Portfolio Analysis

Table of Contents

Equity prices

Question 1

```
% When a dividend was paid ?
% Ex_dividend date : 12 Feb 2020

%What  is  the difference between adjusted and closing prices ?
% The closing price is the raw price, which is just the cash value of
 the last transacted price before the market closes.
% The adjusted closing price amends a stock's closing price to reflect
 that stock's value after accounting for any corporate actions.
```

Question 2

```
Date = (datetime(2012,01,01):calmonths(1):datetime(2013,12,01))';
Price = ([29.530 31.740 32.250 32.020 29.190 30.590 29.470 30.820
 29.780 28.530 26.620 26.730 27.420 27.800 28.610 33.100 34.880 34.530
 31.830 33.400 33.310 33.350 38.130 37.430])';

monthly_dollar_return = [0; diff(Price)];
monthly_simple_return = [0; tick2ret(Price)];
monthly_log_return = [0; diff(log(Price))];
annual_simple_return = monthly_simple_return*(365.25/23);
annual_log_return = monthly_log_return*(365.25/23);

StockPrice = [table(Date),table(Price), table(monthly_dollar_return),
 table(monthly_simple_return), table(monthly_log_return),
 table(annual_simple_return), table(annual_log_return)];

% On a monthly basis the returns are about the same while on a yearly
 basis the single returns are higher.
```

Stock Market Indices

1.Simple and Logarithmic returns

Index import

```
DJI = readtable('C:\Users\Nicolas\Documents\EFREI\Cours\M1\Cours
\Cryptocurrency Design and Risk\TD\Lab\data\^DJI.csv');
FTSE = readtable('C:\Users\Nicolas\Documents\EFREI\Cours\M1\Cours
\Cryptocurrency Design and Risk\TD\Lab\data\^FTSE.csv');
DAX = readtable('C:\Users\Nicolas\Documents\EFREI\Cours\M1\Cours
\Cryptocurrency Design and Risk\TD\Lab\data\^GDAXI.csv');
HSX = readtable('C:\Users\Nicolas\Documents\EFREI\Cours\M1\Cours
\Cryptocurrency Design and Risk\TD\Lab\data\^HSI.csv');
Nikkei = readtable('C:\Users\Nicolas\Documents\EFREI\Cours\M1\Cours
\Cryptocurrency Design and Risk\TD\Lab\data\^N225.csv');
SP500 = readtable('C:\Users\Nicolas\Documents\EFREI\Cours\M1\Cours
\Cryptocurrency Design and Risk\TD\Lab\data\^GSPC.csv');
```

```
Warning: Column headers from the file were modified to make them valid
 MATLAB
identifiers before creating variable names for the table. The original
 column
headers are saved in the VariableDescriptions property.
Set 'VariableNamingRule' to 'preserve' to use the original column
 headers as
table variable names.
Warning: Column headers from the file were modified to make them valid
 MATLAB
identifiers before creating variable names for the table. The original
 column
```

```
headers are saved in the VariableDescriptions property.
Set 'VariableNamingRule' to 'preserve' to use the original column
 headers as
table variable names.
Warning: Column headers from the file were modified to make them valid
 MATLAB
identifiers before creating variable names for the table. The original
 column
headers are saved in the VariableDescriptions property.
Set 'VariableNamingRule' to 'preserve' to use the original column
 headers as
table variable names.
Warning: Column headers from the file were modified to make them valid
 MATLAB
identifiers before creating variable names for the table. The original
 column
headers are saved in the VariableDescriptions property.
Set 'VariableNamingRule' to 'preserve' to use the original column
 headers as
table variable names.
Warning: Column headers from the file were modified to make them valid
 MATLAB
identifiers before creating variable names for the table. The original
 column
headers are saved in the VariableDescriptions property.
Set 'VariableNamingRule' to 'preserve' to use the original column
 headers as
table variable names.
Warning: Column headers from the file were modified to make them valid
 MATLAB
identifiers before creating variable names for the table. The original
 column
headers are saved in the VariableDescriptions property.
Set 'VariableNamingRule' to 'preserve' to use the original column
 headers as
table variable names.
```

Crypto import

```
BTC = readtable('C:\Users\Nicolas\Documents\EFREI\Cours
\M1\Cours\Cryptocurrency Design and Risk\TD\Lab\data
\hist_data_btceur_2020.csv');
```

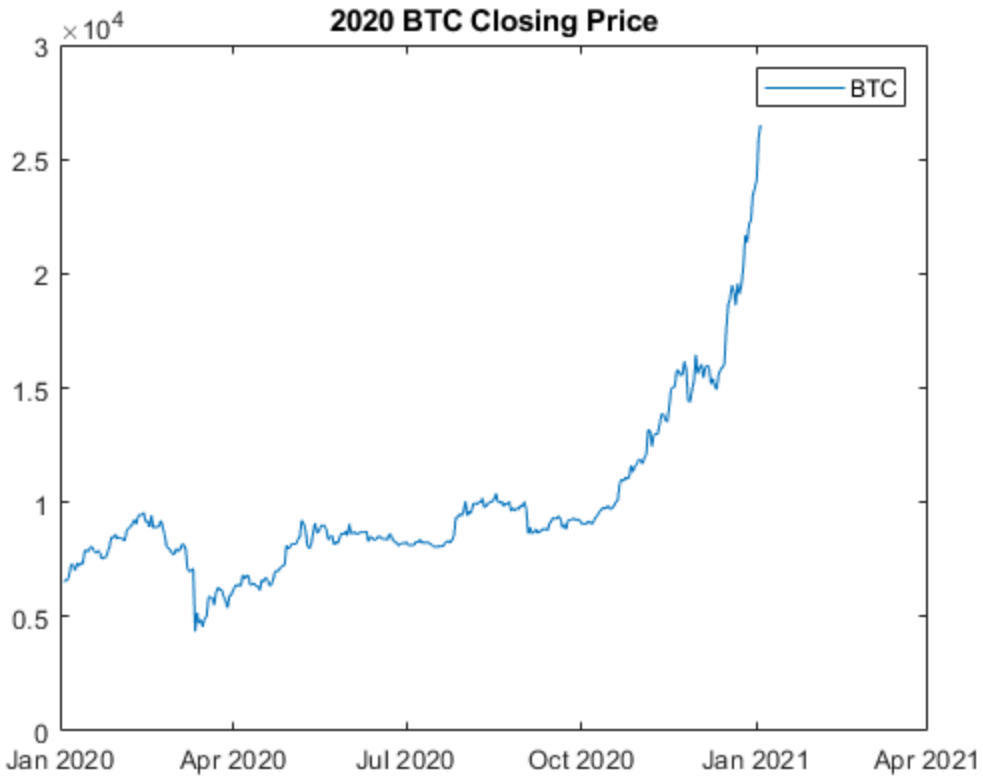
a) Stock Market Indices ploting

```
plot(DJI.Date, DJI.AdjClose);

hold on;

plot(DAX.Date, DAX.AdjClose);
plot(Nikkei.Date, Nikkei.AdjClose);
plot(HSX.Date, HSX.AdjClose);
```

```
plot(SP500.Date, SP500.AdjClose);
plot(FTSE.Date, FTSE.AdjClose);

hold off;

legend('DJIA', 'DAX', 'Nikkei', 'HSX', 'SP500', 'FTSE');
title('2007-2021 Stock Market Indices');
```

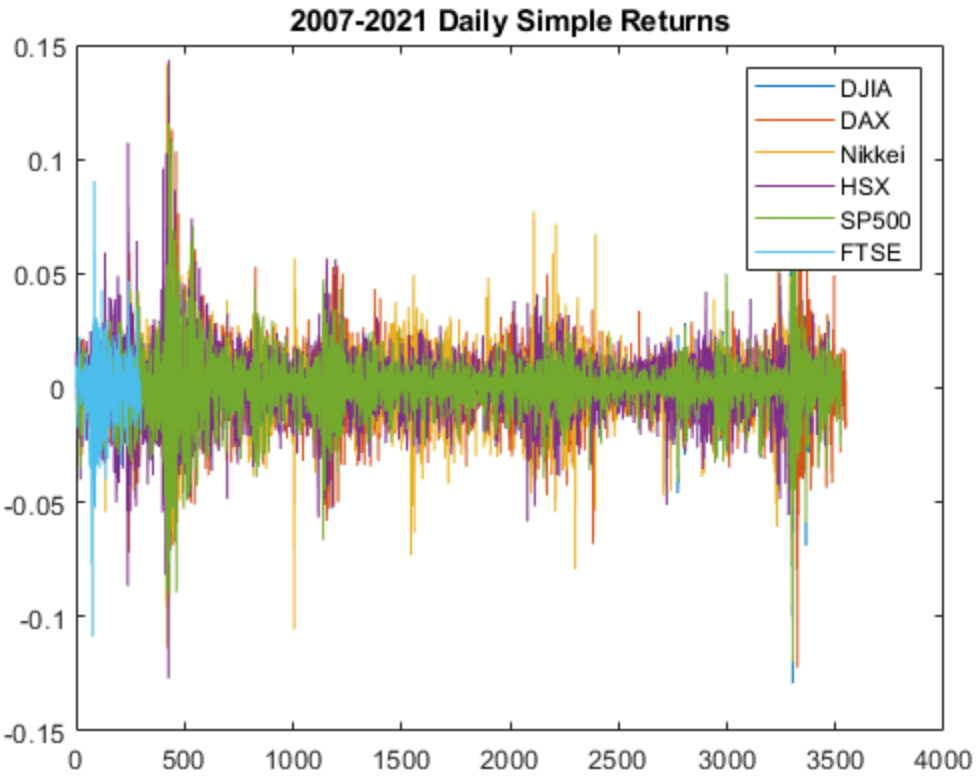
Crypto data ploting

```
plot(BTC.Timestamp, BTC.Close);
hold on
legend('BTC');
title('2020 BTC Closing Price');
hold off;
```



b) Daily simple returns indices ploting

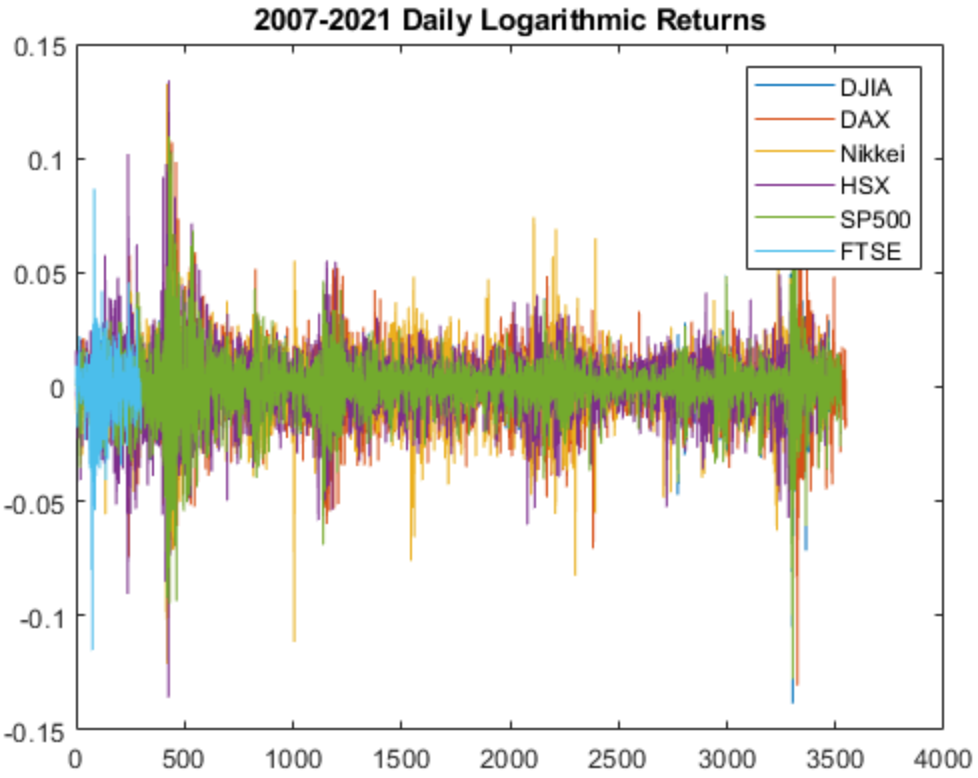
```
daily_simple_return_DJI = tick2ret(DJI.AdjClose);
daily_simple_return_DAX = tick2ret(DAX.AdjClose);
daily_simple_return_Nikkei = tick2ret(Nikkei.AdjClose);
daily_simple_return_HSX = tick2ret(HSX.AdjClose);
daily_simple_return_SP500 = tick2ret(SP500.AdjClose);
daily_simple_return_FTSE = tick2ret(FTSE.AdjClose);

plot(daily_simple_return_DJI);

hold on;

plot(daily_simple_return_DAX);
plot(daily_simple_return_Nikkei);
plot(daily_simple_return_HSX);
plot(daily_simple_return_SP500);
plot(daily_simple_return_FTSE);

hold off;

legend('DJIA', 'DAX', 'Nikkei', 'HSX', 'SP500', 'FTSE');
title('2007-2021 Daily Simple Returns');
```



Daily logarithmic returns indices ploting

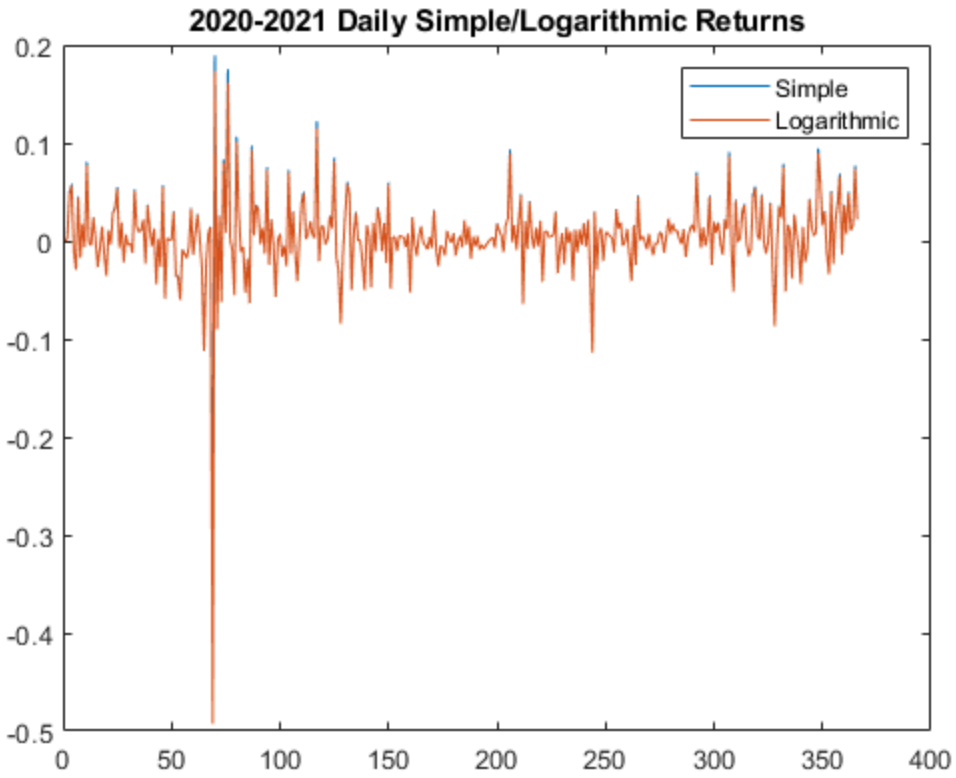
```
daily_log_return_DJI = diff(log(DJI.AdjClose));
daily_log_return_DAX = diff(log(DAX.AdjClose));
daily_log_return_Nikkei = diff(log(Nikkei.AdjClose));
daily_log_return_HSX = diff(log(HSX.AdjClose));
daily_log_return_SP500 = diff(log(SP500.AdjClose));
daily_log_return_FTSE = diff(log(FTSE.AdjClose));

plot(daily_log_return_DJI);

hold on;

plot(daily_log_return_DAX);
plot(daily_log_return_Nikkei);
plot(daily_log_return_HSX);
plot(daily_log_return_SP500);
plot(daily_log_return_FTSE);

hold off;

legend('DJIA', 'DAX', 'Nikkei', 'HSX', 'SP500', 'FTSE');
title('2007-2021 Daily Logarithmic Returns');
```

```
% It can be seen that the differences between the different indices
 are small.
% However, the different crises that have occurred in the past can be
 observed.
% Our dataset goes from 2007 to 2021, so first we have the subprime
 crisis of 2008 (about -10%) where yields were very volatile, the
 Greek crisis of 2011 and finally the crisis of COVID-19 (almost 15%).
```



Daily simple/logarithmic crypto returns ploting

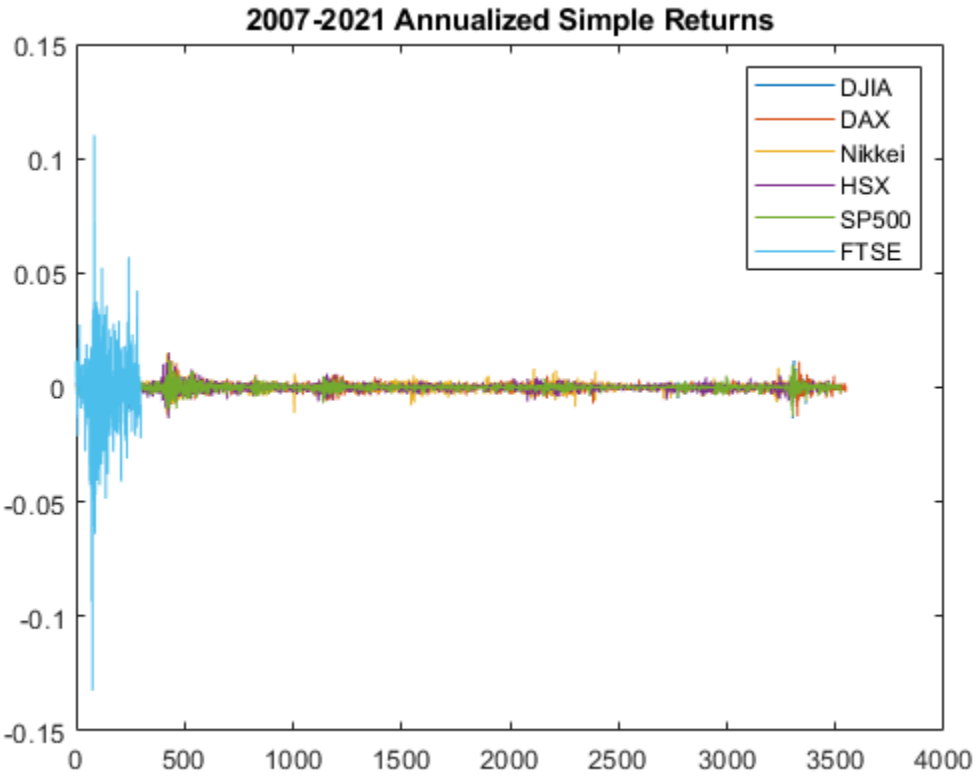
```
daily_simple_return_BTC = tick2ret(BTC.Close);
daily_log_return_BTC = diff(log(BTC.Close));

plot(daily_simple_return_BTC);

hold on;
plot(daily_log_return_BTC);

hold off;

legend('Simple', 'Logarithmic');
title('2020-2021 Daily Simple/Logarithmic Returns');
```



c) Annualized simple returns indices ploting

```
annualized_simple_return_DJI = tick2ret(DJI.AdjClose)*(365.25/
height(DJI));
annualized_simple_return_DAX = tick2ret(DAX.AdjClose)*(365.25/
height(DAX));
annualized_simple_return_Nikkei = tick2ret(Nikkei.AdjClose)*(365.25/
height(Nikkei));
annualized_simple_return_HSX = tick2ret(HSX.AdjClose)*(365.25/
height(HSX));
annualized_simple_return_SP500 = tick2ret(SP500.AdjClose)*(365.25/
height(SP500));
annualized_simple_return_FTSE = tick2ret(FTSE.AdjClose)*(365.25/
height(FTSE));

plot(annualized_simple_return_DJI);

hold on;

plot(annualized_simple_return_DAX);
plot(annualized_simple_return_Nikkei);
plot(annualized_simple_return_HSX);
plot(annualized_simple_return_SP500);
plot(annualized_simple_return_FTSE);
```

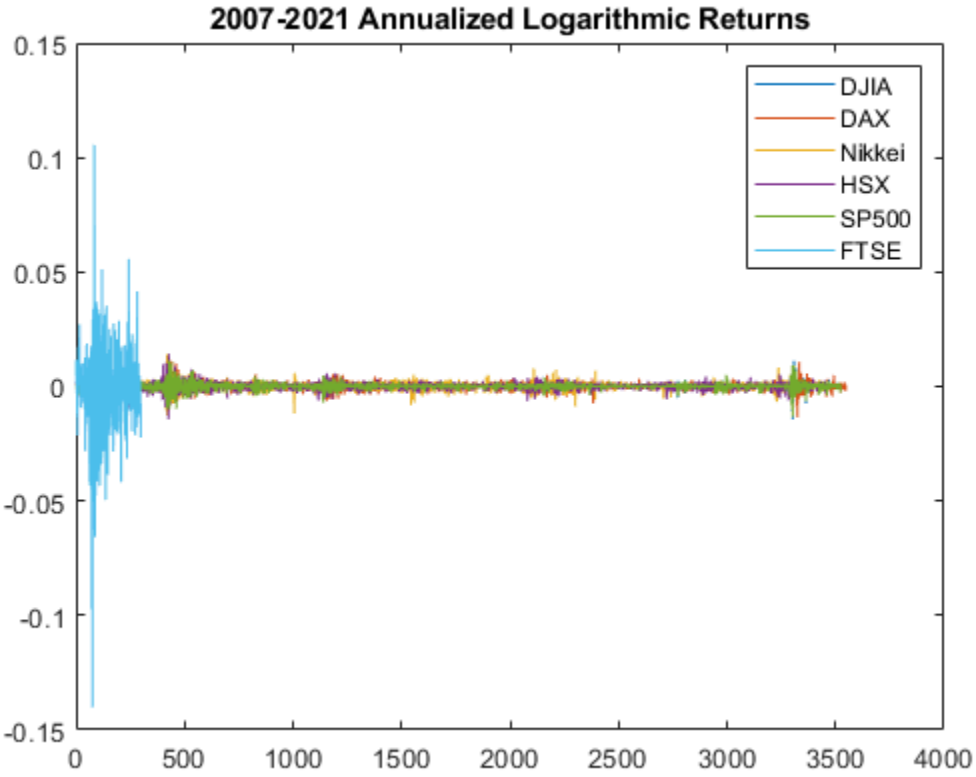
```
hold off;

legend('DJIA', 'DAX', 'Nikkei', 'HSX', 'SP500', 'FTSE');
title('2007-2021 Annualized Simple Returns');
```



c) Annualized Logarithmic returns indices plotting

```
annualized_log_return_DJI = diff(log(DJI.AdjClose))*(365.25/
height(DJI));
annualized_log_return_DAX = diff(log(DAX.AdjClose))*(365.25/
height(DAX));
annualized_log_return_Nikkei = diff(log(Nikkei.AdjClose))*(365.25/
height(Nikkei));
annualized_log_return_HSX = diff(log(HSX.AdjClose))*(365.25/
height(HSX));
annualized_log_return_SP500 = diff(log(SP500.AdjClose))*(365.25/
height(SP500));
annualized_log_return_FTSE = diff(log(FTSE.AdjClose))*(365.25/
height(FTSE));

plot(annualized_log_return_DJI);

hold on;
```

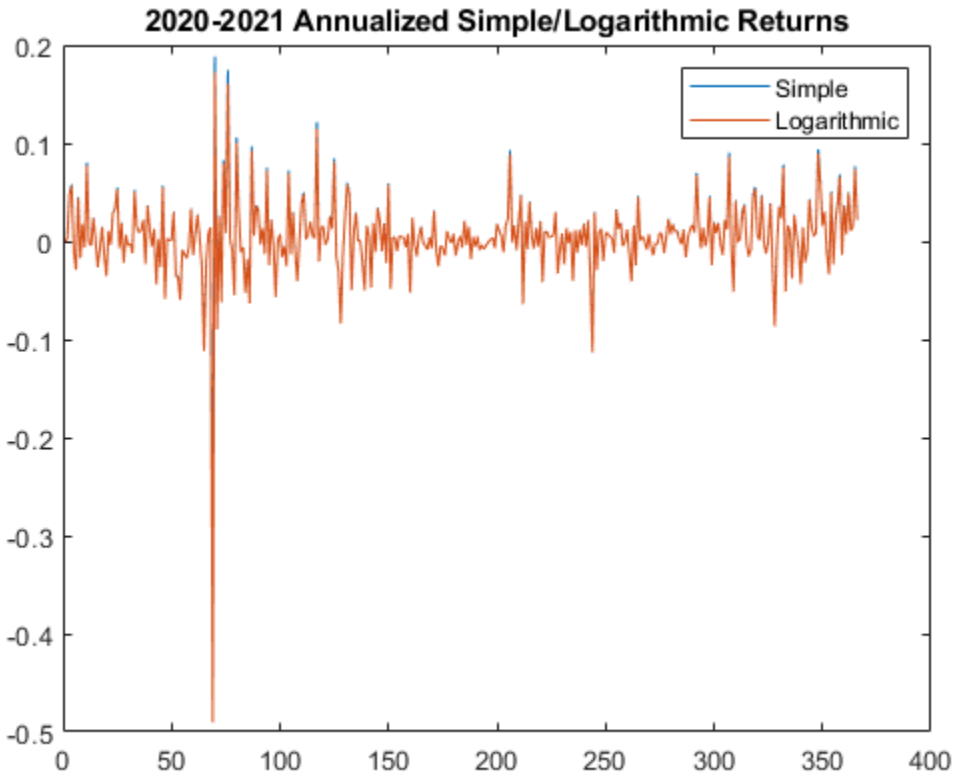
```
plot(annualized_log_return_DAX);
plot(annualized_log_return_Nikkei);
plot(annualized_log_return_HSX);
plot(annualized_log_return_SP500);
plot(annualized_log_return_FTSE);

hold off;

legend('DJIA', 'DAX', 'Nikkei', 'HSX', 'SP500', 'FTSE');
title('2007-2021 Annualized Logarithmic Returns');

% It can be seen that between the simple and logarithmic annual
 returns there are no major differences.
% However, the volatility is much lower than for simple and
 logarithmic daily returns.
```



c) Annualized simple/logarithmic crypto returns ploting

```
annualized_simple_return_BTC = tick2ret(BTC.Close)*(365.25/
height(BTC));
annualized_log_return_BTC = diff(log(BTC.Close))*(365.25/height(BTC));

plot(annualized_simple_return_BTC);
```

```
hold on;
plot(annualized_log_return_BTC);

hold off;

legend('Simple', 'Logarithmic');
title('2020-2021 Annualized Simple/Logarithmic Returns');
```



d) Simple returns of holding each of the indices over the entire period

```
sum(daily_simple_return_DJI);
sum(daily_simple_return_DAX);
sum(daily_simple_return_Nikkei);
sum(daily_simple_return_HSX);
sum(daily_simple_return_SP500);
sum(daily_simple_return_FTSE);
```

d) Logarithmic returns of holding each of the indices over the entire period

```
sum(daily_log_return_DJI);
sum(daily_log_return_DAX);
```

```
sum(daily_log_return_Nikkei);
sum(daily_log_return_HSX);
sum(daily_log_return_SP500);
sum(daily_log_return_FTSE);
```

2.Statistic proprieties of returns

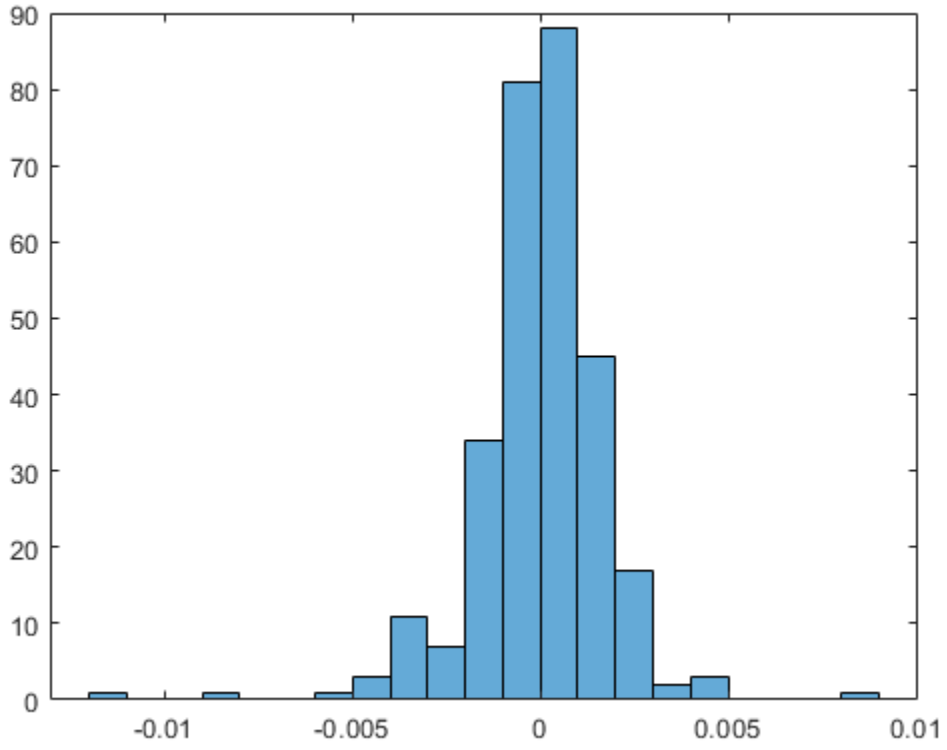
a) Table of statistics for the monthly data

```
DJI_monthly_simple_return = tick2ret(DJI.AdjClose)*((365.25/12)/
height(DJI));
DAX_monthly_simple_return = tick2ret(DAX.AdjClose)*((365.25/12)/
height(DAX));
Nikkei_monthly_simple_return = tick2ret(Nikkei.AdjClose)*((365.25/12)/
height(Nikkei));
HSX_monthly_simple_return = tick2ret(HSX.AdjClose)*((365.25/12)/
height(HSX));
SP500_monthly_simple_return = tick2ret(SP500.AdjClose)*((365.25/12)/
height(SP500));
FTSE_monthly_simple_return = tick2ret(FTSE.AdjClose)*((365.25/12)/
height(FTSE));

DJI_monthly_log_return = diff(log(DJI.AdjClose))*((365.25/12)/
height(DJI));
DAX_monthly_log_return = diff(log(DAX.AdjClose))*((365.25/12)/
height(DAX));
Nikkei_monthly_log_return = diff(log(Nikkei.AdjClose))*((365.25/12)/
height(Nikkei));
HSX_monthly_log_return = diff(log(HSX.AdjClose))*((365.25/12)/
height(HSX));
SP500_monthly_log_return = diff(log(SP500.AdjClose))*((365.25/12)/
height(SP500));
FTSE_monthly_log_return = diff(log(FTSE.AdjClose))*((365.25/12)/
height(FTSE));

DJIMin = min(DJI_monthly_simple_return);
DJIMax = max(DJI_monthly_simple_return);
DJIMean = mean(DJI_monthly_simple_return);
DJIMedian= median(DJI_monthly_simple_return);
DJIStd = std(DJI_monthly_simple_return);
DJIVar = var(DJI_monthly_simple_return);
DJISkew = skewness(DJI_monthly_simple_return);
DJIKurtosis = kurtosis(DJI_monthly_simple_return);
DJIStats = array2table([DJIMin DJIMax DJIMean DJIMedian DJIStd DJIVar
 DJISkew DJIKurtosis]);
C =
 {'DJIMin' 'DJIMax' 'DJIMean' 'DJIMedian' 'DJIStd' 'DJIVar' 'DJISkew' 'DJIKurtosis
DJIStats.Properties.VariableNames = C;

% Kurtosis is very high. The min, max, mean, median, skewness,
 standard deviation, variance is very low.
```

3.Distribution of returns

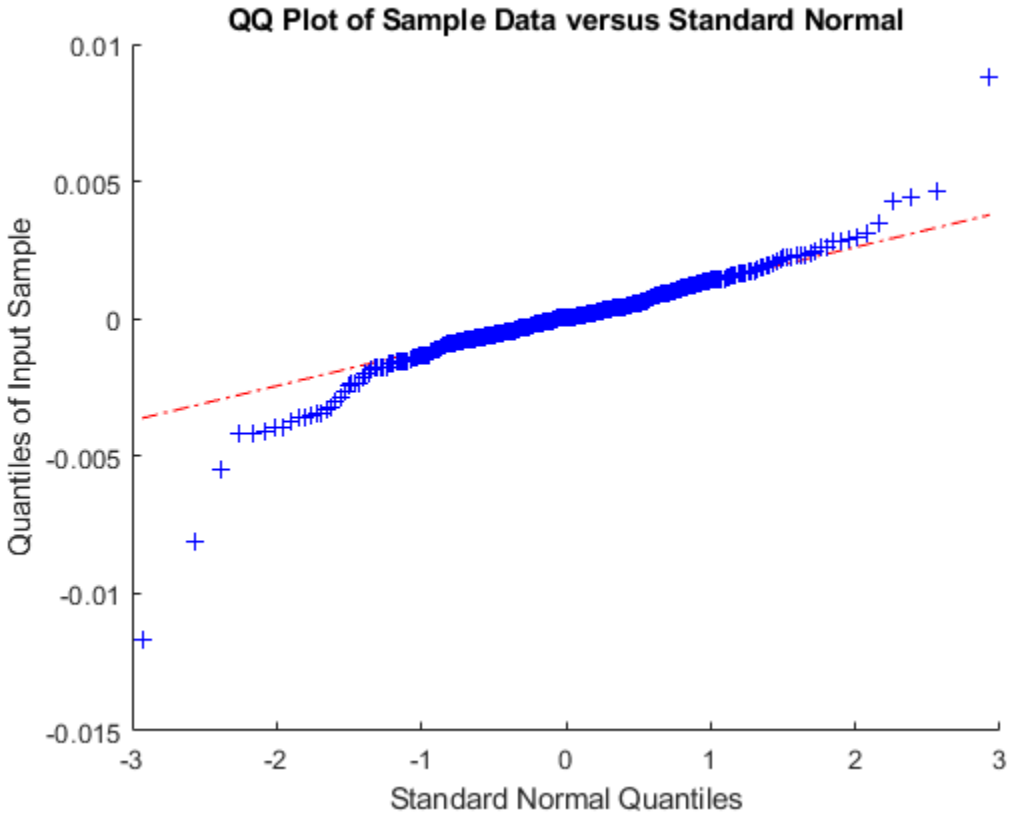
b) Plot the histogram of monthly log prices and the kernel density estimate for all the assets

```
histogram(DJI_monthly_log_return);
[f,xi] = ksdensity(DJI_monthly_log_return);

histogram(DAX_monthly_log_return);
[f,xi] = ksdensity(DAX_monthly_log_return);

histogram(Nikkei_monthly_log_return);
[f,xi] = ksdensity(Nikkei_monthly_log_return);

histogram(HSX_monthly_log_return);
[f,xi] = ksdensity(HSX_monthly_log_return);

histogram(SP500_monthly_simple_return);
[f,xi] = ksdensity(SP500_monthly_simple_return);

histogram(FTSE_monthly_log_return);
[f,xi] = ksdensity(FTSE_monthly_log_return);

% The histograms confirm the comments made to the previous question.
```



c) Draw the Q-Q plot for the monthly data and simulated normal data

```
qqplot(DJI_monthly_log_return);
qqplot(DAX_monthly_log_return);
qqplot(Nikkei_monthly_log_return);
qqplot(HSX_monthly_log_return);
qqplot(SP500_monthly_simple_return);
qqplot(FTSE_monthly_log_return);

% The blue curve is aligned with the red curve so that the samples
 appear normally distributed.
```



4.Statistic tests of normality

d) Calculate the test statistic

```
[T,p] = jbtest(DJI_monthly_log_return, 0.05);
[H,p] = swtest(DJI_monthly_log_return);
[h,p] = kstest(DJI_monthly_log_return);

[T,p] = jbtest(DAX_monthly_log_return, 0.05);
[H,p] = swtest(DAX_monthly_log_return);
```

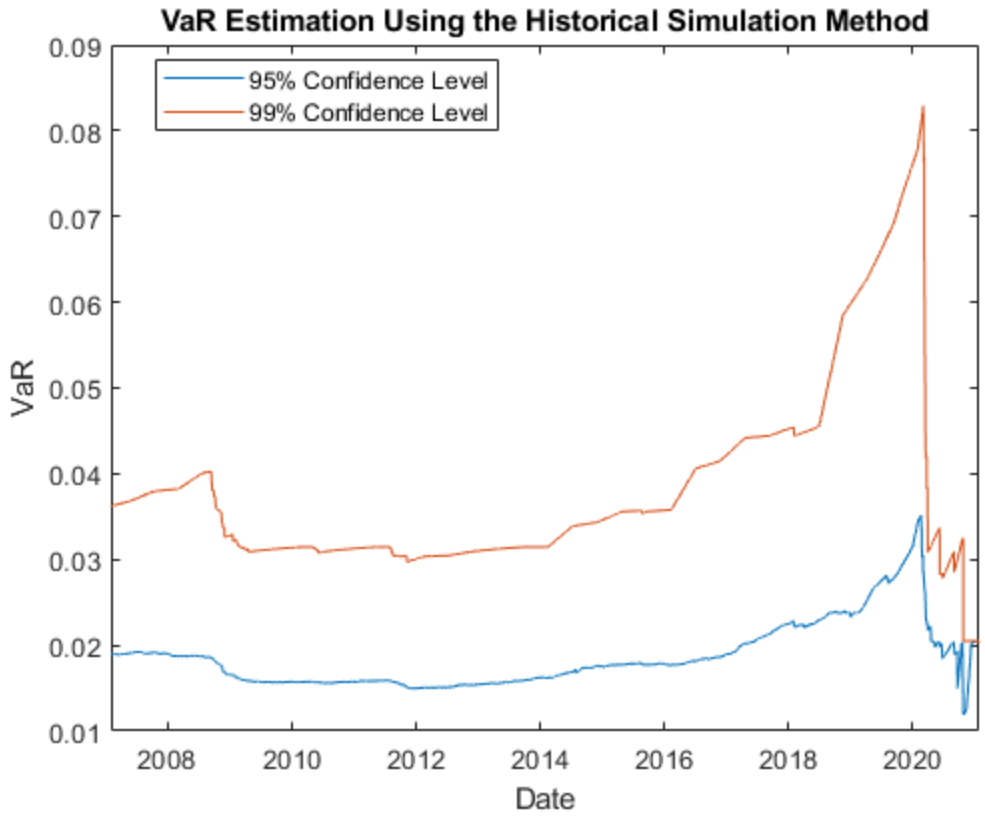
```
[h,p] = kstest(DAX_monthly_log_return);

[T,p] = jbtest(Nikkei_monthly_log_return, 0.05);
[H,p] = swtest(Nikkei_monthly_log_return);
[h,p] = kstest(Nikkei_monthly_log_return);

[T,p] = jbtest(HSX_monthly_log_return, 0.05)
[H,p] = swtest(HSX_monthly_log_return);
[h,p] = kstest(HSX_monthly_log_return);

[T,p] = jbtest(SP500_monthly_simple_return, 0.05);
[H,p] = swtest(SP500_monthly_simple_return);
[h,p] = kstest(SP500_monthly_simple_return);

[T,p] = jbtest(FTSE_monthly_log_return, 0.05);
[H,p] = swtest(FTSE_monthly_log_return);
[h,p] = kstest(FTSE_monthly_log_return);

% The value of the tests is 1, i.e. they reject the null hypothesis
 H0 (Gaussian distribution) and that the samples are not normally
 distributed.

Warning: P is less than the smallest tabulated value, returning
 0.001.
Warning: P is less than the smallest tabulated value, returning
 0.001.
Warning: P is less than the smallest tabulated value, returning
 0.001.
Warning: P is less than the smallest tabulated value, returning
 0.001.

T =

     1


p =

   1.0000e-03

Warning: P is less than the smallest tabulated value, returning
 0.001.
Warning: P is less than the smallest tabulated value, returning
 0.001.
```

5. Calculate VaR

a) Historical Simulation Method

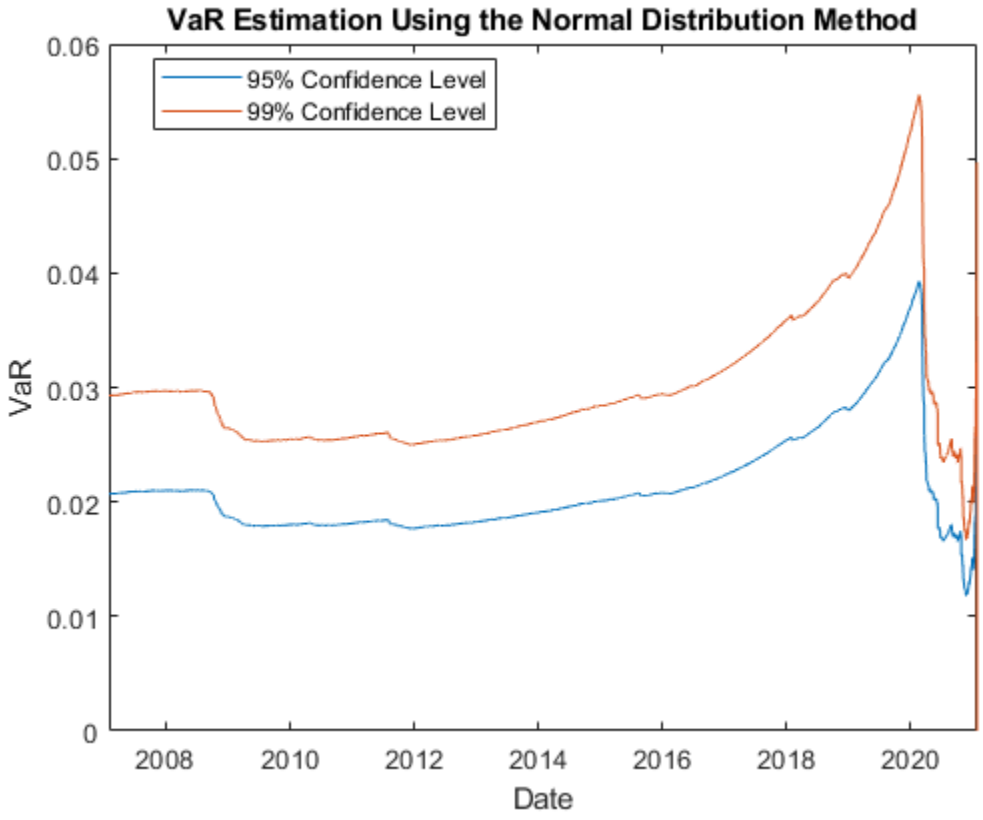
```
Returns = daily_simple_return_DJI;
Date = DJI.Date;
TestWindowEnd = length(Returns);
```

```
TestWindowStart = find(year(Date)==2007,1);
TestWindow = TestWindowStart : TestWindowEnd;
EstimationWindowSize = TestWindowEnd;

pVaR = [0.05 0.01];

Historical95 = zeros(length(TestWindow),1);
Historical99 = zeros(length(TestWindow),1);

for t = TestWindow
    i = t - TestWindowStart + 1;
    EstimationWindow = t:EstimationWindowSize;
    X = Returns(EstimationWindow);
    Historical95(i) = -quantile(X,pVaR(1));
    Historical99(i) = -quantile(X,pVaR(2));
end

figure;
plot(DateReturns(TestWindow),[Historical95 Historical99]);
ylabel('VaR');
xlabel('Date');
legend({'95% Confidence Level','99% Confidence
 Level'},'Location','Best');
title('VaR Estimation Using the Historical Simulation Method');
```



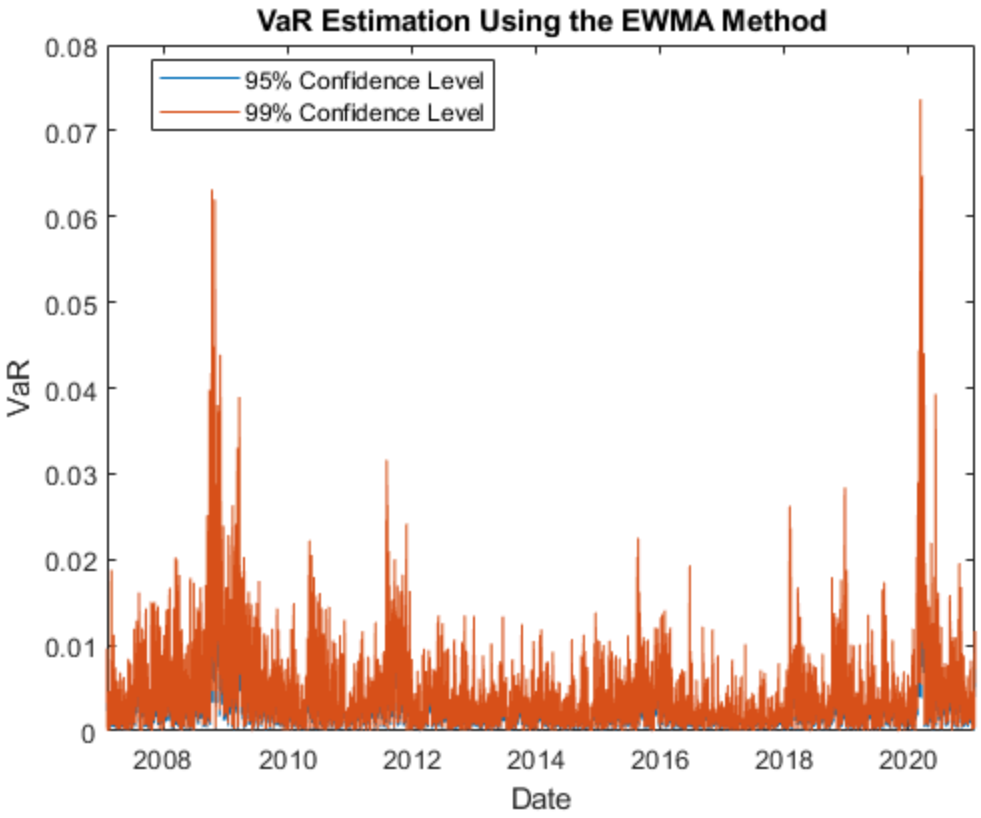
b) Normal Distribution Method

```
Zscore = norminv(pVaR);
Normal95 = zeros(length(TestWindow),1);
Normal99 = zeros(length(TestWindow),1);

for t = TestWindow
i = t - TestWindowStart + 1;
EstimationWindow = t:EstimationWindowSize;
Sigma = std(Returns(EstimationWindow));
Normal95(i) = -Zscore(1)*Sigma;
Normal99(i) = -Zscore(2)*Sigma;
end

figure;
plot(Date(TestWindow),[Normal95 Normal99]);
xlabel('Date');
ylabel('VaR');
legend({'95% Confidence Level','99% Confidence
 Level'},'Location','Best');
title('VaR Estimation Using the Normal Distribution Method');
```



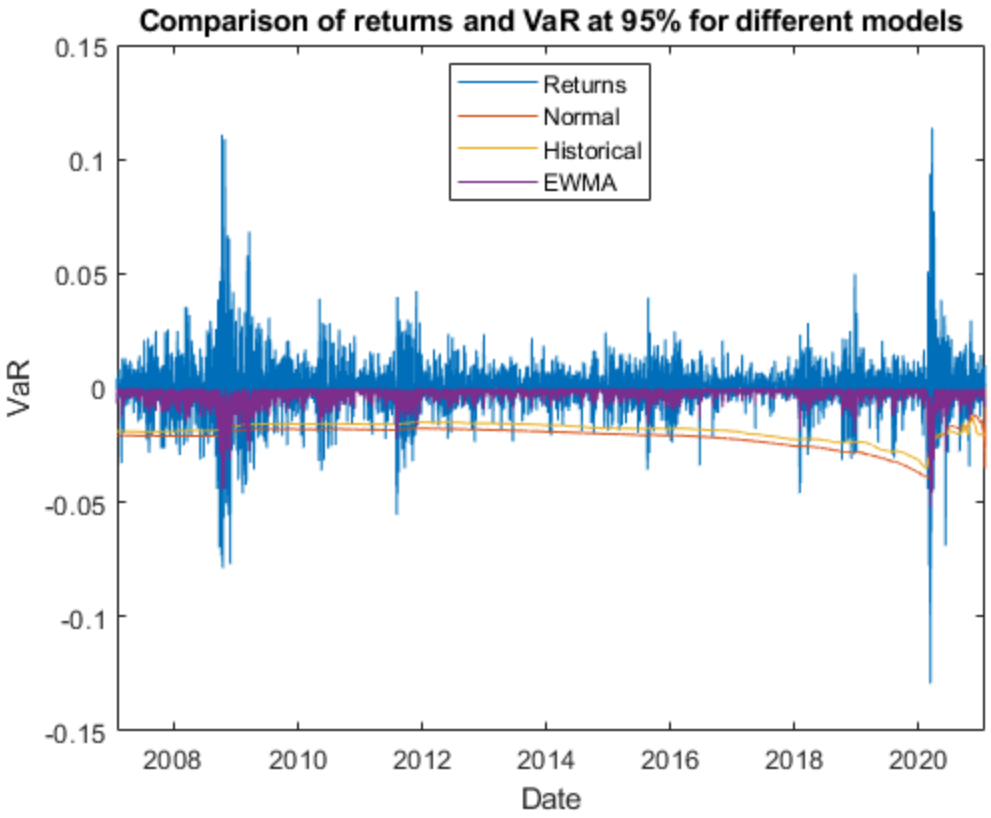
c) Exponential Weighted Moving Average Method (EWMA)

```
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);
EWMA99 = zeros(length(TestWindow),1);

for t = TestWindow
    k  = t - TestWindowStart + 1;
    Sigma2(t) = (1-Lambda) * Returns(t)^2 + Lambda * Sigma2(t);
    Sigma = sqrt(Sigma2(t));
    EWMA95(k) = -Zscore(1)*Sigma;
    EWMA99(k) = -Zscore(2)*Sigma;
end

figure;
plot(Date(TestWindow),[EWMA95 EWMA99]);
ylabel('VaR');
xlabel('Date');
legend({'95% Confidence Level','99% Confidence
 Level'},'Location','Best');
title('VaR Estimation Using the EWMA Method');
```



d) VaR Backtesting

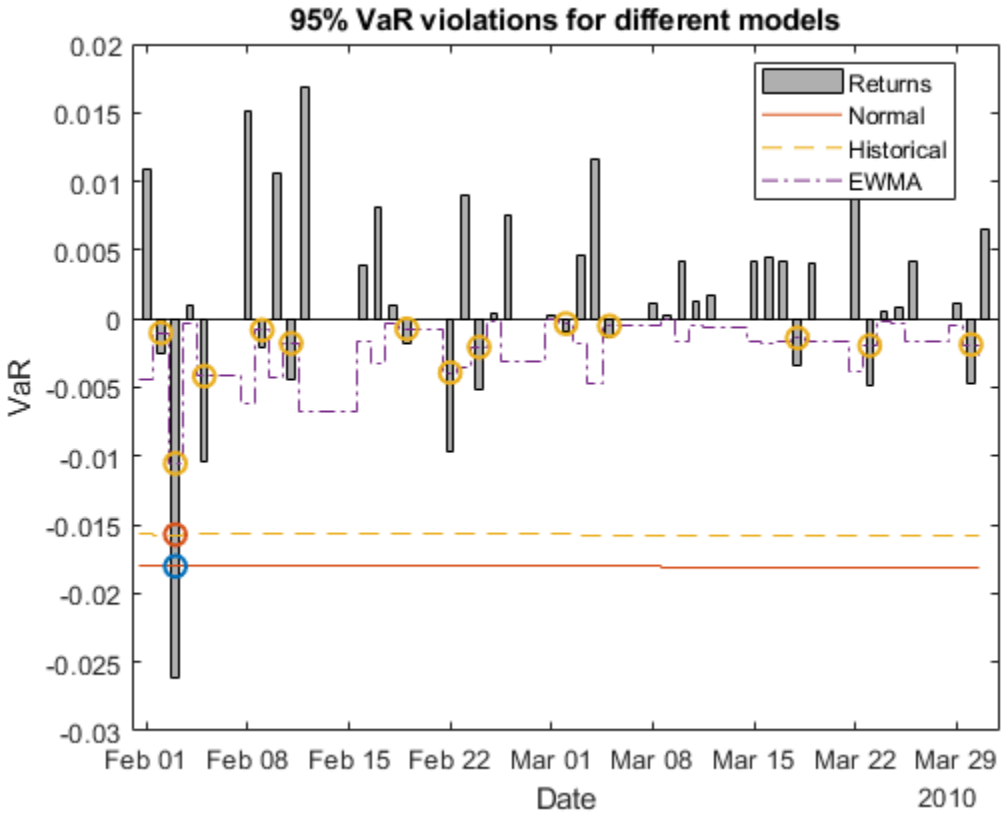
```
ReturnsTest = Returns(TestWindow);
DatesTest = Date(TestWindow);
figure;
plot(DatesTest,[ReturnsTest -Normal95 -Historical95 -EWMA95]);
ylabel('VaR');
xlabel('Date');
legend({'Returns','Normal','Historical','EWMA'},'Location','Best');
title('Comparison of returns and VaR at 95% for different models');
```



e) VaR violation

```
ZoomInd   = (DatesTest >= datetime(2010,01,31)) & (DatesTest <=
 datetime(2010,03,31));
VaRData   = [-Normal95(ZoomInd) -Historical95(ZoomInd) -
EWMA95(ZoomInd)];
VaRFormat = {'-','--','-.'};
D = DatesTest(ZoomInd);
R = ReturnsTest(ZoomInd);
N = Normal95(ZoomInd);
H = Historical95(ZoomInd);
E = EWMA95(ZoomInd);
IndN95    = (R < -N);
IndHS95   = (R < -H);
IndEWMA95 = (R < -E);
figure;
bar(D,R,0.5,'FaceColor',[0.7 0.7 0.7]);
hold on
for i = 1 : size(VaRData,2)
    stairs(D-0.5,VaRData(:,i),VaRFormat{i});
end
ylabel('VaR');
xlabel('Date');
legend({'Returns','Normal','Historical','EWMA'},'Location','Best','AutoUpdate','Of
title('95% VaR violations for different models');
```

```
ax = gca;
ax.ColorOrderIndex = 1;

plot(D(IndN95),-N(IndN95),'o',D(IndHS95),-H(IndHS95),'o',
 D(IndEWMA95),-E(IndEWMA95),'o','MarkerSize',8,'LineWidth',1.5);
xlim([D(1)-1, D(end)+1]);
hold off;
```



Published with MATLAB® R2020b