Question

Use the stocks in SSE 50 to demonstrate whether the theory Beta still applies to current stock market.

Result

In the task, experimental raw data can be derived from **CSMAR database**, **API of Tencent**, and also from http://www.sse.com.cn/market/sseindex/quotation/. In this task, the average annual return of SSE 50 equals to **14.84%**. In conclusion, the results cannot satisfy **equation (1)** below:

$$\boldsymbol{\beta_H \mu_L - \beta_L \mu_H = 0}$$

Where $\beta_H \mu_L$ are nearly twice as much as $\beta_L \mu_H$.

Therefore, the theory Beta does not apply to current stock market in Shanghai, China.

Solution

1. Plot Net Return Index (2004-2018), save as sse50.mat.
MATLAB script: read_sse50.m

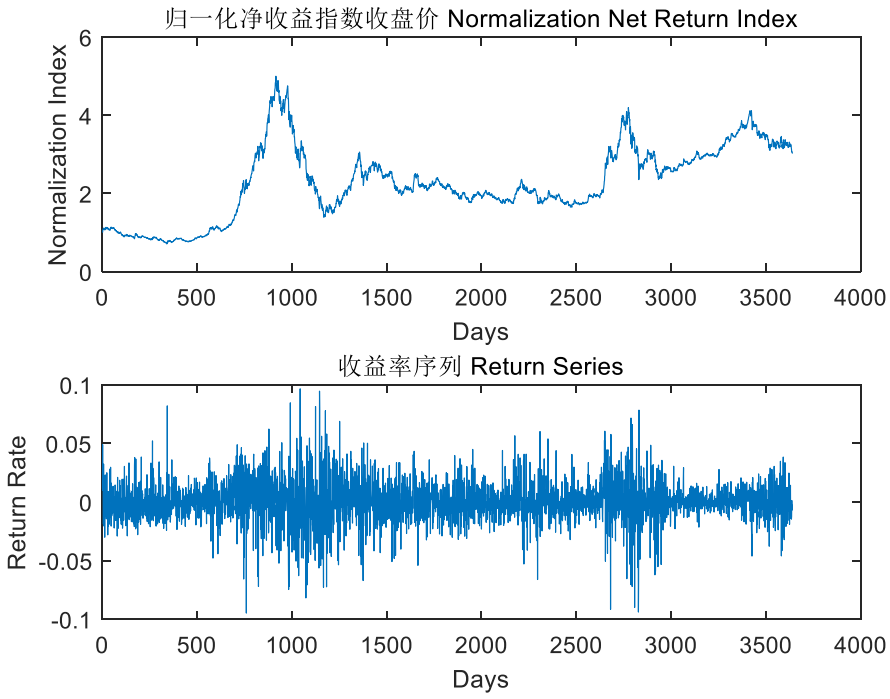

Figure. 1 SSE 50

2. Load sse50.mat, calculate beta and alpha.

MATLAB script: tmp.m

Firstly, I download SSE 50 Index stocks data (2013-2018) from CSMAR in order to calculate beta and alpha in 6 years. I'll show you one graph of them. In **Figure.2**, the two graphs on left represent 600016.ss and the two on right represent SSE 50.

Then, I found it troublesome to download every stock from CSMAR and use API from Tencent to replace download process. Therefore, I can get annual data of each stock. In **Table. 1 & 2**, I show part of the Beta Matrix and Alpha Matrix, respectively. Some stocks are not listed in certain years, so I use NaN to fill up the blank (**Table. 3**), including 601138.ss (2013-2017), 601211.ss(2013-2014), 601229.ss (2013-2015) and 603259.ss (2013-2017).

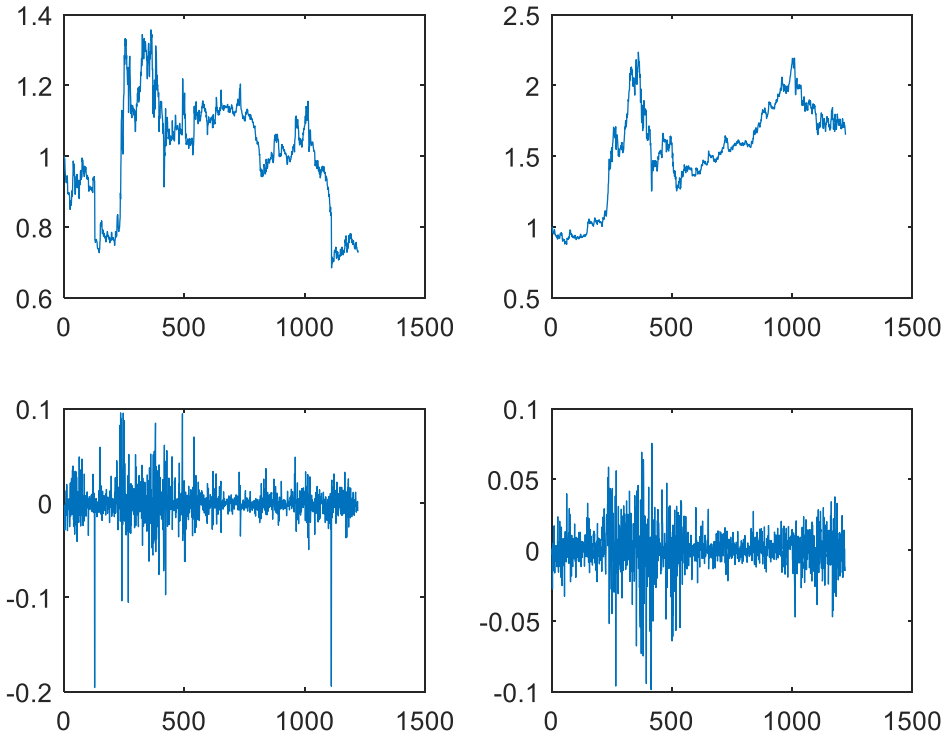

Figure. 2 Comparison between 600016.ss and SSE 50

Table. 1 Beta Matrix (Part of the Whole)

StockNum	2013	2014	2015	2016	2017
600000	0.548186	0.622894	0.784687	0.276287	0.101098
600016	0.535115	0.410773	0.752369	0.667658	0.432219
600019	0.830964	0.428901	0.501115	0.356612	0.105779
600028	0.288677	0.442709	0.740096	0.6979	0.216657
600029	0.718415	0.329711	0.314776	0.287255	0.128524
600030	0.539416	0.408174	0.538641	0.493484	0.309283
600036	0.627487	0.690203	0.790749	0.728499	0.314553

600048	0.447752	0.209198	0.533145	0.40359	0.122627
600050	0.571671	0.552052	0.456709	0.292477	0.180271
600104	0.419652	0.448416	0.721038	0.47786	0.207162

Table. 2 Alpha Matrix (Part of the Whole)

StockNum	2013	2014	2015	2016	2017
600000	0.000719	-0.00027	0.000908	-0.00038	-0.00138
600016	0.000846	-0.00105	-0.00013	-5.7E-05	-0.00107
600019	-0.00053	0.000611	-0.00051	0.001133	0.000683
600028	-0.00128	-5.3E-06	-0.00112	0.000438	2.39E-05
600029	-0.00091	0.001055	0.002874	-0.00037	0.001524
600030	0.000843	0.001037	-0.00171	-0.00045	-0.00046
600036	-0.00014	-0.00037	0.000649	-1.4E-05	0.000331
600048	-0.00149	-0.00102	0.000261	-0.00026	0.001123
600050	3.27E-05	-0.00018	0.001434	0.001022	-0.00209
600104	-0.00028	2.99E-06	-2.8E-05	0.000666	0.000366

Table. 3 Samples of Not Listed Stocks (Part of the Whole)

601211	NaN	NaN	0.00027	-0.00079	-0.00082
601229	NaN	NaN	NaN	-0.00256	-0.00268

3. Analyze the annual return.

MATLAB script: return_analysis.m

```
% 1. Load data from 2014-2018, form 10 quantile portfolios by sorting all Betas from the
% smallest to the largest
% 2. Suppose that Beta won't change in Year.N+1 and analyze the portfolio
% returns for each day of Year.N+1
```

In Table. 4, I give the average portfolios betas. **While betas seem too small in 2017 and I cannot give a reasonable explanation**, others are acceptable. In Figure. 3, I give the average annual return bar of each portfolio. The average annual return of SSE 50 equals to **14.84%**.

Table.4 Average Portfolios Betas

Beta	2013	2014	2015	2016	2017
1	0.140916	0.097078	0.234036	0.131625	0.009995
2	0.239055	0.174771	0.311313	0.284287	0.054475
3	0.316443	0.25904	0.351681	0.328093	0.099828
4	0.425372	0.351449	0.450176	0.351942	0.126216
5	0.510945	0.405625	0.522242	0.399919	0.173707
6	0.576785	0.451857	0.570204	0.468712	0.200356

7	0.65232	0.510385	0.641138	0.529362	0.229793
8	0.716088	0.551877	0.690072	0.704432	0.288107
9	0.811518	0.597398	0.749367	0.838542	0.31833
10	0.929638	0.774062	0.801732	0.922403	0.45314

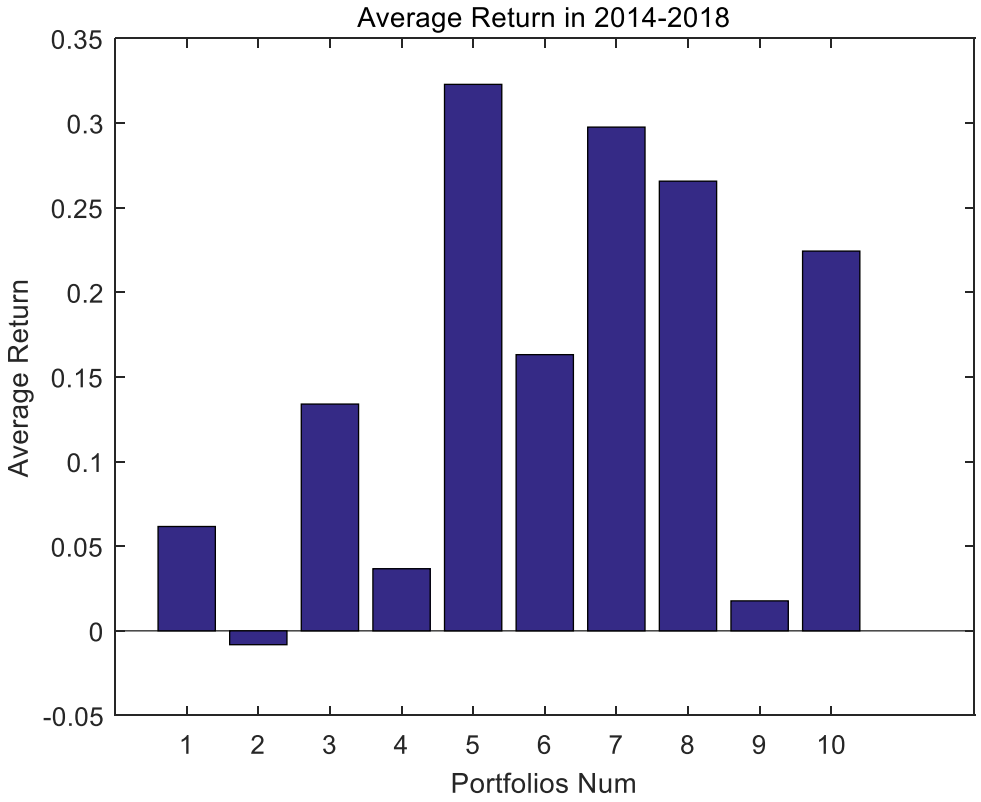

Figure. 3 Average Return of Each Portfolio in 2014-2018

Appendix: Code

MATLAB script: read_sse50.m

```
%文件信息
[status,sheets,format]=xlsfinfo('N00016.xls');
%读取数据
[num,txt,raw]=xlsread('N00016.xls');
sse50=flipud(num);
[r,c]=size(num);
Date=datenum(sse50(1:end,1));

save sse50 sse50

load sse50
figure
hold on
a=sse50(r,2);
subplot(2,1,1)
%归一化
plot(sse50(:,2)/sse50(1,2))
title('归一化净收益指数收盘价  Normalization Net Return Index')
xlabel('Days')
ylabel('Normalization Index')
subplot(2,1,2)
%收益率序列
% Rate=price2ret(sse50(:,2));
Rate=ret(sse50(:,2));
plot(Rate)
title('收益率序列  Return Series')
xlabel('Days')
ylabel('Return Rate')
```

MATLAB script: tmp.m

```
clear;
clc;

%%   load sse50 stock numbers，save as fileNamesNum.mat

fileFolder=fullfile('G:\MS\Beta&Alpha\xls');
dirOutput=dir(fullfile(fileFolder,'*.xls'));%如果存在不同类型的文件，用‘*’读取所有，如果读取特定类型文件，'.'加上文件类型，例如用‘.jpg’
fileNames={dirOutput.name}';
```

```
[r,c]=size(fileNames);
fileNamesNum=zeros(r,c);
for i=1:r
    a=char(fileNames(i,1));
    A=a(isstrprop(a,'digit'));
    B=str2double(A);
    fileNamesNum(i,1)=B;
end
% a=2e7;

%%   Load sse50 && Calculate Continuous Growth Rate

load sse50
sse50_Date=rem(sse50(:,1),2e7);
Rate=price2ret(sse50);

%%   Calculate Beta and Alpha

    % Create Matrix for Beta & Alpha
    Beta=zeros(length(fileNamesNum)+1,6);
    Alpha=zeros(length(fileNamesNum)+1,6);

for Year=2013:2017
% Year=2017;

    Beta(1,Year-2012+1)=Year;
    Alpha(1,Year-2012+1)=Year;

    for i=1:length(fileNamesNum)

        stockName=fileNamesNum(i);
        Beta(i+1,1)=stockName;Alpha(i+1,1)=stockName;
        temp=mat2str(stockName);
        stockInfo=tencent_history(temp,Year);
        stock_Date=stockInfo(:,1);     %Stock Dates Array

        % x=rem(Year,2000);y=fix(stockInfo(1,1)/1e4);
        if stockInfo~=0

        finalTable=zeros(length(stock_Date),3);

            for j=1:length(stock_Date)
                a=stock_Date(j);
```

```
                    if find(sse50_Date==a)
                        [m,n]=find(sse50_Date==a);
                        finalTable(j,1)=a;
                        finalTable(j,2)=sse50(m,2);
                        finalTable(j,3)=stockInfo(j,3);
                    else
                        finalTable(j,:)=0;
                    end

                end         % end for j

                % Delete 0
                    [p,q]=find(~finalTable(:,1));
                    if p
                                finalTable(p,:)=[];
                    end

                % Calculate Rate of Return
                Rate_sse50=ret(finalTable(:,2));
                Rate_stock=ret(finalTable(:,3));

                % Calculate Beta & Alpha
                beta_tmp=portbeta(Rate_sse50,Rate_stock);
                Beta(i+1,Year-2012+1)=beta_tmp;

                % set risk-free rate 0
                Cash=zeros(length(Rate_sse50),1);
                % Calculate alpha without risk-adjusted return
                % [alpha,RAReturn]=portalpha(Rate_Stock,Rate_sse50,Cash,'xs');
                % Calculate alpha with risk-adjusted return

[alpha_tmp,RAReturn_tmp]=portalpha(Rate_stock,Rate_sse50,Cash,'capm');

                Alpha(i+1,Year-2012+1)=alpha_tmp;

            else

                % data acquirement failed
                Beta(i+1,Year-2012+1)=NaN;Alpha(i+1,Year-2012+1)=NaN;

            end         % end if StockInfo~=0

        end             % end for i
end                     % end for Year
```

```
save Beta_1 Beta
save Alpha_1 Alpha
```

MATLAB script: return_analysis.m

```
load sse50
load fileNamesNum
load Beta_1

% load data from 2014-2018
% 1. Calculate the Beta and Alpha in Year.N(2013-2017)
% form 10 quantile portfolios by sorting all Betas from the smallest to the largest
% 2. Suppose that Beta won't change in Year.N+1 and analyze the portfolio
% returns for each day of Year.N+1

q=10;     %10-quantiles
basic=2013;
Num=length(fileNamesNum)/q;
Beta_mean=zeros(q+1,Num+1); Return=zeros(q+1,Num+1);

for Year=2014:2018

    Beta_seq=sortrows(Beta,Year-basic+1);
    [m,n]=find(Beta_seq(:,1)==0);
    size_1=fix((m-1)/q); size_2=rem(m-1,q);
    portfoliosList=zeros(Num+1,q);
    portfoliosList(1,:)=1:10; Beta_mean(:,1)=0:10; Return(:,1)=0:10;
    Beta_mean(1,Year-basic+1)=Year-1;Return(1,Year-basic+1)=Year;

    for i=1:size_2
        a=0;
        for j=1:Num
            portfoliosList(j+1,i)=Beta_seq((i-1)*Num+j,1);
            a=a+Beta_seq((i-1)*Num+j,Year-basic+1);
        end % end for j
        a=a/Num;
        Beta_mean(i+1,Year-basic+1)=a;
    end % end for i

    div=size_2*Num;

    for i=size_2+1:q
        a=0;
        for j=1:size_1
```

```
                portfoliosList(j+1,i)=Beta_seq(div+(Num-1)*(i-size_2-1)+j,1);
                a=a+Beta_seq(div+(Num-1)*(i-size_2-1)+j,Year-basic+1);
            end % end for j
            a=a/size_1;
            Beta_mean(i+1,Year-basic+1)=a;
        end % end for i

        % Calculate Return

        for x=1:q
            startDay=0; endDay=0;
            for y=2:Num+1
                if portfoliosList(y,x)>=6e5
                    stock_code=mat2str(portfoliosList(y,x));
                    temp=tencent_history(stock_code,Year);
                    endDay=endDay+temp(end,3);startDay=startDay+temp(1,3);
                end % end if
            end % end for y=1:Num
            % Annual Return
            annualReturn=(endDay-startDay)/startDay;
            Return(x+1,Year-basic+1)=annualReturn;
        end % end for x=1:q

    end % end Year

    averageReturn=zeros(1,10);
    for i=1:q
        R=Return(i+1,2:end);
        averageReturn(1,i)=mean(R);
    end
    bar(averageReturn)
    title('Average Return in 2014-2018')
    xlabel('Portfolios Num')
    ylabel('Average Return')
```