N/2), stocknew);
plot(stocknew, y)
```



```
bls=blsdelta(stocknew,K,r,T/2,sigma)
```

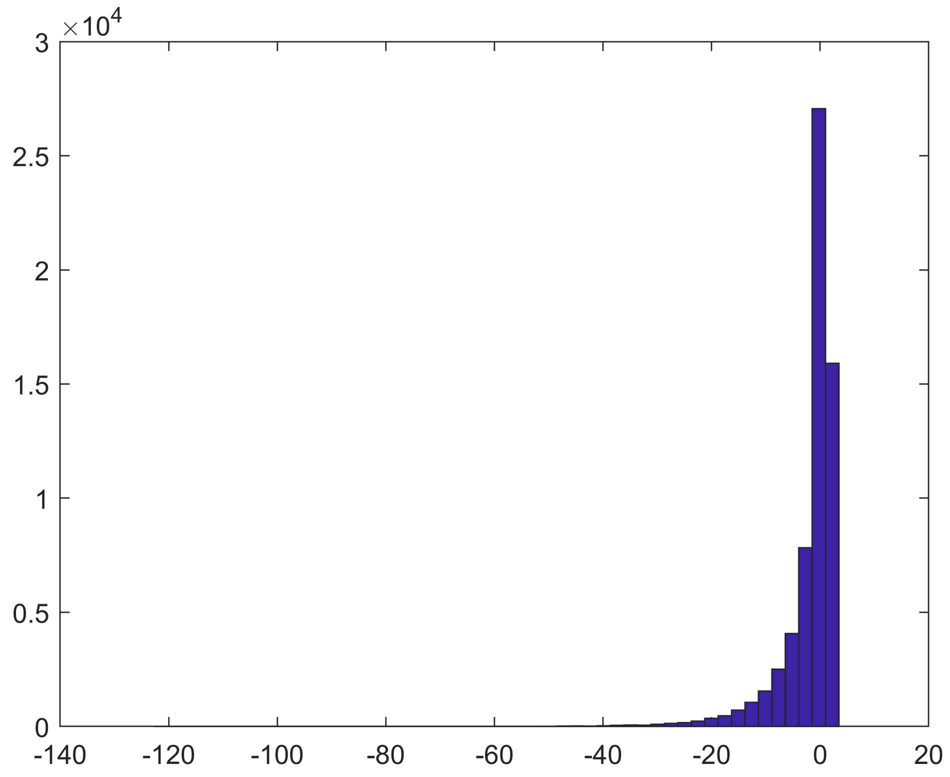
```
bls = 1×100
    0.1095    0.1184    0.1278    0.1376    0.1479    0.1585    0.1696    0.1810 ⋯
```

```
plot(stocknew, bls)
```

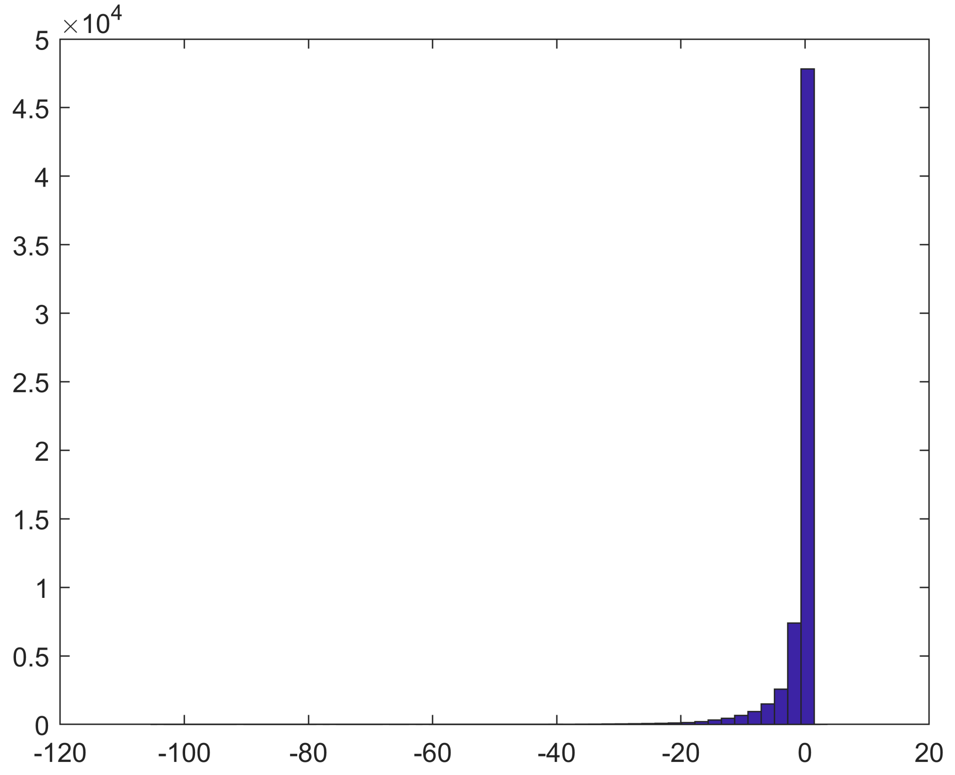


3d

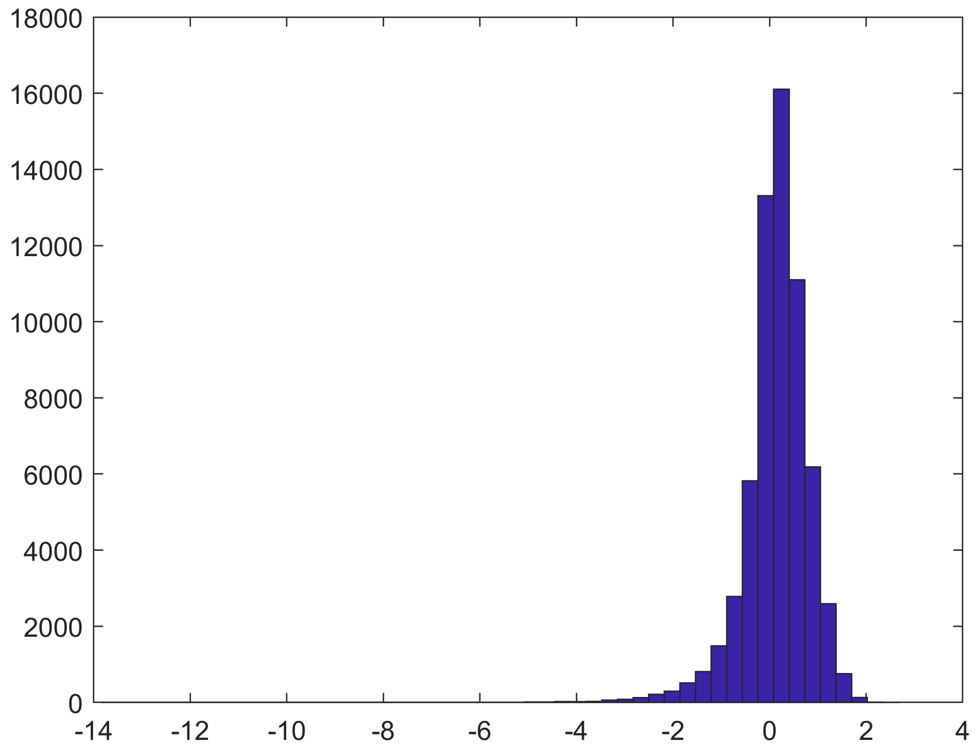
```
N=250;
dt = T/N;
% number of simulation is O((1/dt)^2)
M=N^2;
m=2;
mu=0.18;
beta = 0.05;
[v,s,delta]=binomialDeltaPowerCall(S0,r,sigma,T,N,m,K);

% data for drawing table
tabvar = zeros(5,1);
tabcvar = zeros(5,1);

% no hedging

% at beginning, we sell option, and hold bond=val(C0)
bond0 = v;
S=ones(M,1).*s(1,1);
for tstep = 1:N
    S=S + mu*dt*S + normrnd(0,1,M,1).*sqrt(dt).*sigma.*S;
end
netvaluef = bond0*exp(T*r) - max(S-K , 0).^m;
pandl = exp(-r*T).*netvaluef./v;
drawPandL(pandl)
```



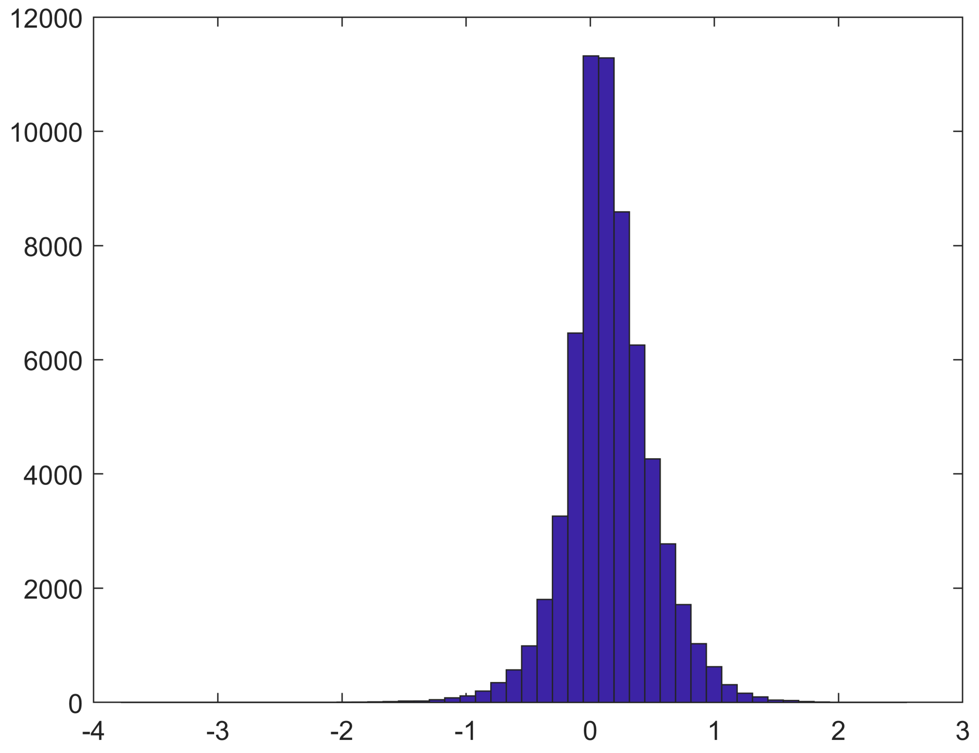
```
[tabvar(1,1),tabcvar(1,1)]=dVaRCVaR(pandl, beta);


% hedge once
% setting hedging freqency to once every N+2 time just means never hedge
% because we only have N timesteps
netvaluef=dynamichedging(v,s,delta, mu, sigma, T, N, M, N+2,K,m,r);
pandl = exp(-r*T).*netvaluef./v;
drawPandL(pandl)
```



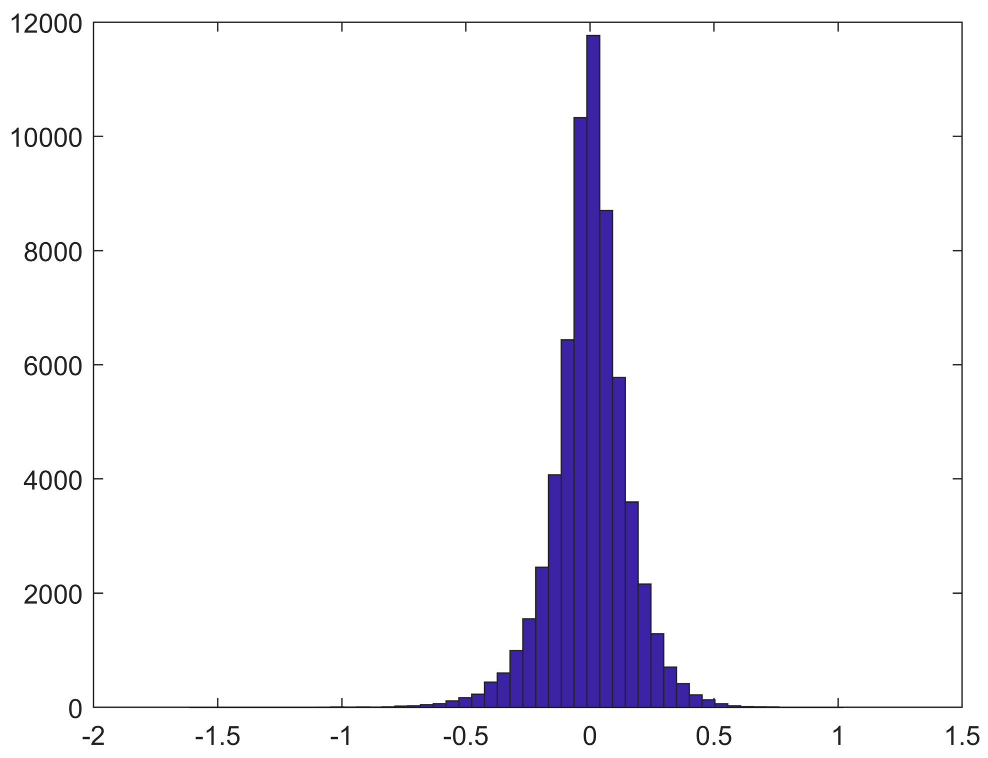
```
[tabvar(2,1),tabcvar(2,1)]=dVaRCVaR(pandl, beta);


% hedge every month (once every 20 timesteps)
netvaluef=dynamichedging(v,s,delta, mu, sigma, T, N, M, 20,K,m,r);
pandl = exp(-r*T).*netvaluef./v;
drawPandL(pandl)
```



```
[tabvar(3,1),tabcvar(3,1)]=dVaRCVaR(pandl, beta);


% hedge every week (once every 5 timesteps)
netvaluef=dynamichedging(v,s,delta, mu, sigma, T, N, M, 5,K,m,r);
pandl = exp(-r*T).*netvaluef./v;
drawPandL(pandl)
```



```
[tabvar(4,1),tabcvar(4,1)]=dVaRCVaR(pandl, beta);

% hedge every day
netvaluef=dynamichedging(v,s,delta, mu, sigma, T, N, M, 1,K,m,r);
pandl = exp(-r*T).*netvaluef./v;
drawPandL(pandl)
```



```
[tabvar(5,1),tabcvar(5,1)]=dVaRCVaR(pandl, beta);

frequency = {'0';'1';'5';'20';'everyday'};

table(tabvar, tabcvar, 'RowNames',frequency)
```

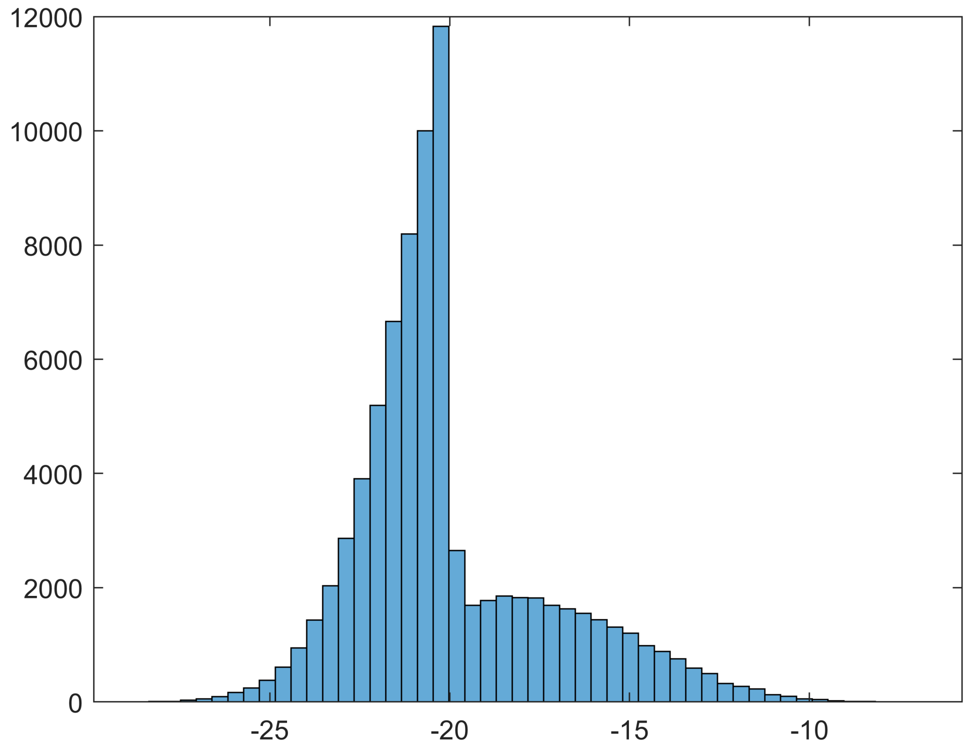
ans = 5×2 table

	tabvar	tabcvar
1 0	-12.2133	-20.3517
2 1	-7.2329	-14.4877
3 5	-0.9956	-1.7236
4 20	-0.3716	-0.6214
5 everyday	-0.2547	-0.3771

4

```
ns = [100,200,400,800];
cmean = zeros(4,1);
var = zeros(4,1);
cvar  = zeros(4,1);
cstd  = zeros(4,1);
```

```
tab = zeros(4,4);
for n = 1:4
    [cmean(n,1), var(n,1), cvar(n,1), cstd(n,1)] =  naivehedging(ns(n));
end
```



```
timesteps = {'100';'200';'400';'800'};

[a,b,c,d ] = naivehedging(200)
```

```
a = -20.0045
b = -23.5550
c = -24.4721
d = 2.7298
```

```
table(cmean, var, cvar, cstd, 'RowNames',timesteps)
```

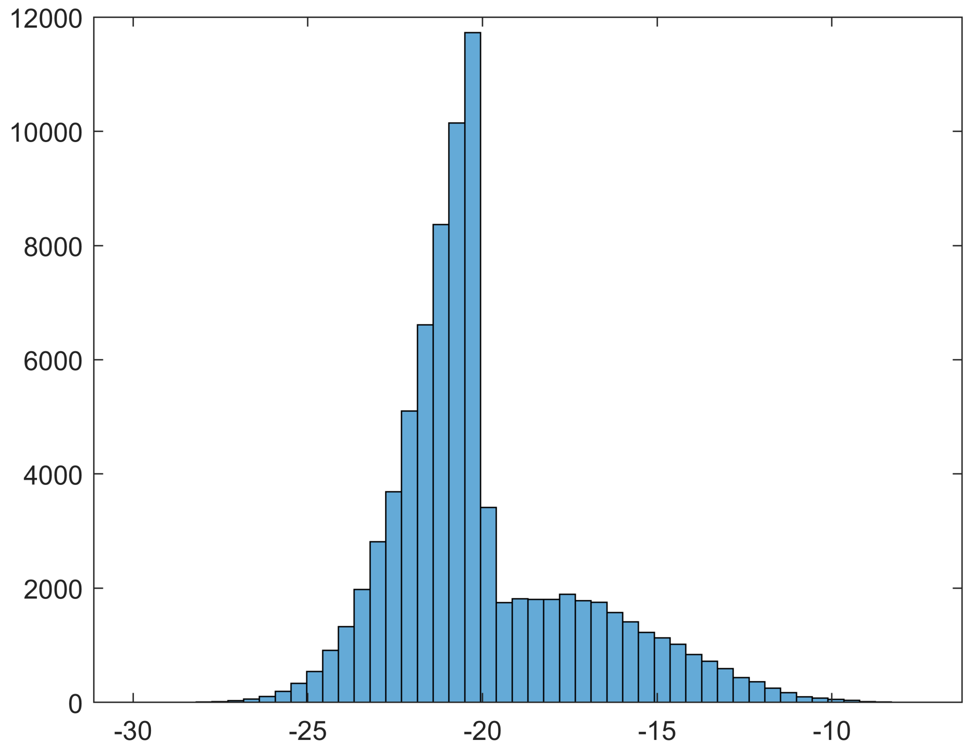
ans = 4×4 table

	cmean	var	cvar	cstd
1 100	-20.0118	-23.5496	-24.4621	2.7409
2 200	-19.9883	-23.5421	-24.4337	2.7354
3 400	-20.0119	-23.5546	-24.4680	2.7316
4 800	-20.0016	-23.5221	-24.4345	2.7211

```
    S0=95;
    K=105;
    M=80000;
    T=1;
    delta=zeros(M,1);
    S=ones(M,1).*S0;
    N=100
```

```
N = 100
```

```
    dt=T/N;
    r=0;
    mu =0.18 ;
    sigma = 0.23;
    C0=blsprice (S0,K,r,T,sigma);

    bond0=ones(M,1).*C0;
    for timestep = 1:N
        Snew = S.*exp((mu-0.5*sigma^2)*dt + sigma.* normrnd(0,1,M,1)*sqrt(dt));
        indicator = (Snew>=K)-(S>=K);
        delta = delta+indicator;
        bond0 = bond0 - indicator.*Snew;
        S=Snew;
    end
    netvalue = (bond0+S.*delta-max(0,S));
    pandl = netvalue./C0;
    [var,cvar] = dVaRCVaR(pandl, 0.05);
    stdscore = (std(pandl));
    histogram(pandl, 50)
```



```
    pandl = mean(pandl)
```

```
pandl = -20.0227
```

```
C0=blsprice (S0,K,r,T,sigma);
```

```
sigma=0.2;
r=0.05;
K=95;
S0=95 ;
muu=0.32 ;
mud= 0.3;
pup = 0.4 ;
lambda = 0.1;
N=800;
dt=T/N ;
kappa = 0;
for timestep = 1:N
    jumpmask = rand(1)<lambda*dt ;
    S=S+S*(r-lambda*kappa);
end
```

```
function [V ,S_mat, delta_mat] = binomialDeltaPowerCall(S0, r, sigma, T, N, m, K)
    % S_mat[i,j] indicates the stock price at timestep i,
    % given stock price went down j times
    % delta_mat[i,j] indicates the amount of stock required at timestep i,
    % in order to hedge the option, given stock price went down j times.
    dt = T/(N);
    % up and down ratio
    u=exp(sigma*sqrt(dt));
    d=1/u;
    % up probability
    q = (exp(r*dt)-d)/(u-d);
    S_mat = zeros(N+1,N+1);
    delta_mat = zeros(N+1,N+1);
    % stock price at T
    S_mat(1:N+1,N+1) = S0*u.^(N:-1:0).*d.^(0:N);
    % option value at T
    V = max(S_mat(1:N+1,N+1)-K , 0).^m;
    % backward iterations:
    for t = N:-1:1
        S_mat(1:t,t) = exp(-r*dt)* (S_mat(1:t,t+1)*(q)+S_mat(2:t+1,t+1)*(1-q));
        v_change = V(1:t)-V(2:t+1);
        % we need dV/dS stock to hedge the option
        delta_mat(1:t,t) = v_change./(S_mat(1:t,t+1)-S_mat(2:t+1,t+1));
        V = exp(-r*dt)* (V(1:t)*(q)+V(2:t+1)*(1-q));
    end
end

function [ret] = interpDelta(deltan, Sn, S)
    % add fake value to Sn and deltan, so that for stock price out of
    % bound, we get the value of delta corresponding to the closest Stock
    % price.
    % NOTE: here we assume Sn is increasing! otherwise this function does
    % not work
    Sn=[0;Sn;100^100];
    deltan=[deltan(1,1);deltan;deltan(length(deltan),1)];
    ret=interp1(Sn,deltan,S);
end

function netvaluef = dynamichedging(v,s, delta, mu, sigma, T, N, M, freq,K,m,r)
    % freq means we trade once every freq timesteps
    % we would hedge once at t0 no matter what

    %input: s, delta are matrix of stock price and delta hedging from
    %binomial lattice, other inputs are numbers

    dt = T/N;
    % stock price at beggining is given
    Si = s(1,1);
    S= ones(M,1)*Si;
    % at t0, we do the hedging, getting our initial delta vector
    deltaold=ones(M,1).* delta(1,1);
    % prepare the bond such that initial profolio value is 0
    bond = ones(M,1).* ( v-delta(1,1)*Si );
    % all bond, deltaold, S are vectors of length M, each indice indicate
    % one simulation.

    for tstep = 1:N
```

```
        S=S + mu*dt*S + normrnd(0,1,M,1).*sqrt(dt).*sigma.*S;
        if mod(tstep, freq)==0
            newdelta = interpDelta(delta(1:tstep, tstep), s(1:tstep, tstep), S);
            % bond get interest rate for time dt, then we buy stock
            bond = bond.*exp(r*dt) - (newdelta-deltaold) .* S ;
            deltaold=newdelta;
        end
    end
    netvaluef =  bond.*exp(r*dt) - max(S-K , 0).^m + deltaold .* S;
end

function drawPandL(pandl)
    % convinient function to draw histograms
    minv=(min(pandl));
    maxv=(max(pandl));
    range = minv: (maxv-minv)/50:maxv;
    counts = histc(pandl , range);
    bar(range, counts,'histc')
end

function [var,cvar] = dVaRCVaR(pandl, beta)
    n=length(pandl);
    pandl = sort(pandl);
    var = pandl(round(beta*n));
    cvar = mean(pandl(1:round(beta*n)));
end

function [pandl, var, cvar, stdscore] = naivehedging(N)
    S0=95;
    K=105;
    M=80000;
    T=1;
    delta=zeros(M,1);
    S=ones(M,1).*S0;
    dt=T/N;
    r=0;
    mu =0.18 ;
    sigma = 0.23;
    C0=blsprice (S0,K,r,T,sigma);

    bond0=ones(M,1).*C0;
    for timestep = 1:N
        Snew = S.*exp((mu-0.5*sigma^2)*dt + sigma.* normrnd(0,1,M,1)*sqrt(dt));
        indicator = (Snew>=K)-(S>=K);
        delta = delta+indicator;
        bond0 = bond0 - indicator.*Snew;
        S=Snew;
    end
    netvalue = (bond0+S.*delta-max(0,S));
    pandl = netvalue./C0;
    [var,cvar] = dVaRCVaR(pandl, 0.05);
    stdscore = (std(pandl));
    if (N==800)
        histogram(pandl, 50)
    end
    pandl = mean(pandl);

end
```