PRMIA Risk Management Challenge
--------- MATLAB Challenge Livescript

Krushna Mehta **Qiwen Wang** **Yuanhao Feng** **Jiewei Wu**

Positions

First, the market data obtained from exchange market API should be updated tickly for traders to monitor. Here we just read the relevant data from csv files for outlook.

```
data = readtable('position.csv','PreserveVariableNames',true);
fig = uifigure('Position',[500 500 750 350]);
uit = uitable(fig);
uit.Position = [20 20 710 310];
uit.Data = data;
data = table2array(data(:,2:9));
uit.RowName = 'numbered';
[row, col] = find(data(:,8) > 0);
col = col + 8;
s = uistyle('BackgroundColor','green');
addStyle(uit,s,'cell',[row,col]);
[row, col] = find(data(:,8) < 0);
col = col + 8;
s = uistyle('BackgroundColor','red');
addStyle(uit,s,'cell',[row,col]);
```

Performance Analysis | Risk Management

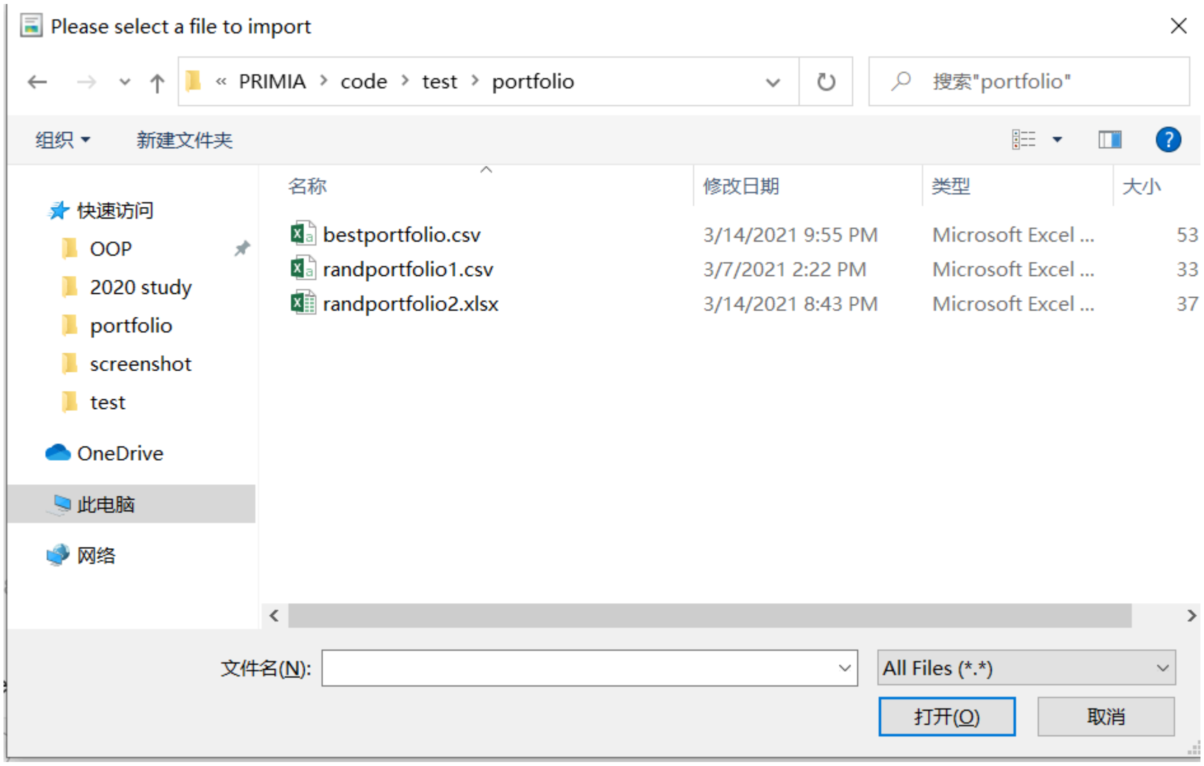
Positions

Contract	Bid	Ask	Strike	SOD	BuyQty	SellQty	NetPos	P/L	Delta	Theta	Mid volitality	Gamma	Vega
February10 C1.1975	0.0073	0.0080	1.1975	-10	0	4	-6	2320	93.56%	-0.02%	6.11%	34.24%	0.01%
February10 C1.2000	0.0050	0.0058	1.2000	5	2	0	7	2320	85.46%	-0.04%	5.81%	58.42%	0.02%
February10 C1.2025	0.0015	0.0038	1.2025	32	0	3	35	13100	71.80%	0.04%	5.57%	85.42%	0.03%
February10 C1.2050	0.0015	0.0023	1.2050	20	4	3	27	8840	50.62%	-0.05%	5.42%	99.42%	0.04%
February10 C1.2075	0.0006	0.0013	1.2075	-7	0	0	-7	4320	28.96%	-0.04%	5.45%	85.36%	0.03%
February10 C1.2100	0.0001	0.0008	1.2100	23	2	45	66	2520	14.38%	0.03%	5.58%	58.12%	0.02%
March17 P1.1975	0.0013	0.0021	1.1975	34	-6	0	28	-1200	-22.30%	-0.02%	6.00%	33.62%	0.08%
March17 P1.2000	0.0020	0.0027	1.2000	23	0	0	23	1540	29.47%	0.02%	5.90%	38.64%	0.07%
March17 P1.2025	0.0028	0.0036	1.2025	-2	5	23	26	1325	-38.19%	-0.02%	5.81%	42.42%	0.07%
March17 P1.2050	0.0039	0.0047	1.2050	-5	3	0	-2	1240	-48.28%	-0.02%	5.78%	44.13%	0.08%
March17 P1.2075	0.0052	0.0060	1.2075	4	0	3	7	2400	58.66%	0.02%	5.74%	43.24%	0.07%
March17 P1.2100	0.0068	0.0076	1.2100	32	4	0	36	3200	-68.02%	-0.02%	5.77%	39.98%	0.07%
April24 C1.1975	0.0103	0.0110	1.1975	-20	0	11	-9	2320	72.78%	-0.01%	5.92%	27.96%	0.09%
April24 C1.2000	0.0085	0.0093	1.2000	100	0	12	112	2540	66.73%	0.02%	5.85%	30.45%	0.09%
April24 C1.2025	0.0070	0.0077	1.2025	15	0	0	15	-3400	50.82%	-0.02%	5.80%	32.25%	0.10%

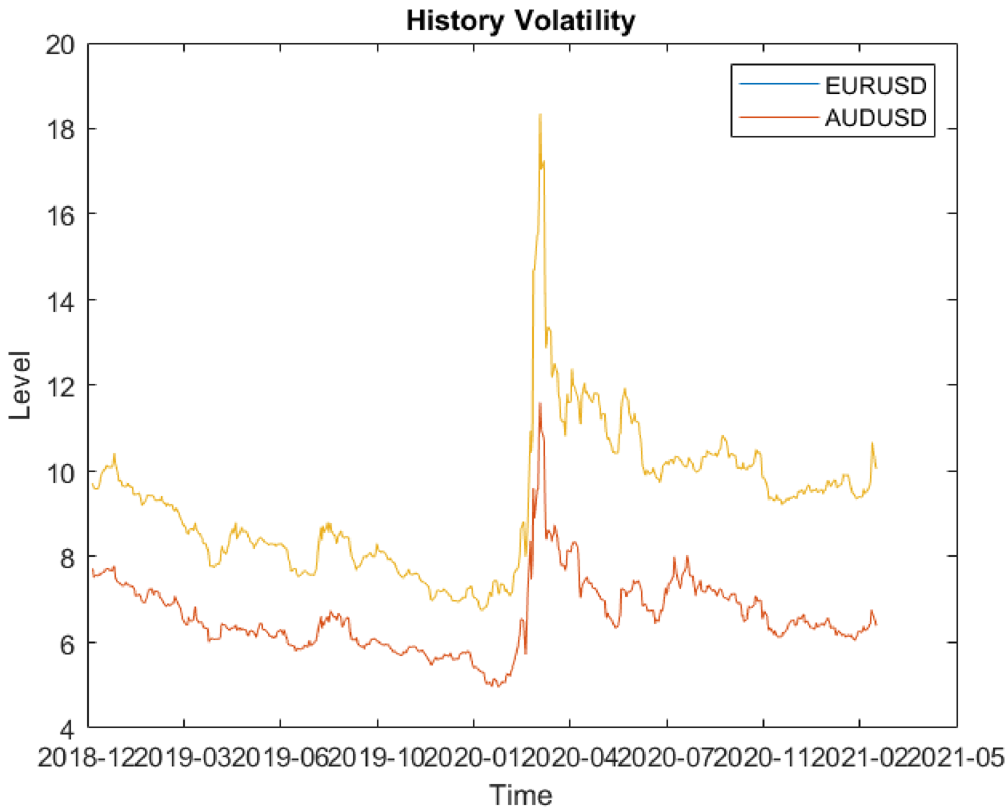
Performance

Once run the above code, A winow of uitable of this will be poped up. The P&L column is colored for visibility.

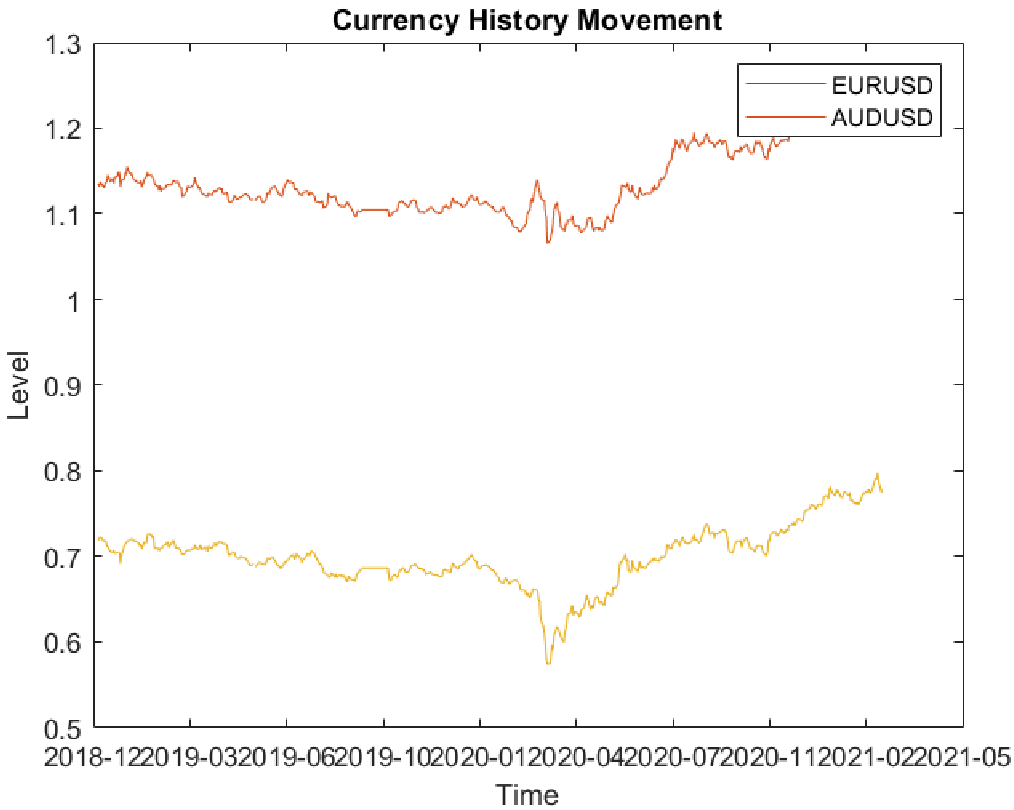
By clicking the Load botton,the code will ask you to select your portfolio file. Once you select the file(which is a csv or excel file), the code will read the position and the currency price and volatility data.




```
[file_name, file_path] = uigetfile('*.*', 'Please select a file to import');
positionS = readtable(convertCharsToStrings(file_path) + "\" +...
convertCharsToStrings(file_name),'PreserveVariableNames',true);

position = positionS;
file_name = split(convertCharsToStrings(file_name),'.');
portfolio_name = file_name(1);

vol_data = readtable('currency_impliedvol.csv','PreserveVariableNames',true);
vol_table = vol_data;
date = datenum(vol_data.Date);
figure(1);
plot(date, vol_data.EURUSD);
hold on;
plot(date, vol_data.AUDUSD);
datetick('x', 'yyyy-mm', 'keepticks');
legend('EURUSD','AUDUSD');
title('History Volatility');
xlabel('Time')
ylabel('Level');
```




```
currency_data = readtable('Currency.csv','PreserveVariableNames',true);
currency_table = currency_data;
date = datenum(currency_data.Date);
figure(2)
plot(date, currency_data.EURUSD);
hold on;
plot(date, currency_data.AUDUSD);
datetick('x', 'yyyy-mm', 'keepticks')
legend('EURUSD','AUDUSD');
title('Currency History Movement');
xlabel('Time')
ylabel('Level');
```



Performance

Reinvest

```
Reinvest = true;
```

Transaction Cost

```
TransactionCost = false;
```

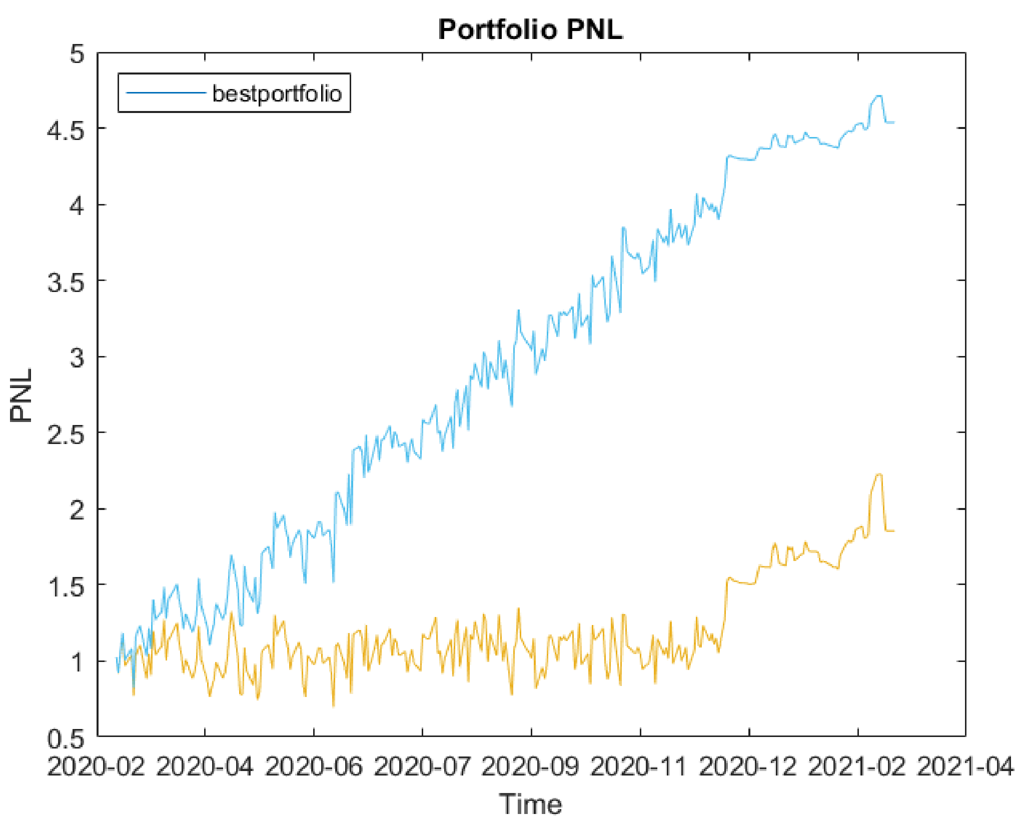
Apply Botton

```
 ;
```

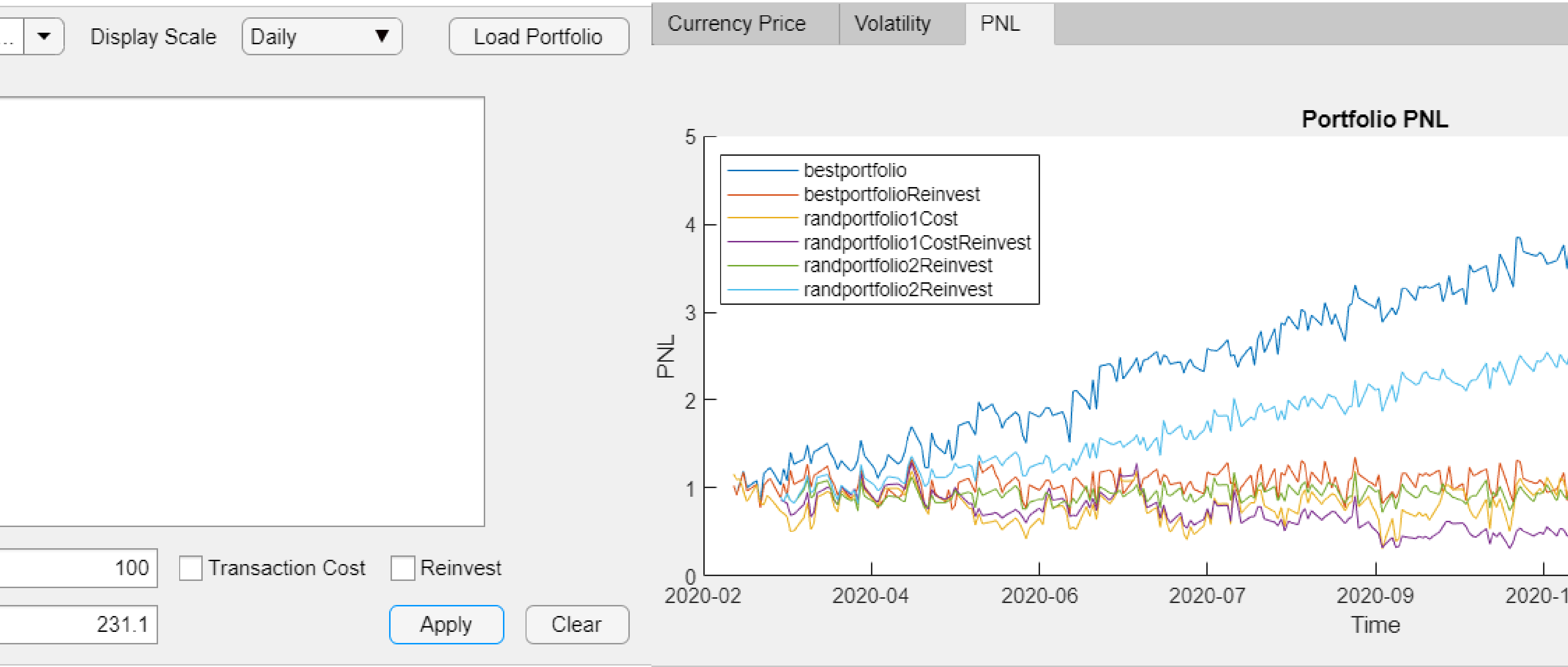
By choosing reinvest and transaction cost or not, by clicking the apply botton, we are able to simulate our previous uploaded portfolio.

```
price = readtable('OptionPrice.csv','PreserveVariableNames',true);
difference = diff(price{:,2:end});
price{:,2:end}(ismissing(price{:,2:end})) = 0;
daily_capital = sum(price{:,2:end} .* position{:,2:end},2);
price{1:end-1,2:end} = difference;
price{:,2:end}(ismissing(price{:,2:end})) = 0;
price{end,2:end} = 0;
pnl = price{:,2:end} .* position{:,2:end};
```

```
if TransactionCost == 1
    app.portfolio_name = app.portfolio_name + "Cost";
    pnl(1:end-1,:) = pnl(1:end-1,:) - ...
        0.02*abs(diff(app.position{:,2:end}).*price{1:end-1,2:end});
end
daily_pnl = sum(pnl,2);
daily_return = daily_pnl./daily_capital;
acc_pnl = 1 + cumsum(daily_return);
date = datenum(price.Date);


figure(3)
plot(date, acc_pnl);
hold('on');
datetick('x', 'yyyy-mm', 'keepticks');
legend(sprintf("%s", portfolio_name),'Location','northwest');
title('Portfolio PNL');
xlabel('Time');
ylabel('PNL');
hold('on');
```



In the dashboard, we also realized start and end date selection, multiple portoflio draw in one plot, and multi-labels interface. Numerically, we also calculated final capital for initialized capital and performance statistics which are necessary to analyze a portfolio, which looks like follows. I emphasize these points in the video demo.

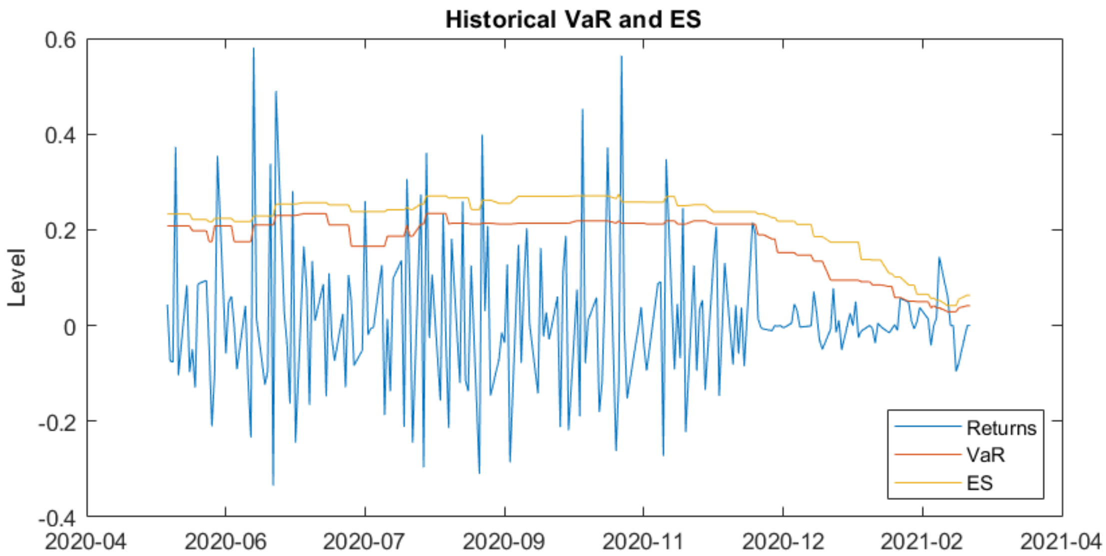


VaR and ES Backtesting

With the daily P&L obtained from the previous section, we can do VaR and ES backtesting.

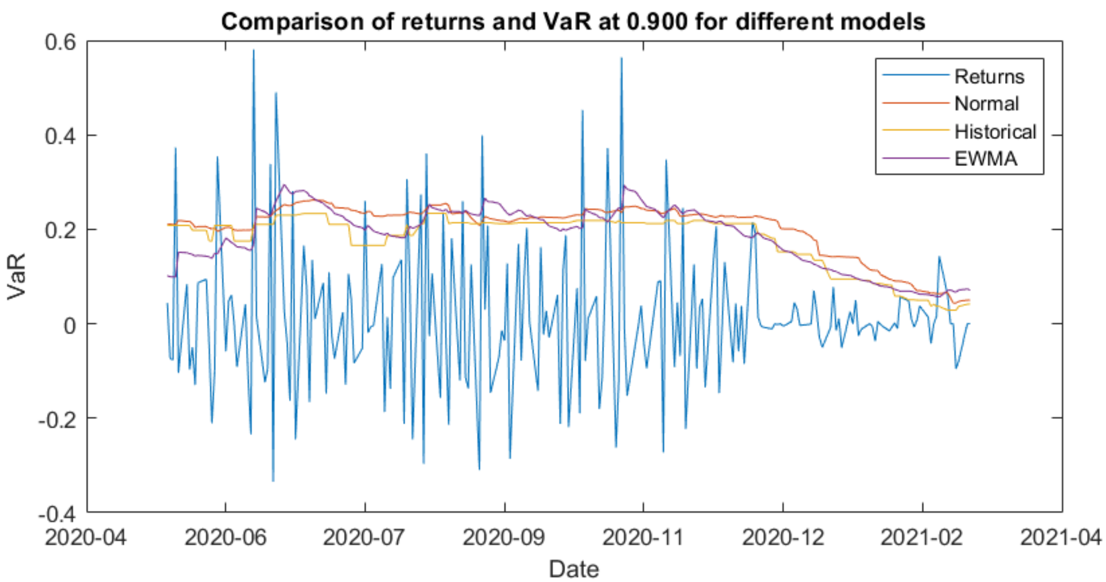
```
VaRLevel = 0.9;
VaRLevel = 0.9;
 ;
Returns = daily_return;
DateReturns = price.Date;
TestWindowStart = 1;
TestWindowEnd = length(DateReturns);

EstimationWindowSize = 50;
TestWindow = TestWindowStart+EstimationWindowSize:TestWindowEnd;


DatesTest = DateReturns(TestWindow);
ReturnsTest = Returns(TestWindow);


VaR_Hist = zeros(length(TestWindow) ,1);
ES_Hist = zeros(length(TestWindow),1);

for t = TestWindow

   i = t - TestWindowStart + 1;
   EstimationWindow = t-EstimationWindowSize:t-1;

   [VaR_Hist(i - EstimationWindowSize),ES_Hist(i - EstimationWindowSize)]...
       = hHistoricalVaRES(Returns(EstimationWindow),VaRLevel);
end
figure(4)
```

```
date_ = datenum(DatesTest);
plot(date_,[ReturnsTest,-VaR_Hist,-ES_Hist])
set(gcf,'position',[500,500,750,350]);
datetick('x', 'yyyy-mm', 'keepticks');
legend({'Returns','VaR','ES'},'Location','southeast')
title('Historical VaR and ES')
ylabel('Level')
```




```


%-----------------------------------------------------------------------------------------
pVaR = 1 - VaRLevel;
Zscore   = norminv(pVaR);
Normal95 = zeros(length(TestWindow),1);

for t = TestWindow
    i = t - TestWindowStart + 1;
    EstimationWindow = t-EstimationWindowSize:t-1;
    Sigma = std(Returns(EstimationWindow));
    Normal95(i - EstimationWindowSize) = -Zscore(1)*Sigma;
end

Lambda = 0.94;
Sigma2     = zeros(length(Returns),1);
Sigma2(1)  = Returns(1)^2;

for i = 2 : (TestWindowStart-1)
    Sigma2(i) = (1-Lambda) * Returns(i-1)^2 + Lambda * Sigma2(i-1);
end
Zscore = norminv(pVaR);
EWMA95 = zeros(length(TestWindow),1);

for t = TestWindow
    k     = t - TestWindowStart + 1;
    Sigma2(t) = (1-Lambda) * Returns(t-1)^2 + Lambda * Sigma2(t-1);
    Sigma = sqrt(Sigma2(t));
```

```
    EWMA95(k - EstimationWindowSize) = -Zscore(1)*Sigma;
end
ReturnsTest = Returns(TestWindow);
DatesTest   = DateReturns(TestWindow);
date_ = datenum(DatesTest);
figure(5)
plot(date_,[ReturnsTest Normal95 -VaR_Hist EWMA95])
set(gcf,'position',[500,500,750,350])
datetick('x', 'yyyy-mm', 'keepticks');
ylabel( 'VaR')
xlabel('Date')
legend({'Returns','Normal','Historical','EWMA'},'Location','Northeast')
title(sprintf('Comparison of returns and VaR at %.3f for different models', VaRLevel))
```



Also, the functionality to change the start and end date and the signal lamp are included in the dashboard for comprehensiveness.

Risk Matrix

The risk matrix is used to measure the sensitiveness of risk metrics with regard to the percentage change of underlying price and volatility. The users should be able to select different metrics, which they think, are relevant to the decision making. Different scales of changes are also included for the traders to monitor risks and get prepared for the worst senarios. Here, because the ListBox for multiple selection is not allowed, we leave the instructions in the demo video.

```
UnderlyingScale = "-10,-5,0";
VolScale = "0,5,10,-5";
data = readtable('matrix.csv','PreserveVariableNames',true);
und = UnderlyingScale;
und_array = split(und, ",");
len = length(und_array);
und_int = zeros(len,1);
for index = 1:len
    und_int(index) = str2num(und_array{index});
```

```
end
cond = ismember(data.("Underlying Pct"),und_int);
data = data(cond,:);
und = VolScale;
und_array = split(und, ",");
len = length(und_array);
und_int = zeros(len,1);
for index = 1:len
    und_int(index) = str2num(und_array{index});
end
cond = ismember(data.("Vol Pct"),und_int);
data = data(cond,:);
```

By adding the UI control, the matrix will be look like as follows:

Risk Matrix

Underlying Scale: -10,-5,0 | Vol Scale: 0,5,10,-5 | Risk Metrics: Gamma, Theta, Vega | Update

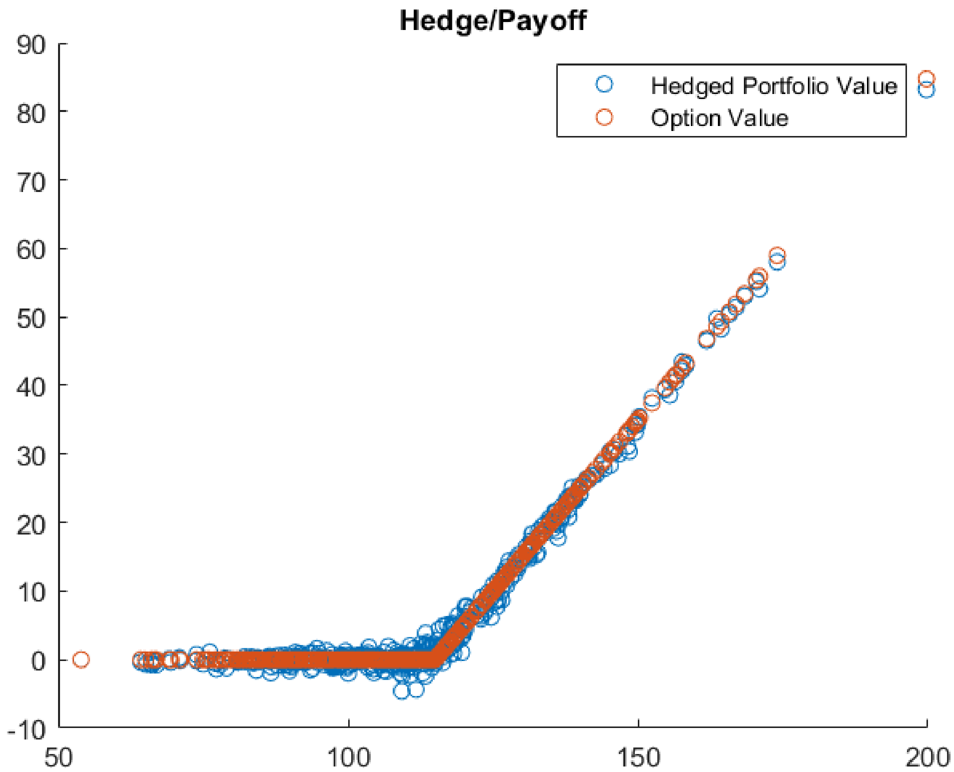
Underlying Pct	Vol Pct	PNL	Delta	Gamma	Theta	Vega
-10	-5	2320	44.819%	36.272%	-0.020%	0.062%
-10	0	13160	47.178%	38.181%	-0.021%	0.066%
-10	5	8840	49.537%	40.090%	-0.022%	0.069%
-10	10	4320	51.896%	41.999%	-0.023%	0.072%
-5	-5	-1200	47.309%	38.287%	-0.021%	0.066%
-5	0	-1540	49.799%	40.302%	-0.022%	0.069%
-5	5	1325	52.289%	42.317%	-0.023%	0.073%
-5	10	1240	54.779%	44.332%	-0.024%	0.076%
0	-5	3200	49.799%	40.302%	-0.022%	0.069%
0	0	2320	52.420%	42.423%	-0.023%	0.073%
0	5	2540	55.041%	44.544%	-0.024%	0.077%
0	10	-3400	57.662%	46.665%	-0.025%	0.080%

Dynamic Delta Hedging

We simulate the underlying asset price with assuming geometric brownian motion using monte-carlo simulation. During the simulation period, we dynamically replicate the option portfolio using the underlying asset and risk free aset. As we can see from the plot, our replicating portfolio have the relative same performance as the option movement.

```
S_0 = 100.0;
```

```
T = 1.0;
K = S_0 * 1.15;
r = 0.05;
mu = 0.1;
sigma = 0.2;
Nhedgepoints = 52;
Npaths = 500;
dt = T/Nhedgepoints;
S_t = zeros(Npaths,1);
Payoff = zeros(Npaths,1);
Hedge = zeros(Npaths,1);
Error = zeros(Npaths,1);
v = zeros(Nhedgepoints + 1, 1);
S = S_0;

for j = 1:Npaths

    %Delta Hedge
    S = S_0;
    V = BSMOption("Call",S,T,K,r,sigma,"Price");
    a = BSMOption("Call",S,T,K,r,sigma,"Delta");
    b = V-a*S;
    for i = 0:Nhedgepoints
        eps = normrnd(0,1);
        S = S*exp((mu-0.5*sigma^2)*dt+sigma*sqrt(dt)*eps);

        V = a*S+b*exp(r*dt);
        T_i = T-i*dt;
        a = BSMOption("Call",S,T_i,K,r,sigma,"Delta");
        b = V - a*S;
    end
    S_t(j) = S;
    Hedge(j) = V;
    Payoff(j) = BSMOption("Call",S,0,K,r,sigma,"Price");
end
figure(1);
scatter(S_t, Hedge);
hold on;
scatter(S_t, Payoff);
title("Hedge/Payoff");
legend('Hedged Portfolio Value','Option Value')
```


Hedge/Payoff
Hedged Portfolio Value
Option Value
90
80
70
60
50
40
30
20
10
0
-10
50
100
150
200