Algorithmic Trading Project

High Frequency Momentum Trading Stategy with Cryptocurrencies

Weihang Ren, April 20th 2020

Table of Contents

Data

Obtained from CryptoCompare via their official Python API, the data consists of the **hourly** prices of 7 cryptocurrencies versus the US Dollar over three chosen stating dates until 14:00 on April 30th 2020:

- `2019-10-30 00:00` (6 months)
- `2019-04-30 00:00` (1 year)
- `2018-04-30 00:00` (2 years)

Select different time range `'TR'` to see visualization and results accordingly.

Import data

```
% initialize
clear
% select time range: 6month, 1year or 2year
TR = "6m";

% config plots x limits
switch TR
    case "6m"
        xrange = [datetime(2019,10,29) datetime(2020,5,1)];
    case "1y"
        xrange = [datetime(2019,4,25) datetime(2020,5,1)];
    case "2y"
        xrange = [datetime(2018,4,25) datetime(2020,5,1)];
    otherwise
        warning('Unexpected plot range. No time selected.')
end

opts = delimitedTextImportOptions("NumVariables", 8);

% Specify range and delimiter
opts.DataLines = [2, Inf];
opts.Delimiter = ",";

% Specify column names and types
opts.VariableNames = ["time", "Bitcoin", "Ethereum", "Ripple", "BitcoinCash", "Tether",
opts.VariableTypes = ["datetime", "double", "double", "double", "double", "double", "do

% Specify file level properties
opts.ExtraColumnsRule = "ignore";
opts.EmptyLineRule = "read";

% Specify variable properties
opts = setvaropts(opts, "time", "InputFormat", "yyyy-MM-dd HH:mm:ss");

% Import the data
crypto = readtimetable("data/crypto_"+TR+".csv", opts);

clear opts
```

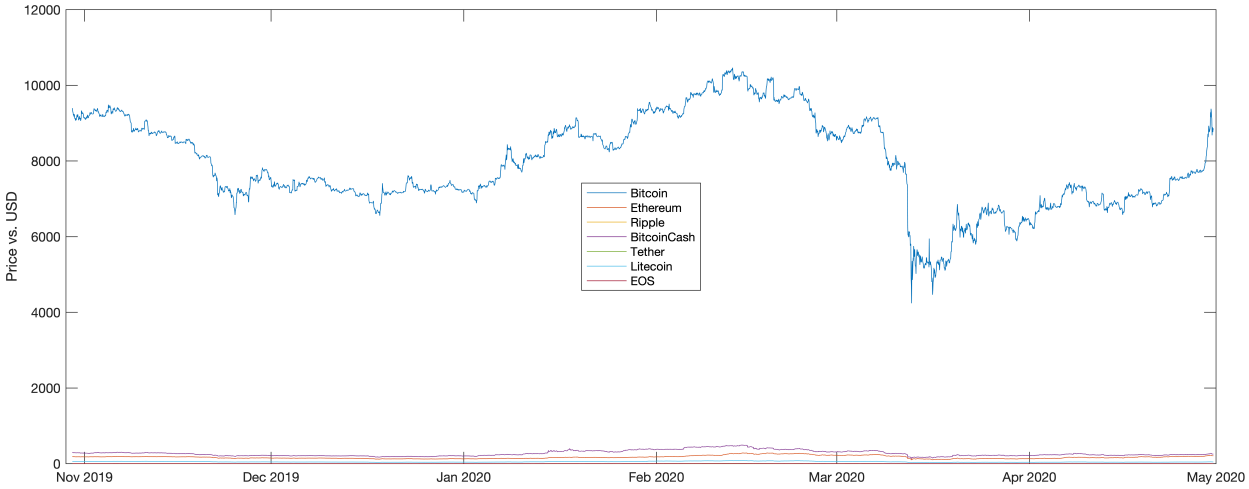
(TODO) Statistics summary

```
% head(crypto)
% corrplot(crypto);
```

Plot price

```
f=figure;
f.Position(3)= 2*f.Position(3);

plot(crypto.time,crypto.Bitcoin,'DisplayName','crypto.Bitcoin');hold on;plot(crypto.tim

legend({'Bitcoin','Ethereum','Ripple','BitcoinCash','Tether','Litecoin','EOS'},'Locatio
ylabel('Price vs. USD')
```

```
xlim(xrange)
```



Index price

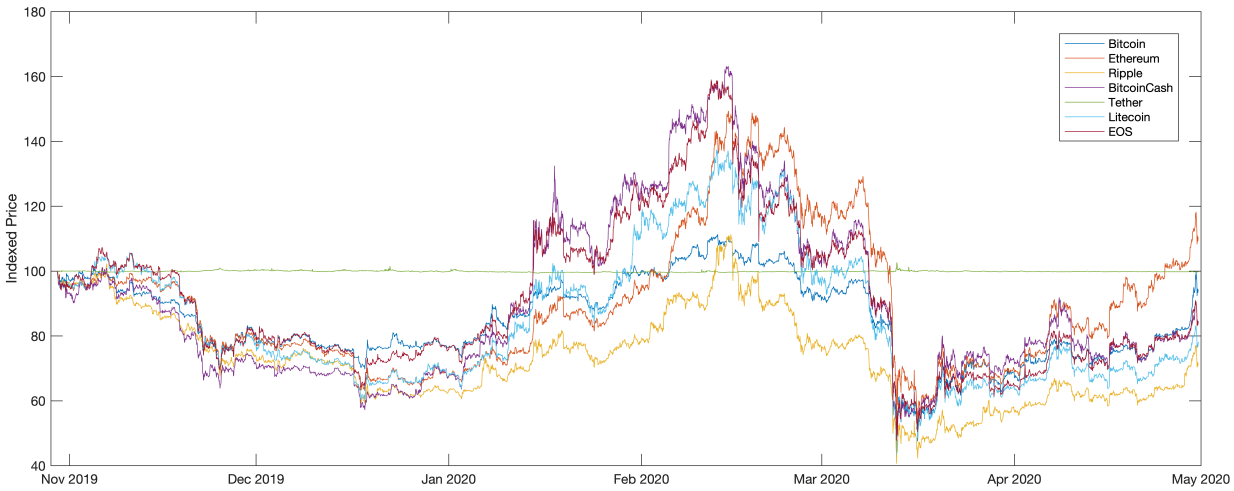
We use indexed price to better capture the volatility across cryptocurrencies and use it to build our signal.
Calculation reference: this page

$$\hat{X}_t = \frac{X_t}{X_0}$$

```
indexedPrice = crypto;
indexedPrice{2:end,:} = indexedPrice{2:end,:}./indexedPrice{1,:}*100;
indexedPrice{1,:} = 100;
```

Plot indexed price

```
plot(indexedPrice.time,indexedPrice.Bitcoin,'DisplayName','indexedPrice.Bitcoin');hold

legend({'Bitcoin','Ethereum','Ripple','BitcoinCash','Tether','Litecoin','EOS'},'Locatio
ylabel('Indexed Price')
xlim(xrange)
```



Tether is a stable coin.

Method

Exponential moving average

Ref: Chu, 2020.

$$EMA_t(P,\alpha) = \alpha \cdot P_t + (1-\alpha)EMA_{t-1}(P,\alpha) \;\; (t>0),\; EMA_0 = P_0 = 1,$$

where P_t is the indexed exchange rate of a cryptocurrency at time t, and $\alpha = \frac{1}{n_k}$ is the exponential smoothing ratio.

Span/windowsize, alpha, half-life conversion see pandas.ewm doc.

```
% arbitrarily, n_k for short and long emas are [n_s, n_l] (days)
n = [8,24;16,48;32,96]
```

```
n = 3×2
     8    24
    16    48
    32    96
```

```
% n = 2.*n-1; % span
```

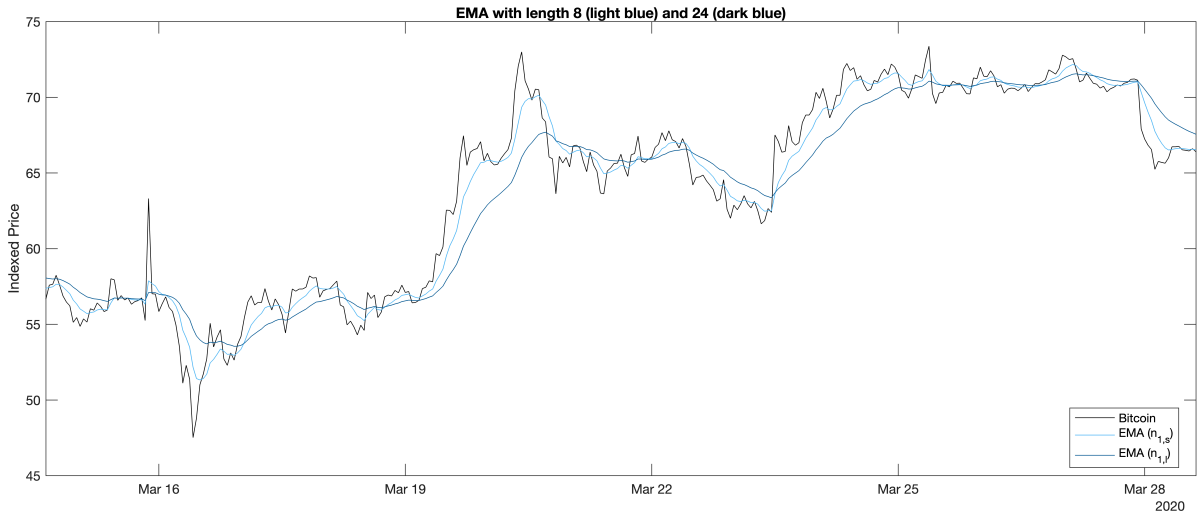
It can often be seen that the exchange rates of cryptocurrencies show short bursts of upwards or downwards movements over the course of a few hours, and an EMA can capture this shorter term influence.

```
ema = indexedPrice;
type = 'exponential';
for i = 1:3
    for j = 1:2
        windowSize = round(n(i,j));
        ema(:,end+1:end+7) = movavg(ema(:,1:7),type,windowSize);
    end
```

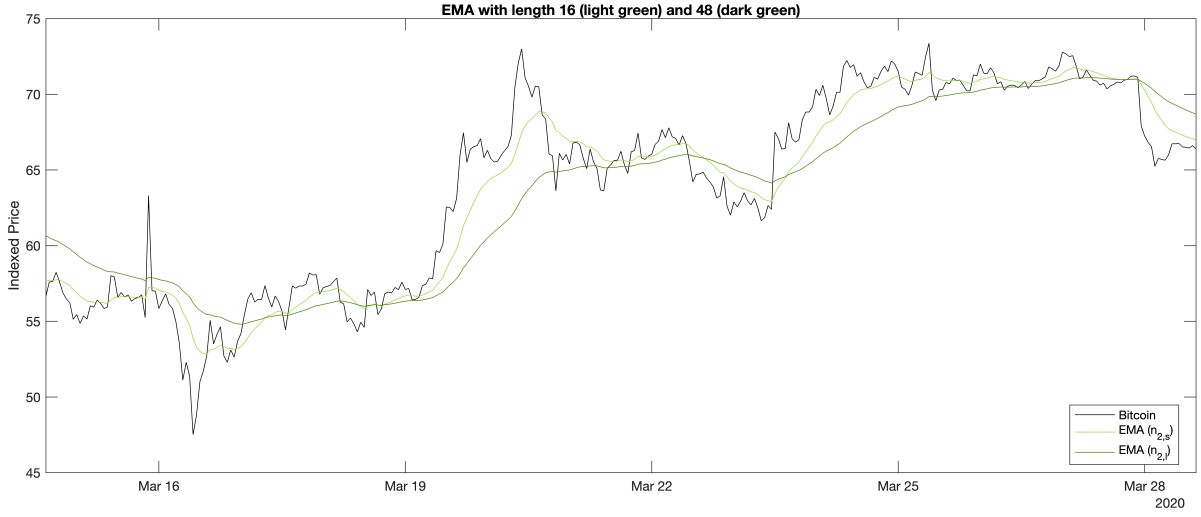
```
end

% head(ema)

bitcoinEma = ema(:,[1 8 15 22 29 36 43]);
bitcoinEma.Properties.VariableNames = {'Bitcoin' '8' '24' '16' '48' '32' '96'};
ethereumEma = ema(:,[2 9 16 23 30 37 44]);
ethereumEma.Properties.VariableNames = {'Ethereum' '8' '24' '16' '48' '32' '96'};
rippleEma = ema(:,[3 10 17 24 31 38 45]);
rippleEma.Properties.VariableNames = {'Ripple' '8' '24' '16' '48' '32' '96'};
bitcoincashEma = ema(:,[4 11 18 25 32 39 46]);
bitcoincashEma.Properties.VariableNames = {'BitcoinCash' '8' '24' '16' '48' '32' '96'};
tetherEma = ema(:,[5    12    19    26    33    40    47]);
tetherEma.Properties.VariableNames = {'Tether' '8' '24' '16' '48' '32' '96'};
litecoinEma = ema(:,[6    13    20    27    34    41    48]);
litecoinEma.Properties.VariableNames = {'Litecoin' '8' '24' '16' '48' '32' '96'};
eosEma = ema(:,[7    14    21    28    35    42    49]);
eosEma.Properties.VariableNames = {'EOS' '8' '24' '16' '48' '32' '96'};
```

Visualization

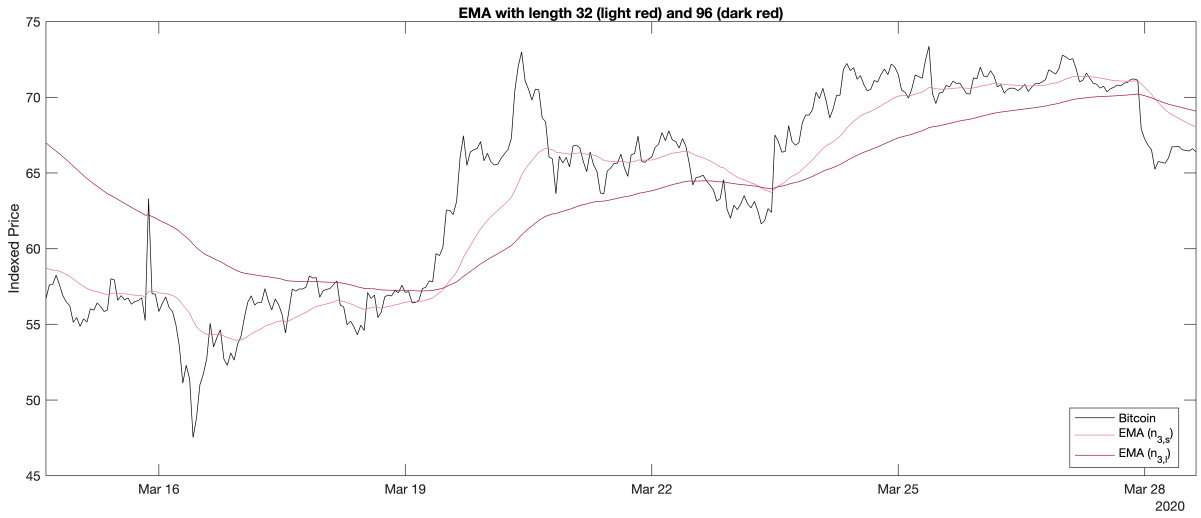
```
% change the symbol to see other crypto
symbol = bitcoinEma;
% change the datetime span to adjust plots horizon
span = [datetime(2020,3,14,15,0,0) datetime(2020,3,28,15,0,0)];


name = symbol.Properties.VariableNames(1);
name = string(name);

% plot n1
plot(symbol.time,symbol.(name),'k','DisplayName','bitcoinEma.Bitcoin');hold on;plot(sym

xlim(span)
legend({name,'EMA (n_{1,s})','EMA (n_{1,l})'},'Location','best')
title('EMA with length 8 (light blue) and 24 (dark blue)')
ylabel('Indexed Price')
```




```
% plot n2
plot(symbol.time,symbol.(name),'k','DisplayName','bitcoinEma.Bitcoin');hold on;plot(sym

xlim(span)
legend({name,'EMA (n_{2,s})','EMA (n_{2,l})'},'Location','best')
title('EMA with length 16 (light green) and 48 (dark green)')
ylabel('Indexed Price')
```




```
% plot n3
plot(symbol.time,symbol.(name),'k','DisplayName','bitcoinEma.Bitcoin');hold on;plot(sym

xlim(span)
legend({name,'EMA (n_{3,s})','EMA (n_{3,l})'},'Location','best')
title('EMA with length 32 (light red) and 96 (dark red)')
ylabel('Indexed Price')
```



Base signal for momentum strategy

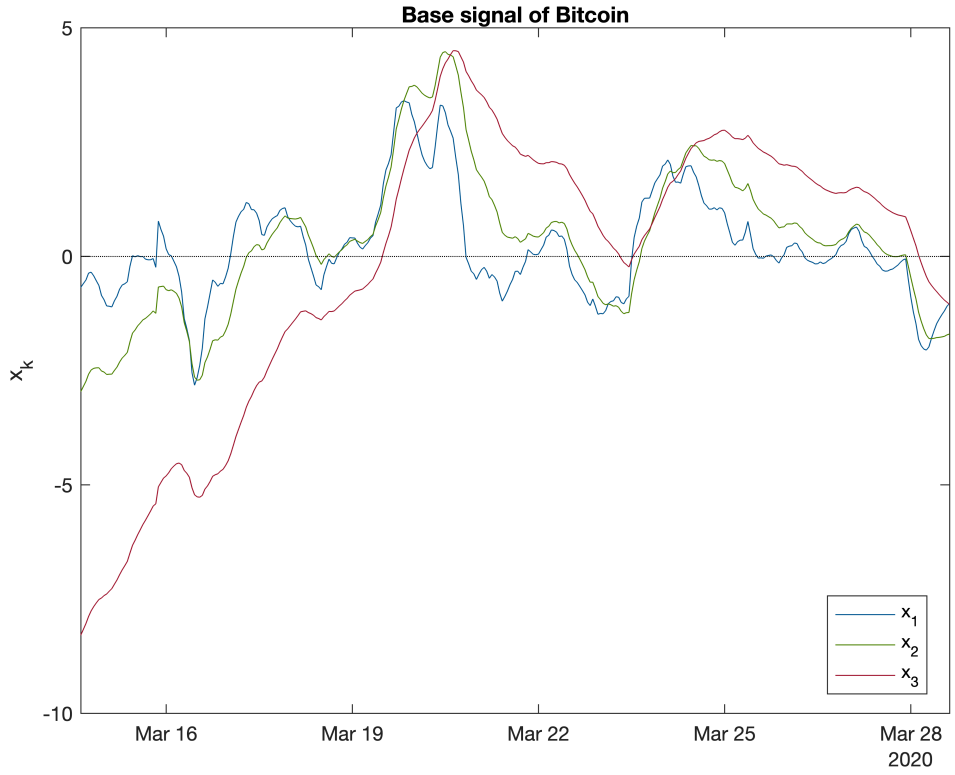
Its sign determines whether one goes long or short in this currency.

$$x_k = EMA\left(P, \frac{1}{n_{k,s}}\right) - EMA\left(P, \frac{1}{n_{k,l}}\right)$$

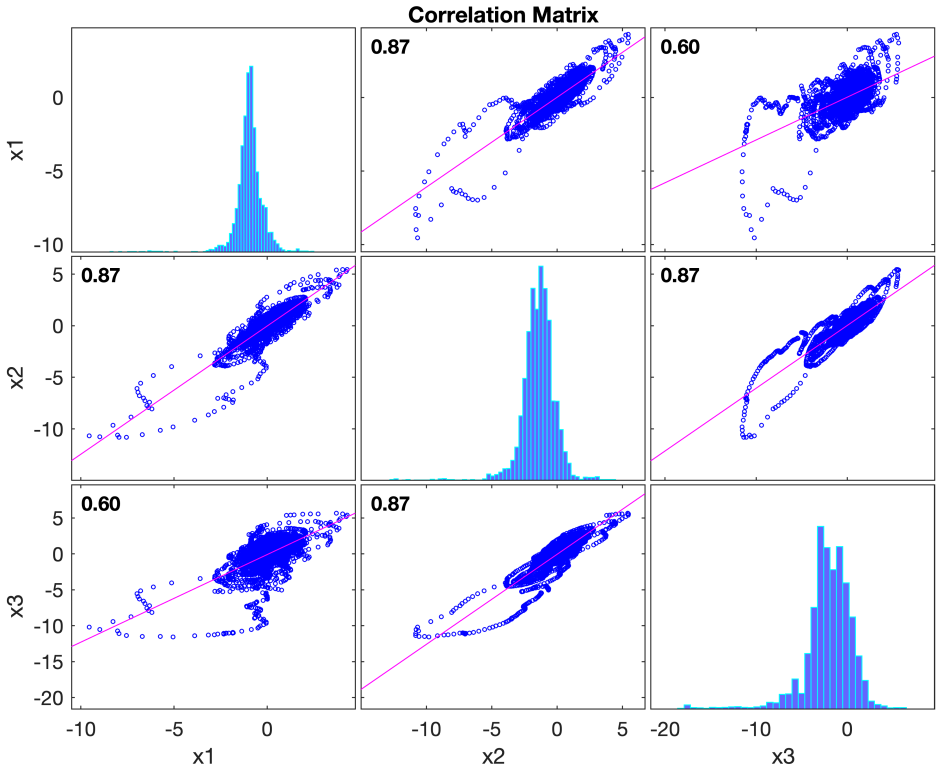
The values of x_k is

- positive: positive trend
- negative: negative trend
- equal to zero: the two EMAs intersect, trend changes direction.

```
% calculate base signal
baseSignal = timetable(symbol.time, symbol.(2)-symbol.(3),symbol.(4)-symbol.(5),symbol.

% plot
figure;
plot(baseSignal.Time,baseSignal.x1,'Color','#005891','DisplayName','baseSignal.x1');hol
hline(0,'k:');
xlim(span)
legend({'x_1','x_2','x_3'},'Location','best')
title('Base signal of '+name)
ylabel('x_k')
```




```
% correlation between x_k
corrplot(baseSignal);
```



Reducing the correlation leads to a longer time window, and vice versa.

(TODO:) One can optimize the n_k for each currency, but the risk of **overfitting** must be considered.

Two-step normalization

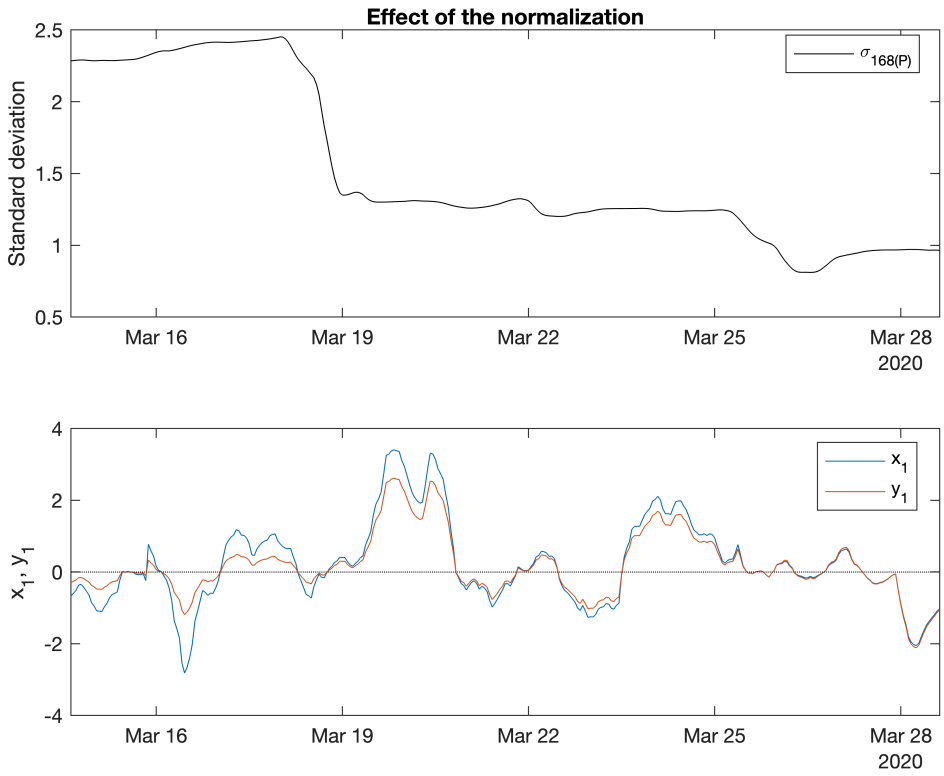
We use time frames of 168 h or one week (short) rolling standard deviations, and 1440 h or two month (long) rolling standard deviations with a 20% hold-out.

$$y_k = \frac{x_k}{\sigma_{168}(P)}, \quad z_k = \frac{y_k}{\sigma_{1440}(P)}$$

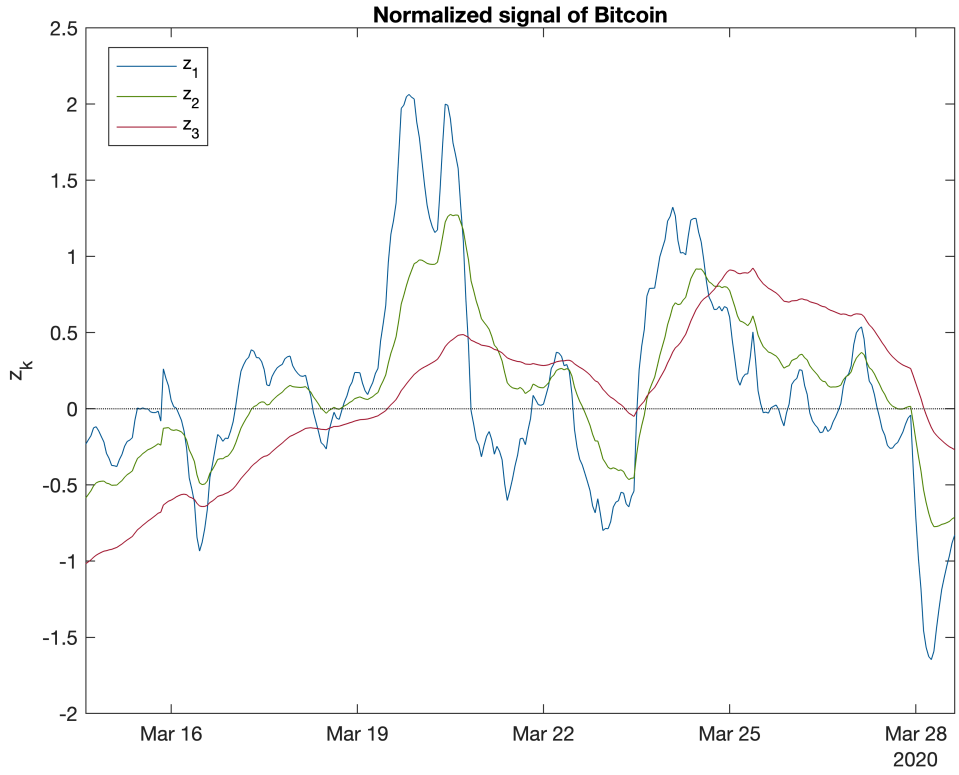
First transformation lowers high volatitly and amplifies low volatility. The second's effect is similar.

```
norm = baseSignal;
shortStd = movstd(norm.Variables,[134 33]);  % 168  = 134  +1 +33
longStd = movstd(norm.Variables,[1152 287]); % 1440 = 1152 +1 +287

yBaseSignal = norm.Variables./shortStd;
zBaseSignal = yBaseSignal./longStd;
norm(:,["y1" "y2" "y3"]) = array2table(yBaseSignal);
norm(:,["z1" "z2" "z3"]) = array2table(zBaseSignal);
norm(:,["std_s1" "std_s2" "std_s3"]) = array2table(shortStd);
norm(:,["std_l1" "std_l2" "std_l3"]) = array2table(longStd);

% plot effect of normalization
figure;
subplot(2,1,1);
```

```
plot(norm.Time,norm.std_s1,'k')
xlim(span)
legend({'\sigma_{168(P)}'},'Location','best')
title('Effect of the normalization')
ylabel('Standard deviation')
subplot(2,1,2);
plot(norm.Time,norm.x1,'DisplayName','norm.x1');hold on;plot(norm.Time,norm.y1,'Display
xlim(span)
hline(0,'k:');
legend({'x_1','y_1'},'Location','best')
ylabel('x_1, y_1');
```




```
% plot z_k
figure;
plot(norm.Time,norm.z1,'Color','#005891','DisplayName','norm.z1');hold on;plot(norm.Tim
hline(0,'k:');
xlim(span)
legend({'z_1','z_2','z_3'},'Location','best')
title('Normalized signal of '+name)
ylabel('z_k')
```

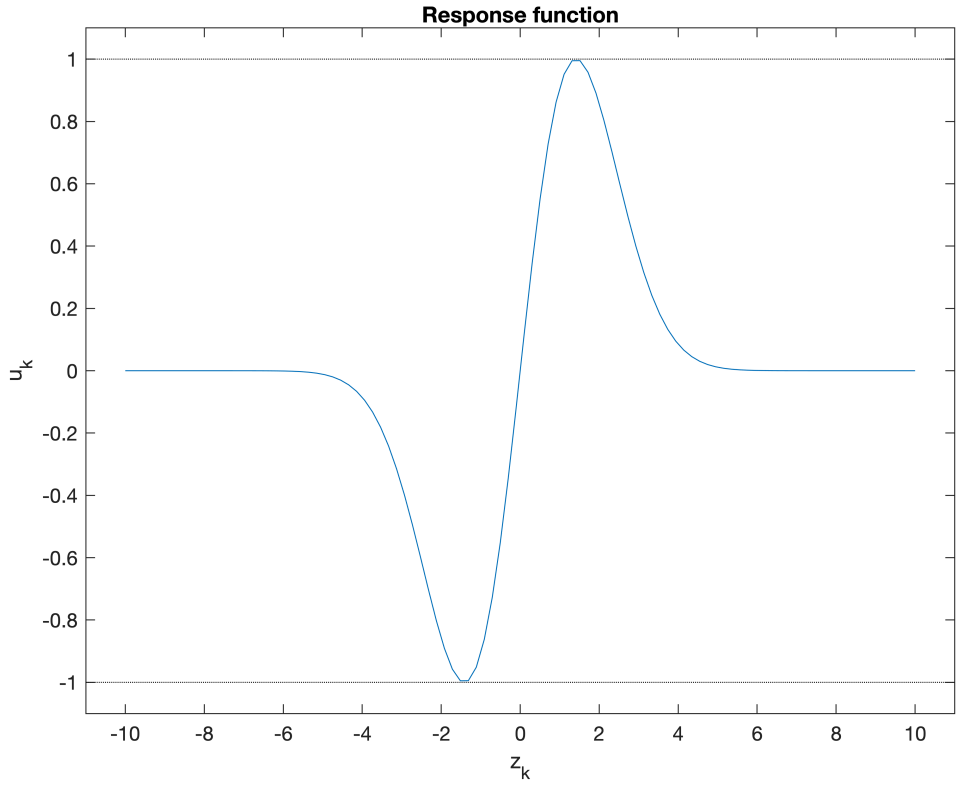


Generate signal

To scale z_k between -1 and 1, we use **response function** proposed by Rohrbach et al. (2017), which used geometric Brownian Motion.

$$u_k(z_k) = \frac{z_k \cdot e^{-z_k^2/4}}{\sqrt{2} \cdot e^{-1/2}}$$

```
z = linspace(-10,10);
u = (z.*exp(-z.^2/4)) / (sqrt(2)*exp(-1/2));
figure;
plot(z,u);
xlim([-11 11])
ylim([-1.1 1.1])
hline(1,'k:');
hline(-1,'k:');
title('Response function')
xlabel('z_k')
ylabel('u_k')
```



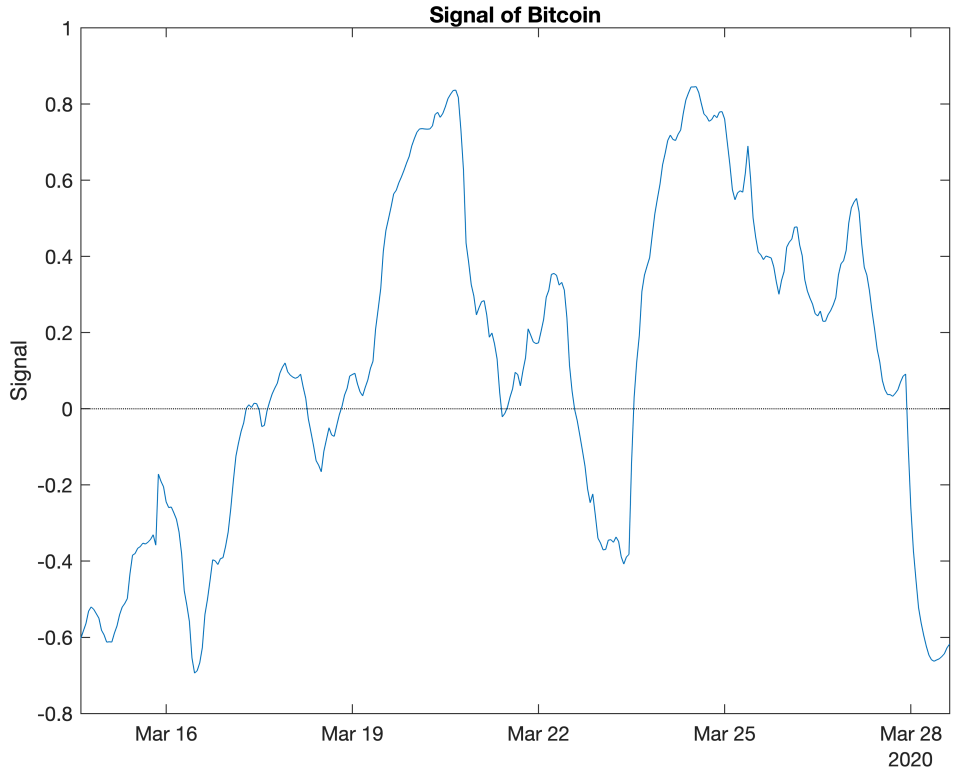

```
z_k = norm{:,["z1" "z2" "z3"]};
u_k = (z_k.*exp(-z_k.^2/4)) / (sqrt(2)*exp(-1/2));
```

Final signal is calculated as

$$Signal = \sum_{k=1}^{3} w_k u_k$$

with equal weighted $w_k = 1/3$.

```
signal = timetable(norm.Properties.RowTimes,mean(u_k,2));
signal.Properties.VariableNames = {'Signal'};
plot(signal.Time,signal.Signal);
xlim(span);
hline(0,'k:');
ylabel('Signal');
title('Signal of '+name);
```



Create signal for all cryptos

```
symbolsEma = {bitcoinEma,ethereumEma,rippleEma,bitcoincashEma,tetherEma,litecoinEma,eos

% table of signals
signalT = genRohrbachSignal(symbolsEma);
head(signalT)
```

```
ans = 8×7 timetable
```

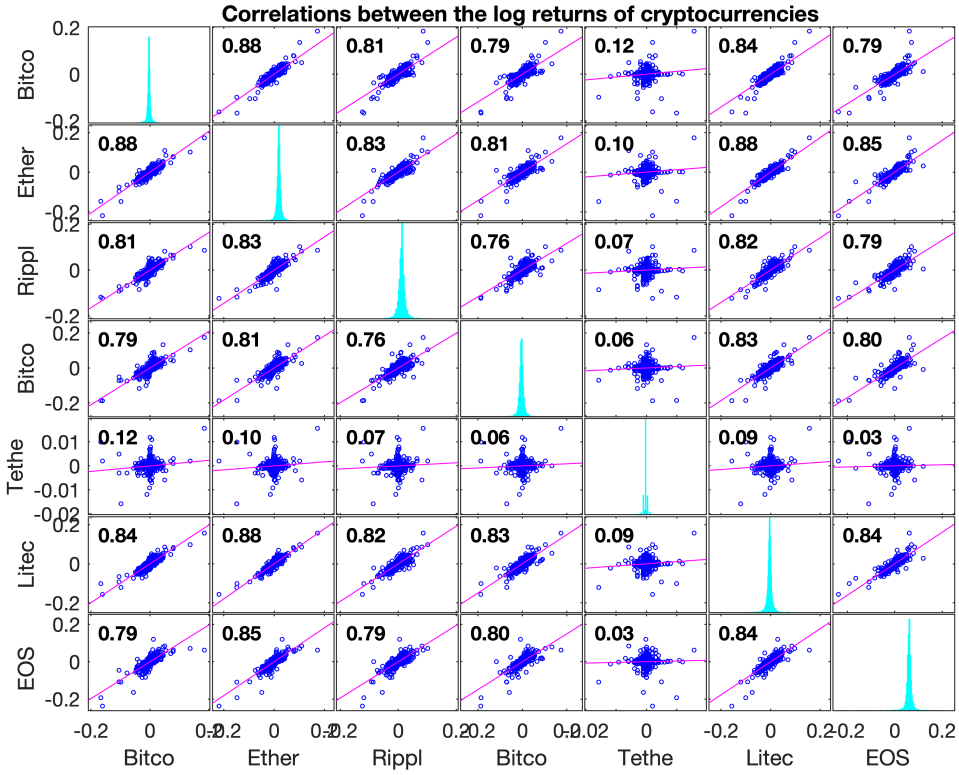
...

	Time	Bitcoin	Ethereum	Ripple	BitcoinCash	Tether	Litecoin
1	2019-10-...	0	0	0	0	0	0
2	2019-10-...	-0.3826	-0.0197	-0.2756	-0.3244	8.9672e-12	-0.1835
3	2019-10-...	-0.6693	-0.2524	-0.3150	-0.5926	1.8074e-11	-0.4759
4	2019-10-...	-0.7596	-0.5990	-0.5407	-0.6709	-4.2300e-09	-0.6628
5	2019-10-...	-0.7043	-0.6551	-0.7059	-0.6285	-8.6794e-08	-0.7370
6	2019-10-...	-0.6778	-0.6397	-0.7283	-0.5931	-3.5904e-30	-0.7759
7	2019-10-...	-0.5665	-0.6094	-0.7108	-0.5591	-4.3093e-64	-0.7870
8	2019-10-...	-0.5258	-0.5948	-0.7001	-0.5534	-1.0021e-106	-0.8177

Portfolio types

We find only Tether has weak correlationwith other cryptos' returns.

```
[rtn, intv] = tick2ret(crypto,'Method','Continuous'); % first timestamp dropped

% correlation
corrplot(rtn)
title('Correlations between the log returns of cryptocurrencies');
```



Times series port

Rebalancing every hour: buy (or sell) **1/n of the signal** units of US dollars for a cryptocurrency. n=7.

$$tsReturn_t = Signal_{t-1} \times \frac{1}{7} \times \frac{1}{P_{t-1}} \times R_t$$

where $1/P_{t-1}$ coincide with the indexed exchange rate of USD/crypto at signal period, R_t is the log return of indexed price at time t.

```
tsPort = signalT{:,:} ./7 ./crypto{:,:}; % cryptos
tsPort = array2table(tsPort,'VariableNames',{'Bitcoin','Ethereum','Ripple','BitcoinCash
tsPort = table2timetable(tsPort,"RowTimes",signalT.Properties.RowTimes);

tsRtn = tsPort{1:end-1,:} .* rtn{:,:};
tsRtn = array2table(tsRtn,'VariableNames',{'Bitcoin','Ethereum','Ripple','BitcoinCash',
tsRtn = table2timetable(tsRtn,"RowTimes",signalT.Properties.RowTimes(1:end-1));
```

Cross-sectional port

Each hour, we long the three cryptocurrencies with the largest signals, and short three with the most negative signals. These 6 trades are equally weighted in USD.

$$csReturn_t = \pm\frac{1}{6} \times \frac{1}{P_{t-1}} \times R_t$$

```
[B,posIdx] = maxk(signalT{:,:},3,2);
filterSignalT = signalT{:,:};
[C,negIdx] = mink(signalT{:,:},3,2);
for i = 2:length(filterSignalT)
    filterSignalT(i,posIdx(i,:)) =  1;
    filterSignalT(i,negIdx(i,:)) = -1;
end
filterSignalT((filterSignalT~=1) & (filterSignalT~=-1)) = 0;

csPort = filterSignalT(:,:) ./6 ./crypto{:,:};
csPort = array2table(csPort,'VariableNames',{'Bitcoin','Ethereum','Ripple','BitcoinCash
csPort = table2timetable(csPort,"RowTimes",signalT.Properties.RowTimes);

csRtn = csPort{1:end-1,:} .* rtn{:,:};
csRtn = array2table(csRtn,'VariableNames',{'Bitcoin','Ethereum','Ripple','BitcoinCash',
csRtn = table2timetable(csRtn,"RowTimes",signalT.Properties.RowTimes(1:end-1));
```

Benchmark

Equally weighted investment.

```
bmPort = 1 /7 ./crypto{:,:};
bmPort = array2table(bmPort,'VariableNames',{'Bitcoin','Ethereum','Ripple','BitcoinCash
bmPort = table2timetable(bmPort,"RowTimes",signalT.Properties.RowTimes);

bmRtn = rtn{:,:} .* bmPort{1:end-1,:};
bmRtn = array2table(bmRtn,'VariableNames',{'Bitcoin','Ethereum','Ripple','BitcoinCash',
bmRtn = table2timetable(bmRtn,"RowTimes",signalT.Properties.RowTimes(1:end-1));
```

Transaction Cost

(Lintilhac 2017) (Platanakis 2019)

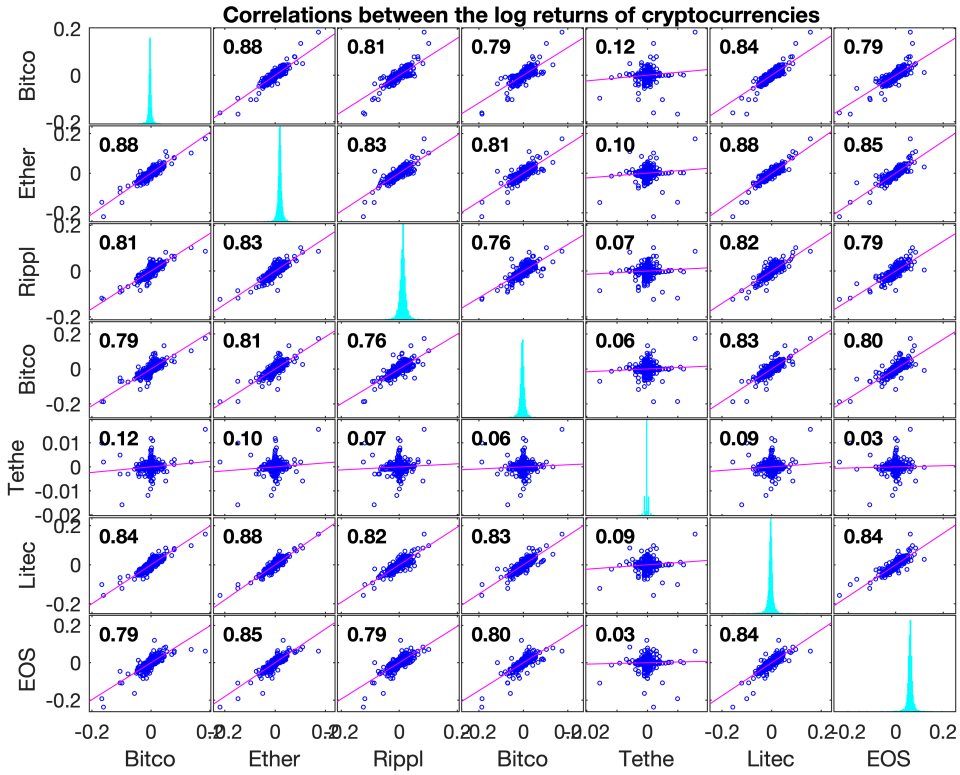
```
riskFreeRate      =   0.01/252; % PERCENT PER DAY
managementFeeRate =   0.01/252; % PERCENT PER DAY
```

(TODO) Market impact

```

```

Reorder colors. (Original third color yellow was hard to distinguish from the second color orange).

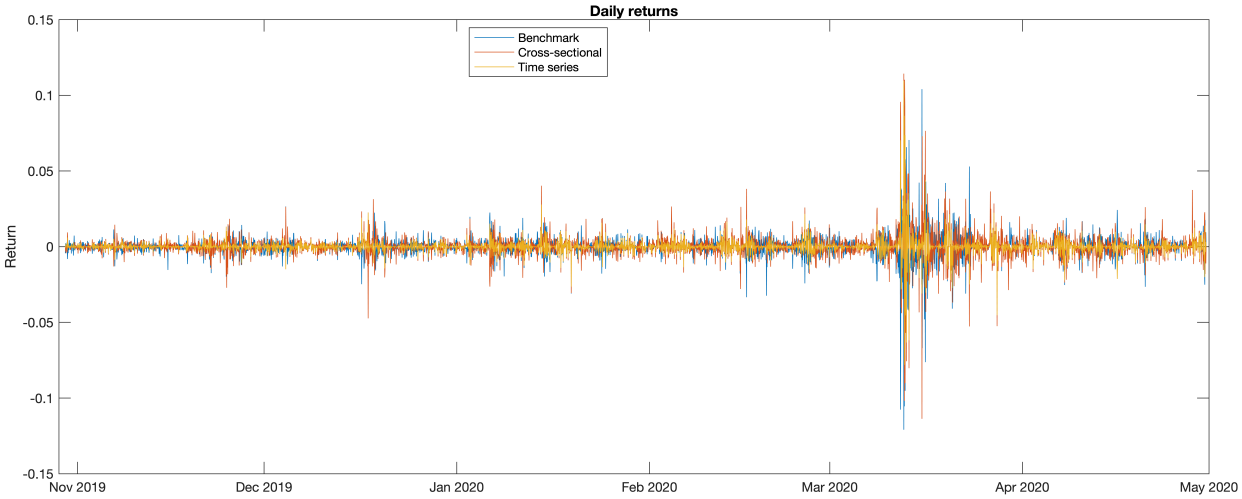
```
newcolors = [0 0.4470 0.7410
             0.8500 0.3250 0.0980
             0.4660 0.6740 0.1880];

colororder(newcolors)
```



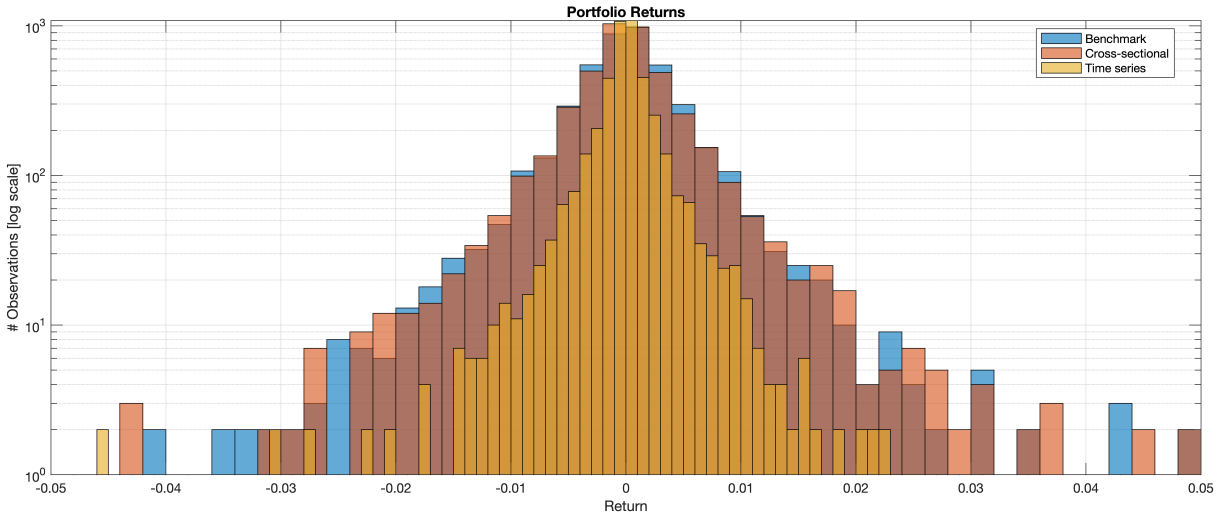
Performance Analysis

Returns

```
tsTotalRtn = sum(tsRtn{:,:},2);
csTotalRtn = sum(csRtn{:,:},2);
bmTotalRtn = sum(bmRtn{:,:},2);

% plot
f=figure;
f.Position(3)= 2*f.Position(3);
plot(indexedPrice.time(1:end-1),[bmTotalRtn csTotalRtn tsTotalRtn])
title('Daily returns');
legend({'Benchmark','Cross-sectional','Time series'},'Location','best');
ylabel('Return');
xlim(xrange)
```



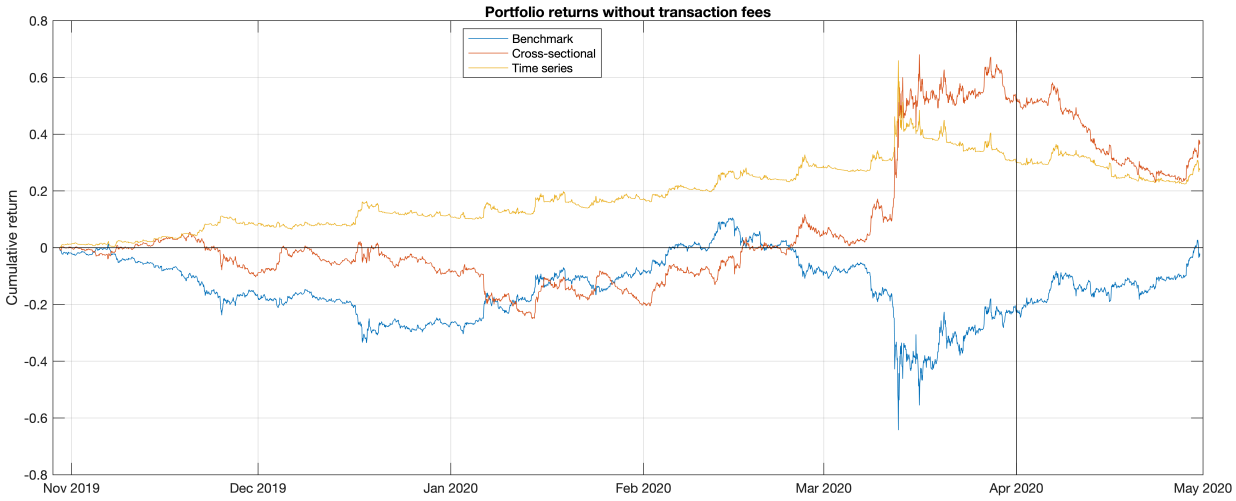

```
% histogram
histogram(bmTotalRtn); hold on;
histogram(csTotalRtn); hold on;
histogram(tsTotalRtn);
set(gca, 'YScale', 'log');
xlabel('Return');
ylabel('# Observations [log scale]');
title('Portfolio Returns');
legend({'Benchmark','Cross-sectional','Time series'},'Location','best');
grid on
% vline(0,'k-')
xlim([-5e-2 5e-2])
hold off;
```



Equity

Cross-sectional portfolio obtained highest cumulative return (without transaction cost) before April 2020.

```
% without transaction cost
tsCumRtn = cumsum(tsTotalRtn(:,:));
csCumRtn = cumsum(csTotalRtn(:,:));
bmCumRtn = cumsum(bmTotalRtn(:,:));

% plot
plot(indexedPrice.time(1:end-1),[bmCumRtn csCumRtn tsCumRtn])
grid on
hline(0,'k-');
vline(datetime(2020,4,1),'k-');
title('Portfolio returns without transaction fees');
legend({'Benchmark','Cross-sectional','Time series'},'Location','best');
ylabel('Cumulative return');
xlim(xrange)
```




```
% with transaction cost
```

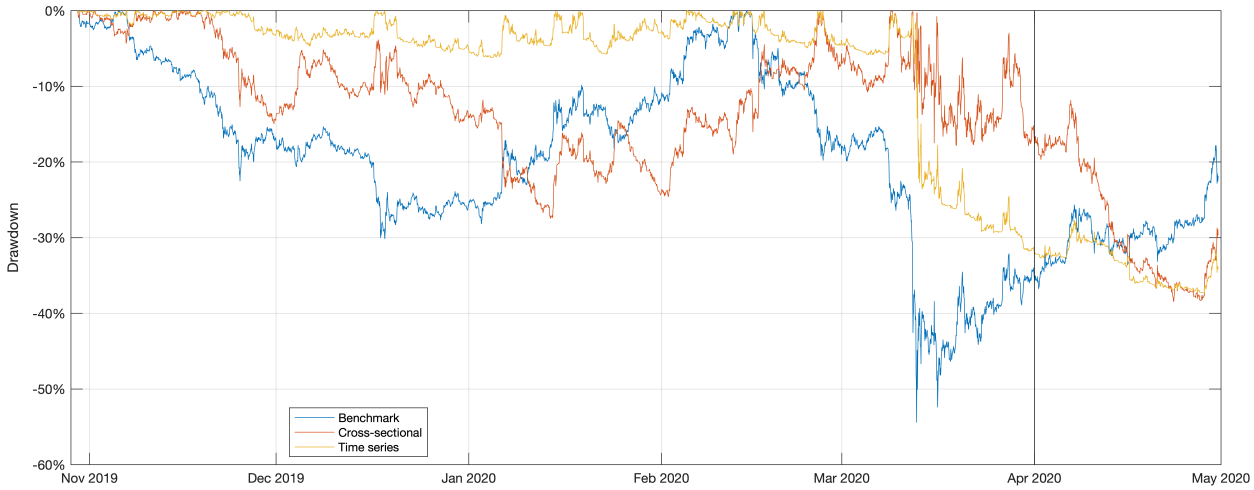
Drawdowns

"A drawdown is a peak-to-trough decline during a specific period for an investment, trading account, or fund. A drawdown is usually quoted as the percentage between the peak and the subsequent trough." (Investopedia)

We can see **cross-sectional** portfolio has least drawdown due to the COVID-19.

```
names = {'Benchmark','Cross-sectional','Time series'};
tsDD = 100.*-calculateDD(tsTotalRtn);
csDD = 100.*-calculateDD(csTotalRtn);
bmDD = 100.*-calculateDD(bmTotalRtn);

plot(rtn.time,[bmDD csDD tsDD]);
vline(datetime(2020,4,1),'k-');
ylabel('Drawdown');
ytickformat('%5.0f%%');
legend(strrep(names,'_',' '),'location','best');
```

```
grid on
xlim(xrange)
```



Performance summary

```
retns = table(bmTotalRtn,csTotalRtn,tsTotalRtn);
summaryPerformance(retns,riskFreeRate)
```

ans = 5×3 table

	bmTotalRtn	csTotalRtn	tsTotalRtn
1 TotalRetn	-0.0224	0.4442	0.3195
2 SharpeRatio	-0.0055	0.0053	0.0044
3 AverageRetn	-0.0000	0.0001	0.0001
4 Volatility	0.0081	0.0082	0.0052
5 MaxDrawdown	0.7486	0.4511	0.4361

Discussion and future work

Discussion

Strengths:

- Cross-sectional portfolio has the highest Sharpe ratio
- Time series portfolio performs better when not in crisis

Weakness:

- No single strategy beats others the whole time
- Long term Sharpe raios are small
- Asset selection is fixed

Future work

- **Data**: use TAQ data
- **Model**: Determine the mechanics set two strategies apart during crisis
- **Backtesting**: Take more specific transaction cost and market impact into account

Reference

1. Chu J, Chan S, Zhang Y. (2020) High frequency momentum trading with cryptocurrencies. *Research in international business and finance*. 52:101176.
2. Rohrbach, Janick and Suremann, Silvan and Osterrieder, Joerg (2017) Momentum and Trend Following Trading Strategies for Currencies and Bitcoin. *Combining Academia and Industry.*
3. P. S. Lintilhac & A. Tourin (2017) Model-based pairs trading in the bitcoin markets. *Quantitative Finance.* 17:5, 703-716.
4. Platanakis, Emmanouil & Urquhart, Andrew (2019) Portfolio management with cryptocurrencies: The role of estimation risk. *Economics Letters*. ISSN: 0165-1765, Volume 177, p. 76.

Local Functions

No need to include this part in slides.

Rohrbach Signal

```
function rohrbachSignal = genRohrbachSignal(symbolsEma)
    for i = 1:length(symbolsEma)
        symbol = symbolsEma{i};
        tick = symbol.Properties.VariableNames(1);
        tick = string(tick);
        baseSignal = timetable(symbol.time, symbol.(2)-symbol.(3),symbol.(4)-symbol.(5)

        shortStd = movstd(baseSignal.Variables,[134 33]);  % 168  = 134  +1 +33
        longStd = movstd(baseSignal.Variables,[1152 287]); % 1440 = 1152 +1 +287
        y = baseSignal.Variables./shortStd;
        z = y./longStd;
        u = (z.*exp(-z.^2/4)) / (sqrt(2)*exp(-1/2));

        signal = timetable(baseSignal.Properties.RowTimes,mean(u,2));
        signal.Properties.VariableNames = tick;

        rohrbachSignal(:,i) = signal;
        rohrbachSignal.Properties.VariableNames(i) = tick;
    end
end
```

Calculate Drawdowns

```
function DD = calculateDD(r)
Px = 100.*(cumprod(1+r)); % PRICE
for j=1:size(r,2)
    for i=1:length(Px)
        if i==1
```

```
            c(1,j) = Px(1,j);
            DD(1,j) = 0;
        else
            c(i,j) = max(Px(i,j),c(i-1,j));
            if c(i,j)==Px(i,j)
                DD(i,j)=DD(i-1,j);
            else
                DD(i,j) = 1-Px(i,j)./c(i,j);
            end
        end
    end
end
end
```

Performance of strategy

```
function performanceTable = summaryPerformance(returnTable,riskFreeRate)
    % compute performance metrics
    retns = returnTable{:,:};
    MeanRetn = mean(retns);
    StdRetn = sqrt(var(retns));
    SharpeRatio = (MeanRetn-riskFreeRate)./StdRetn;

    cmpRetn = ret2tick(retns, 'method', 'Continuous');
    MaxDD = maxdrawdown(cmpRetn,'geometric');
    TotalRetn = cmpRetn(end, :)-1;
    perfData = [TotalRetn', SharpeRatio', MeanRetn', StdRetn', MaxDD']';

    metricNames = {'TotalRetn', 'SharpeRatio', 'AverageRetn', 'Volatility', 'MaxDrawdow
    colStratsNames = (returnTable.Properties.VariableNames)';


    performanceTable = array2table(perfData, 'VariableNames', colStratsNames, 'RowNames

end
```