PORTFOLIO THEORY (STA4028Z)

Data Wrangling and Optimization in MATLAB

Where do I find MATLAB? Here!

Do I have to have MATLAB installed on my computer right now? No, you can use MALAB Online!

Table of Contents

Recommended Lecture Readings:

- Carmona, R., (2014) Statistical Analysis of Financial Data in R, Springer

Recommended Course Reading:

- Elton, E., Gruber, M., Brown, S., and Goetzmann, W., (2014), Modern Portfolio Theory and Investment Analysis, 9-th edition, Wiley [Chapters 13, 17, 26, 28]
- Lee, W., (2000)Theory and Methodology of Tactical Asset Allocation [Chapter 2, 3, 7]
- Prigent, P-L., (2007), Portfolio Optimization and Performance analysis. Chapman & Hall [Chapter 1, ...]
- Ross, S., (2005), Neoclassical Finance, Princeton Press [Chapter 1]

Markdown

```
title = "My assignment title";
title = "Portfolio_Theory_Lecture3";
author = "My Name : 0000000 ";
outputFormat = "docx"
export(title, format = outputFormat)
```

Rules of Engagement

1. You are strongly encouraged to **workshop and discuss assignment outputs**, graphs, tables and algorithms in order to refine your thinking about assignment solutions.
2. **There is to be no code sharing**. Duplicate code will be treated as plagiarism, and as such, intellectual property theft.
3. All MATLAB code is to be included in the appendix of your assignment submission marked up in the appropriate LaTeX environment. Where appropriate referenced by code pattern in the body of the assignment.
4. All equations used should be presented and discussed using the correct TeX markup.
5. MATLAB code executed within the document is to be suppressed where appropriate. The output is what is of interest in the body of the assignment text.

Script file description

Load the Tactical Asset Allocation data from EXCEL

1. ICB Industrial Level Indices
2. ALBI (All Bond Index (ALBI) Total Return Index (TRI) Data)
3. Money Market Data: JIBAR and STEFI TRI
4. Various Indices: JSE Growth, JSE Value, JSE ALSI, JSE SRI

Situation: Load data from *.csv file and convert into timeSeries

To open Excel spreadsheet: Navigate to the folder in the current folder, right click on spreadsheet and choose "Open Outside MATLAB"

Clear environment and remove all plots

```
clc % clears the command window
close all %removes all figures
```

```
clear % clears the workspace
```

Paths

Use your user path and folders to specify the correct path using absolute paths

```
% use your default user path to create the filename and path
fileName = "PT-DATA-ALBI-JIBAR-JSEIND-Daily-1994-2017.xlsx"
```

```
fileName =
"PT-DATA-ALBI-JIBAR-JSEIND-Daily-1994-2017.xlsx"
```

```
fileName = fullfile(userpath,'\STA4028Z\Data\',fileName)
```

```
fileName =
"C:\Users\01404122\OneDrive - University of Cape Town\Documents\MATLAB\STA4028Z\Data\PT-DATA-ALBI-JIBAR-JSEIND-Daily
```

Loading the data from Excel spreadsheets

Load the dataset by sheet using readtimetable. MATLAB has many functions to read in data in the format you would like to use it in. See `readtable`, `readmatrix`, `readvars`, `readcell`.

Timetable is a type of table that associates a time with each row. Like table, the timetable data type can store column-oriented data variables that have the same number of rows. All table functions work with timetables. In addition, timetables provide time-specific functions to align, combine, and perform calculations with one or more timetables. For more information, see Create Timetables or watch Managing Time-Stamped Tabular Data with Timetables.

The loaded Excel sheets will be saved in a cell array. A cell array is a data type with indexed data containers called cells. Each cell can contain any type of data. Cell arrays commonly contain pieces of text, combinations of text and numbers from spreadsheets or text files, or numeric arrays of different sizes.

```
excelSheetNames = sheetnames(fileName);
% Preallocate cell array before looping
data{numel(excelSheetNames)} = [];
% Load the dataset by sheet
for sheet = 1:numel(excelSheetNames)
data{sheet} = readtimetable(fileName,'Sheet',excelSheetNames{sheet},VariableNamingRule='preserv
end
```

Keep only the specified list of Tickers

Find Tickers in the column names and keep the TRI only of sheets JSE ICB 0500 Indices and JSE Various Indices.

strncmp is one of the text comparison functions that can be used. Read more here.

```
variableTickers = string("J5" + (10 : 10 : 90));
entities = {'RATESTEFI', 'ALBI', 'J203', 'J500', variableTickers{:}};

for i = 1:numel(data)
    % 1. Keep TRI only in sheets 3 and 4
```

```
    if i == 3
       allVarTable = data{i};
       TRITable = allVarTable(:,(3:3:27));
       data{i} = TRITable;
    elseif i == 4
       allVarTable = data{i};
       allVarTable = removevars(allVarTable,["SOURCE68779","Var2"]);
       TRITable = allVarTable(:,(4:4:19));
       data{i} = TRITable;
    end
    % 2. Find Tickers in column names,
        % get the table properties
        opts = data{i}.Properties;
        % Preallocate array for matching
        variableMatch = zeros(size(opts.VariableNames));
        % Do a string comparison to find tickers
        if i == 2 %Match 8 letters for STEFI to prevent unwanted variables
            for k = 1:numel(entities)
                variableMatch(strncmp(opts.VariableNames,entities{k},8)) = k;
            end
        else
            for k = 1:numel(entities)
                variableMatch(strncmp(opts.VariableNames,entities{k},4)) = k;
            end
        end
        % Index for unwanted tickers
        idx = find(variableMatch==0);
        %Remove unwanted tickers from the data
        tickersToBeRemoved = opts.VariableNames(idx);
        data{i} = removevars(data{i}, tickersToBeRemoved);
        % clean up the remaining column names
        hasColon = contains(data{i}.Properties.VariableNames, ':');
        data{i}.Properties.VariableNames(hasColon) = extractBefore(data{i}.Properties.VariableN
end
```

Clean and convert into a single timetable

Combine all four tables into one using synchronize.

The synchronize function collects the variables from all input timetables, synchronizes them to a common time vector, and returns the result as a single timetable. The effect is similar to a horizontal concatenation, though the input timetables can have different row times. When the synchronize function synchronizes timetable variables to different times, it also resamples or aggregates the data in the variables using a method that you specify.

```
allDataTable = synchronize(data{1},data{2},data{3},data{4});
%Rename variables
allDataTable = renamevars(allDataTable,{'RATESTEFI','J203'}, {'STEFI','ALSI'});
```

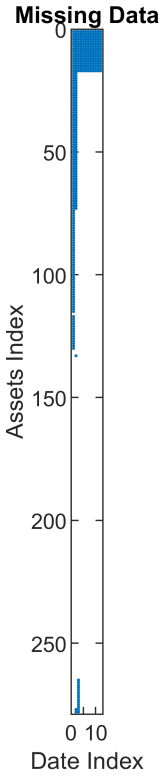
Down-sample (by decimation) and Visualise

Convert from Daily Sampled Data to Monthly Sampled Data.

```
% Decimate the daily data to monthly data
allDataTable = convert2monthly(allDataTable);
```

Remove dates in the timetable where data is not recorded, i.e holidays,weekends once you are at the required sampling frequency. First, visualise where there is missing data (e.g. a NaN).

```
figure;
subplot(1,6,1);
spy(isnan(table2array(allDataTable)));
ylabel("Assets Index")
xlabel("Date Index")
title("Missing Data")
```




```
% [left bottom width height]
%ax.OuterPosition = [ax.OuterPosition(1:3) ax.OuterPosition(4)];
```

Missing Data

The recording of ALBI and STEFI indices starts a lot later than the other indices and removing all missing data would erase any data recorded prior. There are also assets which delisted or were discontinued and hence they

no-longer have recorded data at the end of the dataset. We could fill the missing data with the last measure data set using the prices. There returns would then become zero moving forward.

```
% we are going to use zero-order hold (the missing price is the last price)
allDataTableFilled = fillmissing(allDataTable,'previous');
subplot(1,6,2);
spy(isnan(table2array(allDataTableFilled)));
```

Here we want to find that data set where all the asset are populated sequentially. Having a single missing data point is okay but more than two in a sequence is a problem. We can find a variable with the least nans and use it as a proxy to remove the missing data from the beginning of the dataset up until that point.

```
[minNans,idx] = min(sum(isnan(allDataTable{:,:}),1));
rmmissingProxy =  allDataTable.Properties.VariableNames{idx};
allDataTable = rmmissing(allDataTable,"DataVariables",rmmissingProxy);
subplot(1,6,3);
spy(isnan(table2array(allDataTable)));
```

The problem is that we see that there is still missing data in the first 3 asset, and for the 3rd asset at the end too. This asset is J500, this is the old Oil&Gas Index, the index was discontinued. Here we will remove this from our dataset.

```
idxJ500=contains(allDataTable.Properties.VariableNames,'J500');
idx = 1:width(allDataTable);
allDataTable = allDataTable(:,idx(~idxJ500))
```

allDataTable = 261×11 timetable

...

	Time	ALBI	STEFI	J510	J520	J530	J540	J550
1	30-Jun-1995	NaN	NaN	357.5400	493.6400	232.9200	512.3600	992.2600
2	31-Jul-1995	NaN	NaN	360.3800	510.1100	235.1000	527.5700	995.5000
3	31-Aug-1995	NaN	NaN	369.7800	528.5500	232.5800	543.3600	997.1400
4	30-Sep-1995	NaN	NaN	380.0600	553.0800	244.2400	555.3100	1.0460e+03
5	31-Oct-1995	NaN	NaN	398.0400	604.0900	267.4500	580.9000	1.1076e+03
6	30-Nov-1995	NaN	NaN	365.9800	620.5400	267.9800	610.0700	1.1684e+03
7	31-Dec-1995	NaN	NaN	372.7700	644.3000	285.9700	651.1400	1.2097e+03
8	31-Jan-1996	NaN	NaN	394.0600	714.8100	319.0800	681.4500	1.2730e+03
9	29-Feb-1996	NaN	NaN	355.2000	669.8800	300.6400	656.2500	1.2754e+03
10	31-Mar-1996	NaN	NaN	375.6500	664.5500	295.9300	621.5900	1.2487e+03
11	30-Apr-1996	NaN	NaN	394.2700	659.2300	313.8800	605.5200	1.2509e+03
12	31-May-1996	NaN	NaN	353.0500	630.0100	314.8100	622.9700	1.1682e+03
13	30-Jun-1996	NaN	NaN	351.0200	642.3200	328.9200	607.3300	1.2249e+03
14	31-Jul-1996	NaN	NaN	337.0600	606.3100	330.4700	574.9400	1.1736e+03

	Time	ALBI	STEFI	J510	J520	J530	J540	J550
15	31-Aug-1996	NaN	NaN	325.9900	600.9300	329.4100	578.7000	1.1825e+03
16	30-Sep-1996	NaN	NaN	336.9100	625.8400	358.2500	603.0700	1.1899e+03
17	31-Oct-1996	NaN	NaN	344.2300	617.7400	351.4700	601.0800	1.1935e+03
18	30-Nov-1996	NaN	NaN	321.2700	615.9400	333.0500	592.1800	1.1355e+03
19	31-Dec-1996	NaN	NaN	316.4200	611.7500	314.4300	595.8100	1.1415e+03
20	31-Jan-1997	NaN	NaN	325.3700	637.4200	309.8300	609.9600	1.1606e+03
21	28-Feb-1997	NaN	NaN	314.6500	668.9600	307.8600	663.6800	1.2353e+03
22	31-Mar-1997	NaN	NaN	316.5600	686.8300	291.7800	689.2200	1.2723e+03
23	30-Apr-1997	NaN	NaN	333.6900	694.9600	320.6000	664.0300	1.2964e+03
24	31-May-1997	NaN	NaN	312.2800	675.0200	300.2300	633.9000	1.2715e+03
25	30-Jun-1997	NaN	NaN	315.6100	731.2400	329.2900	693.1800	1.3726e+03
26	31-Jul-1997	NaN	NaN	350.3900	760.7900	340.3700	719.1800	1.4725e+03
27	31-Aug-1997	NaN	NaN	333.4400	738.4700	316.0400	670.7800	1.4648e+03
28	30-Sep-1997	NaN	NaN	303.3600	703.4200	311.4700	643.4300	1.4286e+03
29	31-Oct-1997	NaN	NaN	257.7100	634.9900	285.4900	604.4700	1.3775e+03
30	30-Nov-1997	NaN	NaN	244.3000	607.0100	264.4700	576	1.3301e+03
31	31-Dec-1997	NaN	NaN	220.8600	564.3800	258.5200	574.9900	1.2794e+03
32	31-Jan-1998	NaN	NaN	202.7600	580.8400	266.5100	620.9500	1.3500e+03
33	28-Feb-1998	NaN	NaN	217.1500	588.2900	262.1700	641.0800	1.5529e+03
34	31-Mar-1998	NaN	NaN	233.4000	622.5700	319.4200	675.5700	1.6921e+03
35	30-Apr-1998	NaN	NaN	270.2400	713.0900	332.6900	752.2800	1.8093e+03
36	31-May-1998	NaN	NaN	260.7200	663.0500	345.5600	662.7000	1.7371e+03
37	30-Jun-1998	NaN	NaN	204.4200	542.1500	353.9800	558.8000	1.4817e+03
38	31-Jul-1998	NaN	NaN	219.4000	540.0600	389.5600	555.2300	1.4643e+03
39	31-Aug-1998	NaN	NaN	182.6100	388.9500	303.9600	417.9600	976.6300
40	30-Sep-1998	NaN	NaN	194.3700	418.3000	337.9100	409.9000	926.6700
41	31-Oct-1998	NaN	NaN	225.3500	470.6400	342.7200	487.0600	1.0737e+03
42	30-Nov-1998	NaN	NaN	211.1100	466.3700	378.4800	476.6900	1.0221e+03
43	31-Dec-1998	NaN	NaN	195.1900	438.0900	380.3100	484.3400	1.0395e+03
44	31-Jan-1999	NaN	NaN	195.7500	491.5300	465.1200	477.0900	1.1262e+03
45	28-Feb-1999	NaN	NaN	192.5300	503.0500	433.5900	469.5200	1.2080e+03
46	31-Mar-1999	NaN	NaN	220.1700	561.8600	457.5500	540.9000	1.3284e+03
47	30-Apr-1999	NaN	NaN	312.3300	585.5100	472.1100	525.3600	1.3428e+03

	Time	ALBI	STEFI	J510	J520	J530	J540	J550
48	31-May-1999	NaN	NaN	275.4300	545.9600	454.1800	502.9900	1.2300e+03
49	30-Jun-1999	NaN	NaN	304.9600	628.9800	527.6200	543.0400	1.3269e+03
50	31-Jul-1999	NaN	NaN	369.5200	596.6200	528.5900	514.1400	1.2665e+03
51	31-Aug-1999	NaN	NaN	377.2600	572.8400	515.5000	516.4200	1.2083e+03
52	30-Sep-1999	NaN	NaN	380.3800	555.3100	537.8000	519.1200	1.1333e+03
53	31-Oct-1999	NaN	NaN	355.2700	569.1500	536.5300	564.4300	1.2699e+03
54	30-Nov-1999	NaN	NaN	381.3200	629.6500	617.8400	603.6300	1.4272e+03
55	31-Dec-1999	NaN	NaN	418.9600	730.5500	684.5900	674.1000	1.5814e+03
56	31-Jan-2000	NaN	NaN	445.8400	718.6500	683.3800	657.5800	1.5788e+03
57	29-Feb-2000	NaN	94.2640	391.0300	710.1700	659.7600	568.3100	1.5147e+03
58	31-Mar-2000	NaN	95.0910	365.8800	688.0400	731.4300	573.9900	1.4364e+03
59	30-Apr-2000	NaN	95.8809	345.4600	666.1600	737.5100	571.8800	1.3242e+03
60	31-May-2000	NaN	96.6960	352.9700	668.0700	759.3300	515.7400	1.1719e+03
61	30-Jun-2000	NaN	97.4900	357.8700	719.3800	804.3900	566.2500	1.2242e+03
62	31-Jul-2000	NaN	98.3170	378.5500	727.2400	831.3800	565.0200	1.2078e+03
63	31-Aug-2000	NaN	99.1550	402.8100	809.4600	873.6300	580.0500	1.2956e+03
64	30-Sep-2000	NaN	99.9726	373.4300	752.7600	923.8900	566.6700	1.2122e+03
65	31-Oct-2000	NaN	100.8250	353.7600	703.7000	936.9500	529	1.0459e+03
66	30-Nov-2000	NaN	101.6580	341.7500	727.5400	942.2600	545.5600	893.5500
67	31-Dec-2000	NaN	102.5262	381.1300	813.3500	873.4700	624.2600	920.6500
68	31-Jan-2001	NaN	103.4010	415.4200	873.0900	918.2400	648.3700	1.0298e+03
69	28-Feb-2001	NaN	104.1970	444.0600	896.8300	864.1500	669.3400	923
70	31-Mar-2001	NaN	105.0850	443.8800	845.1500	778.8200	611.1100	846.7300
71	30-Apr-2001	NaN	105.9520	497.3000	897.8500	884.9300	651.4900	870.3700
72	31-May-2001	NaN	106.8570	516.0200	901.6300	883.7300	702.2100	891.9900
73	30-Jun-2001	NaN	107.7080	494.4100	971.0800	907.7000	735.1700	954.9400
74	31-Jul-2001	NaN	108.6370	502.4500	937.1800	871.0800	764.7400	952.1400
75	31-Aug-2001	NaN	109.5253	565.9300	964.5600	909	775.2100	979.1800
76	30-Sep-2001	NaN	110.3200	526.6500	817.7900	791.6600	707.0300	861.8600
77	31-Oct-2001	NaN	111.2470	572.6000	844.5200	855.5400	748.0400	888.3900
78	30-Nov-2001	NaN	112.0790	674.5800	955.6400	903.0700	842.6400	950.2200
79	31-Dec-2001	NaN	112.9510	737.0200	916.9500	1.0014e+03	926.5600	897.7800
80	31-Jan-2002	NaN	113.8570	781.1500	874.2400	1.0305e+03	900.8700	844.8300

	Time	ALBI	STEFI	J510	J520	J530	J540	J550
81	28-Feb-2002	NaN	114.7020	882.2700	901.5800	1.0796e+03	906.5800	798.2800
82	31-Mar-2002	NaN	115.6700	888.7100	861.3900	1.1758e+03	936.2300	792.4800
83	30-Apr-2002	NaN	116.6400	866.3200	921.8600	1.0855e+03	1.0131e+03	878.0800
84	31-May-2002	NaN	117.6820	910.3700	980.7300	1.1401e+03	995.8500	980.1100
85	30-Jun-2002	NaN	118.7330	949.8900	956.0500	1.0340e+03	984.2000	952.0400
86	31-Jul-2002	NaN	119.8630	858.8700	877.1300	884.1900	890.2200	921.0600
87	31-Aug-2002	NaN	121.0320	904.1100	903.2700	912.9400	916.3800	929.9000
88	30-Sep-2002	NaN	122.2000	869	897.4100	728.6300	902.2200	939.9600
89	31-Oct-2002	NaN	123.4590	914.8700	938.0700	762.8800	879.0400	964.9700
90	30-Nov-2002	NaN	124.7210	913.4100	994.6000	819.4700	869.0700	1.0592e+03
91	31-Dec-2002	NaN	126.0660	894.3700	981.1900	748.6200	839.8200	1.0461e+03
92	31-Jan-2003	NaN	127.4410	916.4900	917.1200	670.8600	779.9900	1.0240e+03
93	28-Feb-2003	NaN	128.7030	868.6200	886.2000	610.6800	722.9100	986.9000
94	31-Mar-2003	NaN	130.1180	750.3600	852.2100	494.5000	694.5900	941.5600
95	30-Apr-2003	NaN	131.4990	730.7100	871.1400	510.5900	714.1800	963.7200
96	31-May-2003	NaN	132.9390	792.2700	933.9700	611	770.5400	1055
97	30-Jun-2003	NaN	134.3230	760.8700	945.4100	592.4900	752.9800	1.1079e+03
98	31-Jul-2003	NaN	135.6990	787.5200	1.0312e+03	638.3000	796.6700	1.2162e+03
99	31-Aug-2003	173.7190	137.0220	833.0100	1.0272e+03	670.2000	839.0400	1.2417e+03
100	30-Sep-2003	NaN	138.2380	822.6700	1.0075e+03	675.9200	837.8300	1.2192e+03

⋮

```
subplot(1,6,4);
spy(isnan(table2array(allDataTable)));
```

We can try remove all the missing data in the early part of the dataset.

Note: Take care not to include data prior to this.

```
[countNans,idx] = max(sum(isnan(allDataTable{:,:}),1));
rmmissingProxy =  allDataTable.Properties.VariableNames{idx};
allDataTable0 = rmmissing(allDataTable,"DataVariables",rmmissingProxy);
spy(isnan(table2array(allDataTable0)));
```

Notice that we now had non-uniformly sampled data:

```
allDataTable0(1:3,:)
```

```
ans = 3×11 timetable
```

...

	Time	ALBI	STEFI	J510	J520	J530	J540	J550
1	31-Aug-2003	173.7190	137.0220	833.0100	1.0272e+03	670.2000	839.0400	1.2417e+03
2	30-Nov-2004	205.7050	151.8650	1.1884e+03	1.8525e+03	913.2900	1.4354e+03	2402
3	31-Dec-2004	212	152.8080	1.2370e+03	1.9385e+03	968.1500	1.4279e+03	2.5499e+03

We can check for uniformity in the data ranges to show that this is only in the first row of data

```
find(diff(datenum(allDataTable0.Time))>31)
```

```
ans = 1
```

We want to remove all the data upto the maximum NaN count and we will do this by reference

```
idx = isnan(allDataTable{:,:});
allDataTable = allDataTable(max(find(idx,max(max(cumsum(idx)))))+1:end,:);
allDataTable(1:3,:)
```

```
ans = 3×11 timetable
```

...

	Time	ALBI	STEFI	J510	J520	J530	J540	J550
1	30-Nov-2004	205.7050	151.8650	1.1884e+03	1.8525e+03	913.2900	1.4354e+03	2402
2	31-Dec-2004	212	152.8080	1.2370e+03	1.9385e+03	968.1500	1.4279e+03	2.5499e+03
3	31-Jan-2005	215.0800	NaN	1.1900e+03	1910	965.7200	1.4187e+03	2.5414e+03

```
find(diff(datenum(allDataTable.Time))>31)
```

```
ans =

  0×1 empty double column vector
```

```
subplot(1,6,5);
spy(isnan(table2array(allDataTable)));
```

There is still sequential missing data at the end of the data set. We should remove this too.

```
countNans = max(max(cumsum(isnan(allDataTable{:,:}))));
allDataTable = allDataTable(1:end-countNans+1,:);
```

We will then fill in the single missing STEFI missing data point.

```
allDataTableFilled = fillmissing(allDataTable,'previous')
```

```
allDataTableFilled = 146×11 timetable
```

...

	Time	ALBI	STEFI	J510	J520	J530	J540	J550
1	30-Nov-2004	205.7050	151.8650	1.1884e+03	1.8525e+03	913.2900	1.4354e+03	2402
2	31-Dec-2004	212	152.8080	1.2370e+03	1.9385e+03	968.1500	1.4279e+03	2.5499e+03
3	31-Jan-2005	215.0800	152.8080	1.1900e+03	1910	965.7200	1.4187e+03	2.5414e+03

	Time	ALBI	STEFI	J510	J520	J530	J540	J550
4	28-Feb-2005	219.3550	154.6010	1.2389e+03	1.9594e+03	959.2800	1.4677e+03	2.5543e+03
5	31-Mar-2005	211.2980	155.5450	1.2033e+03	1.8547e+03	1.0052e+03	1.4874e+03	2.4252e+03
6	30-Apr-2005	215.6110	156.4603	1.0833e+03	1.7659e+03	944.2000	1.4008e+03	2.3822e+03
7	31-May-2005	215.4030	157.3830	1.1287e+03	1.8156e+03	1.0528e+03	1.5858e+03	2.4843e+03
8	30-Jun-2005	221.1870	158.2690	1.1632e+03	1.9079e+03	1.1558e+03	1.6067e+03	2.5618e+03
9	31-Jul-2005	223.3150	159.1759	1.2259e+03	2.0750e+03	1.2085e+03	1.7843e+03	2.8796e+03
10	31-Aug-2005	223.4430	160.0890	1.2202e+03	2.1389e+03	1.2502e+03	1.7797e+03	2.9619e+03
11	30-Sep-2005	223.6150	160.9700	1.3738e+03	2.3621e+03	1.3542e+03	1.9527e+03	3.1025e+03
12	31-Oct-2005	225.0240	161.8840	1.3305e+03	2.2540e+03	1.3478e+03	1.9491e+03	2.9561e+03
13	30-Nov-2005	230.3570	162.7810	1.3934e+03	2.2638e+03	1.3399e+03	1.8691e+03	3.0162e+03
14	31-Dec-2005	234.8890	163.7173	1.4614e+03	2.4506e+03	1.4474e+03	1.9402e+03	3.2973e+03
15	31-Jan-2006	237.1490	164.6610	1.6408e+03	2.6834e+03	1.5155e+03	2.2443e+03	3.6351e+03
16	28-Feb-2006	238.9940	165.5200	1.5401e+03	2.7828e+03	1.4765e+03	2.2110e+03	3.6641e+03
17	31-Mar-2006	238.4470	166.4760	1.6758e+03	3.0321e+03	1.5590e+03	2.4178e+03	3.8338e+03
18	30-Apr-2006	241.2120	167.4060	1.8013e+03	3.0318e+03	1.6473e+03	2.5434e+03	3.9275e+03
19	31-May-2006	238.5130	168.3750	1.8365e+03	2.7242e+03	1.6000e+03	2.2696e+03	3.5637e+03
20	30-Jun-2006	229.9050	169.3230	2.0223e+03	2.7113e+03	1.6347e+03	2.2949e+03	3.3095e+03
21	31-Jul-2006	231.4790	170.3290	1.9414e+03	2.7306e+03	1.6789e+03	2.2909e+03	3.3346e+03
22	31-Aug-2006	231.5700	171.3680	2.0617e+03	2.9629e+03	1.7735e+03	2.5091e+03	3.4693e+03
23	30-Sep-2006	234.7170	172.4123	2.0654e+03	2.9758e+03	1.8713e+03	2.6221e+03	3.4857e+03
24	31-Oct-2006	242.0080	173.5310	2.1482e+03	3.2839e+03	1.8758e+03	2.6156e+03	3.8818e+03
25	30-Nov-2006	244.0790	174.6580	2.1806e+03	3.4014e+03	1.9439e+03	2.7531e+03	4.1405e+03
26	31-Dec-2006	247.7530	175.8610	2.1787e+03	3.6577e+03	2.0868e+03	2.8044e+03	4.4185e+03
27	31-Jan-2007	249.4600	177.0830	2.2067e+03	3.8483e+03	2.0997e+03	2.9979e+03	4.7445e+03
28	28-Feb-2007	253.0070	178.2150	2.2897e+03	3.8645e+03	2.0755e+03	2.9458e+03	4.8181e+03
29	31-Mar-2007	251.7720	179.4910	2.5528e+03	3.9990e+03	2.0720e+03	2.9327e+03	4.9844e+03
30	30-Apr-2007	255.6270	180.7590	2.5489e+03	4.2206e+03	2.2150e+03	3.1030e+03	5.3221e+03
31	31-May-2007	252.8060	182.0850	2.7175e+03	4.2089e+03	2.2263e+03	3.1832e+03	5.2868e+03
32	30-Jun-2007	247.6080	183.3934	2.7277e+03	4.2220e+03	2.2143e+03	3.0021e+03	5.0139e+03
33	31-Jul-2007	248.4450	184.7920	2.7419e+03	4.2225e+03	2.2957e+03	2.6715e+03	4.9259e+03
34	31-Aug-2007	250.0620	186.2270	2.7133e+03	4.2043e+03	2.3693e+03	2.7314e+03	4.9517e+03
35	30-Sep-2007	255.9130	187.6565	3.0758e+03	4.2733e+03	2.3786e+03	2.6304e+03	4.9714e+03
36	31-Oct-2007	260.0130	189.1710	3.0799e+03	4.5063e+03	2.4173e+03	2.8791e+03	5.2732e+03

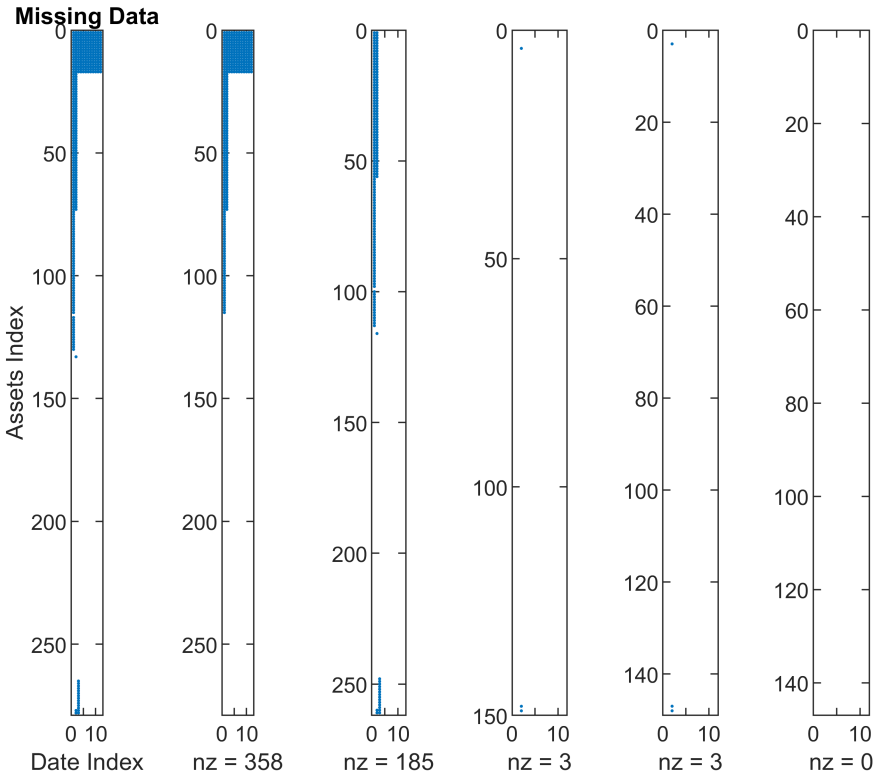
	Time	ALBI	STEFI	J510	J520	J530	J540	J550
37	30-Nov-2007	255.9810	190.6850	2.9679e+03	4.3021e+03	2.4224e+03	2.6807e+03	4.7960e+03
38	31-Dec-2007	258.1780	192.3010	2.7788e+03	4.3247e+03	2.3894e+03	2.7239e+03	4.6964e+03
39	31-Jan-2008	256.8120	193.9800	2.8487e+03	3.6846e+03	2.0809e+03	2.3425e+03	3.9295e+03
40	29-Feb-2008	254.6700	195.5810	3.3769e+03	4.1106e+03	2.1975e+03	2.3068e+03	4.2180e+03
41	31-Mar-2008	253.3310	197.3270	3.2739e+03	3.9778e+03	2.2551e+03	2.1880e+03	4.0014e+03
42	30-Apr-2008	251.4540	199.0540	3.4136e+03	3.9829e+03	2.2812e+03	2.1792e+03	4.2368e+03
43	31-May-2008	245.1690	200.8907	3.6250e+03	3.9665e+03	2.4446e+03	2.1504e+03	4.1756e+03
44	30-Jun-2008	240.9310	202.7250	3.6698e+03	3.5846e+03	2.1935e+03	2.0122e+03	3.9388e+03
45	31-Jul-2008	261.4270	204.6800	2.9590e+03	3.6601e+03	2.1165e+03	2.2373e+03	4.3054e+03
46	31-Aug-2008	264.7790	206.6795	2.8496e+03	3.8388e+03	2.2342e+03	2.4593e+03	4.6570e+03
47	30-Sep-2008	271.2230	208.6420	2.2296e+03	3.6902e+03	2.0220e+03	2.3016e+03	4.3204e+03
48	31-Oct-2008	269.9930	211	1.8222e+03	3.4168e+03	2.1151e+03	2.0086e+03	4.2428e+03
49	30-Nov-2008	282.4280	213	1.9204e+03	3.0381e+03	2.1808e+03	2.1897e+03	3.9092e+03
50	31-Dec-2008	301.9990	215	1.9288e+03	3.1923e+03	2.2494e+03	2.2606e+03	4.2100e+03
51	31-Jan-2009	294.8220	217	1.8685e+03	2.9844e+03	2.1582e+03	2.5625e+03	4.1216e+03
52	28-Feb-2009	286.3800	219	1.6893e+03	2.6678e+03	1.9323e+03	2516	3.7873e+03
53	31-Mar-2009	286.4840	221	1.9420e+03	2.7846e+03	1.9537e+03	2.5397e+03	3.9187e+03
54	30-Apr-2009	289.2230	222	1.9026e+03	3.0293e+03	2.0022e+03	2.6644e+03	4.1327e+03
55	31-May-2009	287.9890	224	2.2106e+03	3.1274e+03	2.2564e+03	2.7970e+03	4.4314e+03
56	30-Jun-2009	287.3190	226	2.0221e+03	3.1782e+03	2.1721e+03	2.9434e+03	4.5953e+03
57	31-Jul-2009	291.0320	227	2.2374e+03	3.4043e+03	2.5021e+03	3.2220e+03	5.0545e+03
58	31-Aug-2009	295.5940	228.7301	2.2797e+03	3.7012e+03	2.5858e+03	3.2073e+03	5.2721e+03
59	30-Sep-2009	295.8260	230.1487	2.2770e+03	3.8689e+03	2.6038e+03	3.2980e+03	5.5559e+03
60	31-Oct-2009	295.1850	231.5433	2.4398e+03	3.9232e+03	2.8993e+03	3.5946e+03	5.9167e+03
61	30-Nov-2009	295.4780	232.9687	2.5981e+03	3.7283e+03	3.0103e+03	3.6421e+03	5.7139e+03
62	31-Dec-2009	299.0130	234.3978	2.6651e+03	3.8835e+03	3.0913e+03	3.9961e+03	6.1025e+03
63	31-Jan-2010	299.8170	235.7405	2.5022e+03	3.7844e+03	3.0664e+03	3.7849e+03	5.8477e+03
64	28-Feb-2010	305.8650	237.0374	2.4909e+03	3.8350e+03	2.9680e+03	3.9097e+03	6.2339e+03
65	31-Mar-2010	312.3060	238.5726	2.7338e+03	4.1231e+03	3.2115e+03	4.1991e+03	6.6886e+03
66	30-Apr-2010	316.2460	239.9547	2.6833e+03	4.1285e+03	3.3177e+03	4.3176e+03	6.6937e+03
67	31-May-2010	314.9270	241.3610	2.5117e+03	3.9968e+03	3.0727e+03	4.1686e+03	6.7081e+03
68	30-Jun-2010	315.7910	242.7089	2.4034e+03	3.8334e+03	3.1329e+03	4.0967e+03	6.4484e+03
69	31-Jul-2010	328.8100	244.0478	2.5609e+03	4.0601e+03	3.2807e+03	4.3283e+03	7.2305e+03

	Time	ALBI	STEFI	J510	J520	J530	J540	J550
70	31-Aug-2010	338.6320	245.4669	2.3997e+03	4.0742e+03	3.2019e+03	4.3443e+03	7.0461e+03
71	30-Sep-2010	341.1960	246.7810	2.5712e+03	4.4423e+03	3.5615e+03	4.6831e+03	8.0489e+03
72	31-Oct-2010	344.7510	248.0240	2.7962e+03	4.5588e+03	3.6041e+03	4.7469e+03	8.3727e+03
73	30-Nov-2010	337.8870	249.3703	2.7776e+03	4.5598e+03	3.7467e+03	4.7870e+03	8.3134e+03
74	31-Dec-2010	343.7440	250.6353	2.9776e+03	4.8162e+03	3.8927e+03	4.8379e+03	8.7135e+03
75	31-Jan-2011	336.4010	251.8723	2.9734e+03	4.5559e+03	3.8426e+03	4.6329e+03	8.1002e+03
76	28-Feb-2011	336.7030	252.9735	3.1119e+03	4.4805e+03	3.9071e+03	4.6174e+03	8.4730e+03
77	31-Mar-2011	338.3500	254.1845	3.0418e+03	4.4012e+03	3.9564e+03	4.5710e+03	8.4060e+03
78	30-Apr-2011	345.9980	255.3145	3.0585e+03	4.5046e+03	4.1168e+03	4.6771e+03	8.8494e+03
79	31-May-2011	350.9690	256.5621	2.9847e+03	4.5311e+03	4.2118e+03	4.8494e+03	8.7677e+03
80	30-Jun-2011	351.5260	257.7353	2.8944e+03	4.5195e+03	4.1242e+03	4.8881e+03	8.6464e+03
81	31-Jul-2011	356.6690	258.8732	2.7782e+03	4.4873e+03	4.1716e+03	4.8598e+03	8.5429e+03
82	31-Aug-2011	369.1210	260.1723	2.7267e+03	4.4074e+03	4.1165e+03	4.8136e+03	8.8601e+03
83	30-Sep-2011	361.3910	261.3548	2.5808e+03	4.3113e+03	4.0374e+03	5.0099e+03	8.5394e+03
84	31-Oct-2011	371.3330	262.5788	2.8616e+03	4.5754e+03	4.6686e+03	5.2069e+03	9.2959e+03
85	30-Nov-2011	371.3830	263.7658	2.8943e+03	4.6579e+03	4.6297e+03	5.2286e+03	9.5841e+03
86	31-Dec-2011	374.0700	264.9581	2.7301e+03	4.6510e+03	4.4634e+03	5.2637e+03	9.4595e+03
87	31-Jan-2012	381.7820	266.2382	2.9675e+03	5.0282e+03	4.6847e+03	5.4439e+03	9.8836e+03
88	29-Feb-2012	382.4520	267.4030	2.9118e+03	5.2996e+03	4.8435e+03	5.6491e+03	1.0330e+04
89	31-Mar-2012	382.9110	268.6154	2.6762e+03	5.4971e+03	4.9486e+03	6.0280e+03	1.0598e+04
90	30-Apr-2012	389.8360	269.8765	2.7478e+03	5.5799e+03	5.1355e+03	6.3066e+03	1.1117e+04
91	31-May-2012	389.7830	271.1448	2.5423e+03	5.3987e+03	5.0318e+03	6.2797e+03	1.0846e+04
92	30-Jun-2012	402.8020	272.3362	2.5988e+03	5.4741e+03	5.0448e+03	6.6829e+03	1.1194e+04
93	31-Jul-2012	418.7450	273.6514	2.5355e+03	5.7757e+03	5.3411e+03	7.2664e+03	1.1862e+04
94	31-Aug-2012	419.0430	274.9015	2.4915e+03	5.7888e+03	5.7389e+03	7.2644e+03	1.2438e+04
95	30-Sep-2012	422.9590	276.0126	2.6464e+03	5.8369e+03	5.5672e+03	7.2970e+03	1.2692e+04
96	31-Oct-2012	420.4150	277.3019	2.8223e+03	5.9540e+03	5.8758e+03	7802	1.3399e+04
97	30-Nov-2012	424.1300	278.4666	2.7642e+03	5.9602e+03	6.5284e+03	7.7129e+03	1.3608e+04
98	31-Dec-2012	433.8990	279.6686	2.8766e+03	6.3502e+03	6.3799e+03	8.3971e+03	1.4088e+04
99	31-Jan-2013	434.1860	280.8690	2.9699e+03	6.4584e+03	7.0024e+03	8.2872e+03	1.3461e+04
100	28-Feb-2013	437.0600	281.9545	2.7684e+03	6.7963e+03	6.9615e+03	8.4143e+03	1.3344e+04

⋮

```
subplot(1,6,6);
```

```
spy(isnan(table2array(allDataTableFilled)));
```




```
find(diff(datenum(allDataTable.Time))>31)
```

```
ans =

  0×1 empty double column vector
```

We are keeping the data with this single missing point because this represent a case when measurements were not made but the index exists. This is different from the situations where the indices did not exist or were discontinued.

Save your pre-processed data

The workspace is not maintained across sessions of MATLAB. When you quit MATLAB, the workspace clears. However, you can save any or all the variables in the current workspace to a MAT-file (`.mat`). You can then reuse the workspace variables later during the current MATLAB session or during another session by loading the saved MAT-file.

There are several ways to save workspace variables interactively:

- To save all workspace variables to a MAT-file, on the **Home** tab, in the **Variable** section, click **Save Workspace**.
- To save a subset of your workspace variables to a MAT-file, select the variables in the Workspace browser, right-click, and then select **Save As**. You also can drag the selected variables from the Workspace browser to the Current Folder browser.

- To save variables to a MATLAB script, click the **Save Workspace** button or select the **Save As** option, and in the **Save As** window, set the **Save as type** option to **MATLAB Script**. Variables that cannot be saved to a script are saved to a MAT-file with the same name as that of the script.

You also can save workspace variables programmatically using the `save` function.

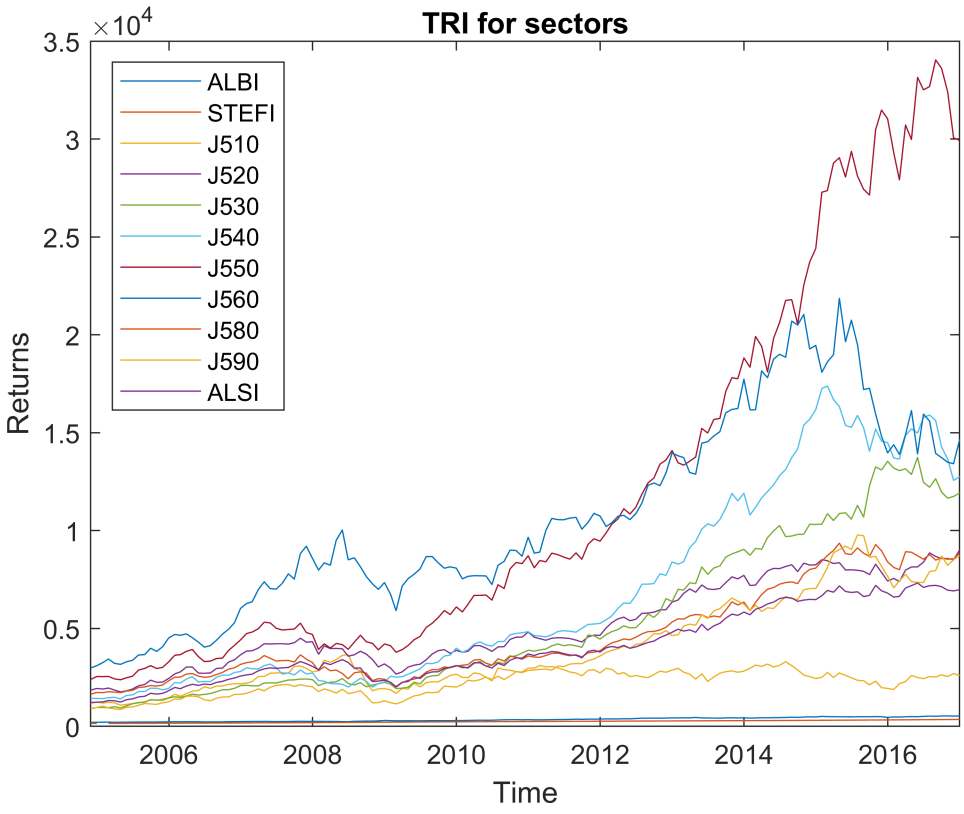
```
save("PortfolioTheory.mat",'allDataTableFilled')
```

Read more about saving options here.

Visualisation

We can now plot the data (unfilled) as times series plots

```
% Visualise the data on a single plot
%plot the timeseries
figure;
plot(allDataTable.Time,allDataTable{:,:})
ylabel("Returns")
xlabel("Time")
title("TRI for sectors")
%include the legend
legend(allDataTable.Properties.VariableNames,Location="northwest")
```



Compute returns

Compute returns Simple returns (rather than Continous returns)

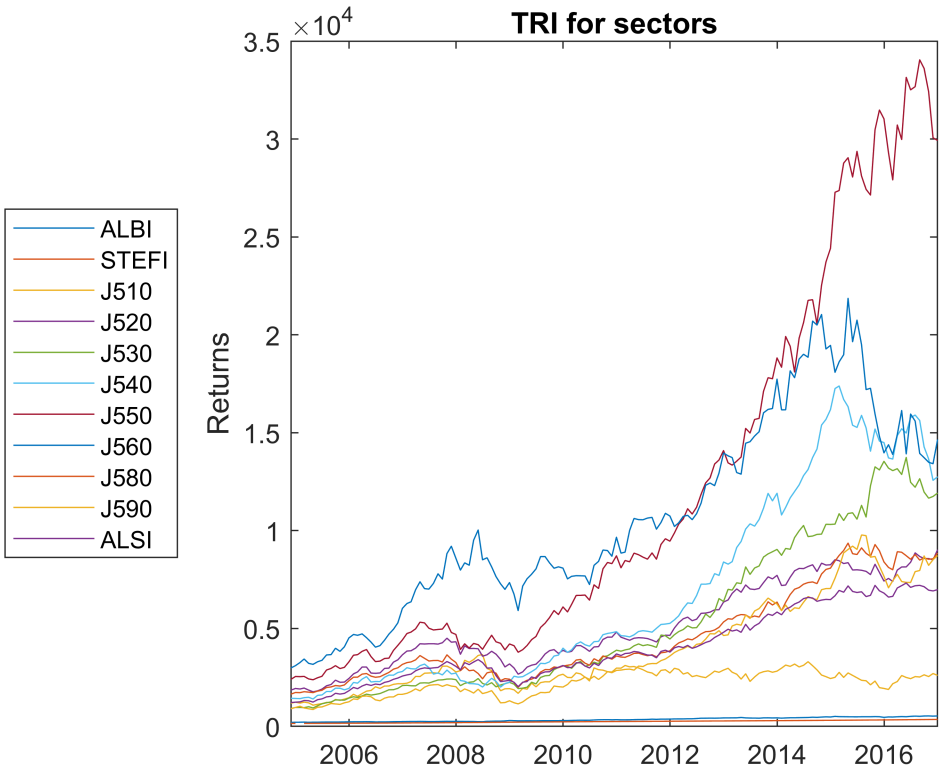
$$R_t = \frac{P_t - P_{t-1}}{P_{t-1}}$$

```
% allDataReturnsTable = tick2ret(allDataTable,'Method','continuous');
allDataReturnsTable = tick2ret(allDataTable,'Method','Simple');
```

Visualise the Index and Return data

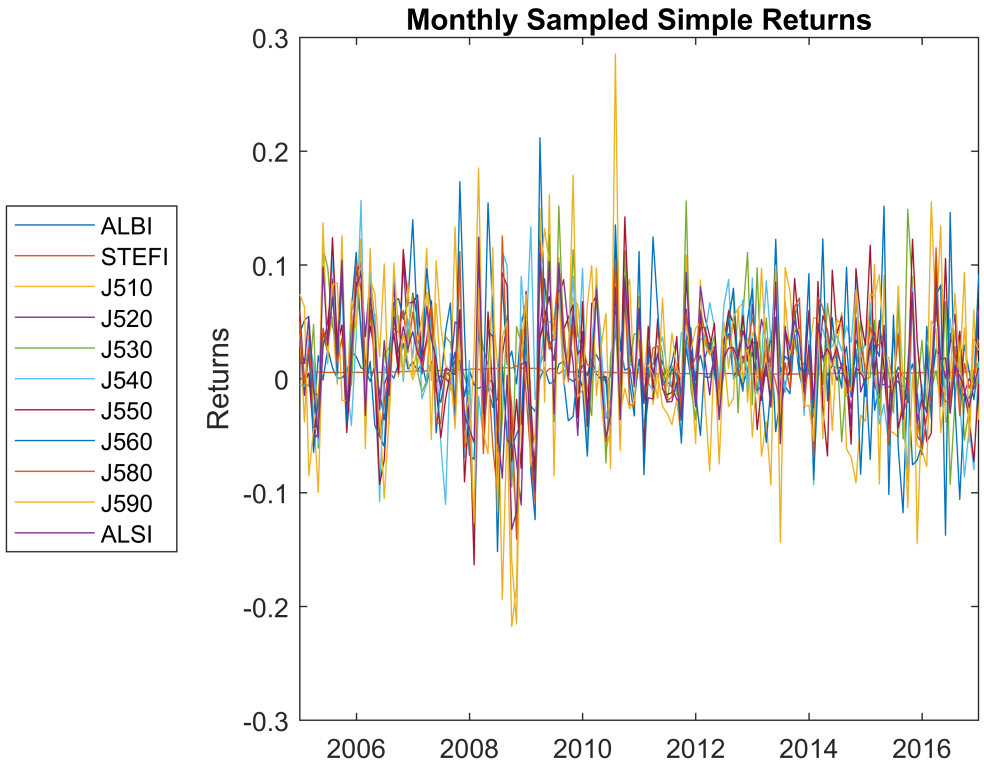
Plot two plots on the same figure

```
% Top plot
%plot the timeseries
plot(allDataTable.Time,allDataTable{:,:})
ylabel("Returns")
title("TRI for sectors")
%include the legend
legend(allDataTable.Properties.VariableNames,Location="westoutside")
% Bottom plot
nexttile
```




```
%plot the timeseries
plot(allDataReturnsTable.Time,allDataReturnsTable{:,:})
ylabel("Returns")
```

```
title("Monthly Sampled Simple Returns")
%include the legend
legend(allDataTable.Properties.VariableNames,Location="westoutside")
```



Manage Missing Data

Checking for missing data

```
any(isnan(allDataReturnsTable{:,:}))
```

```
ans = 1×11 logical array
   0   1   0   0   0   0   0   0   0   0   0
```

Compute the Arithmetic means without correcting for missing data (NAN)

```
portfolioMean = mean(allDataReturnsTable{:,:});
portfolioStandardDev = std(allDataReturnsTable{:,:});
```

If there are any missing data points exclude these from the estimation

```
portfolioMean = mean(allDataReturnsTable{:,:},'omitnan');
portfolioStdDev = std(allDataReturnsTable{:,:},1,'omitnan');
portfolioVariance = var(allDataReturnsTable{:,:},'omitnan');
```

We either remove all the rows with missing data or interpolate using zero-order hold (so pad with zero's for returns and with the previous price for prices

```
allDataTable = fillmissing(allDataTable,'previous')
```

allDataTable = 146×11 timetable

...

	Time	ALBI	STEFI	J510	J520	J530	J540	J550
1	30-Nov-2004	205.7050	151.8650	1.1884e+03	1.8525e+03	913.2900	1.4354e+03	2402
2	31-Dec-2004	212	152.8080	1.2370e+03	1.9385e+03	968.1500	1.4279e+03	2.5499e+03
3	31-Jan-2005	215.0800	152.8080	1.1900e+03	1910	965.7200	1.4187e+03	2.5414e+03
4	28-Feb-2005	219.3550	154.6010	1.2389e+03	1.9594e+03	959.2800	1.4677e+03	2.5543e+03
5	31-Mar-2005	211.2980	155.5450	1.2033e+03	1.8547e+03	1.0052e+03	1.4874e+03	2.4252e+03
6	30-Apr-2005	215.6110	156.4603	1.0833e+03	1.7659e+03	944.2000	1.4008e+03	2.3822e+03
7	31-May-2005	215.4030	157.3830	1.1287e+03	1.8156e+03	1.0528e+03	1.5858e+03	2.4843e+03
8	30-Jun-2005	221.1870	158.2690	1.1632e+03	1.9079e+03	1.1558e+03	1.6067e+03	2.5618e+03
9	31-Jul-2005	223.3150	159.1759	1.2259e+03	2.0750e+03	1.2085e+03	1.7843e+03	2.8796e+03
10	31-Aug-2005	223.4430	160.0890	1.2202e+03	2.1389e+03	1.2502e+03	1.7797e+03	2.9619e+03
11	30-Sep-2005	223.6150	160.9700	1.3738e+03	2.3621e+03	1.3542e+03	1.9527e+03	3.1025e+03
12	31-Oct-2005	225.0240	161.8840	1.3305e+03	2.2540e+03	1.3478e+03	1.9491e+03	2.9561e+03
13	30-Nov-2005	230.3570	162.7810	1.3934e+03	2.2638e+03	1.3399e+03	1.8691e+03	3.0162e+03
14	31-Dec-2005	234.8890	163.7173	1.4614e+03	2.4506e+03	1.4474e+03	1.9402e+03	3.2973e+03
15	31-Jan-2006	237.1490	164.6610	1.6408e+03	2.6834e+03	1.5155e+03	2.2443e+03	3.6351e+03
16	28-Feb-2006	238.9940	165.5200	1.5401e+03	2.7828e+03	1.4765e+03	2.2110e+03	3.6641e+03
17	31-Mar-2006	238.4470	166.4760	1.6758e+03	3.0321e+03	1.5590e+03	2.4178e+03	3.8338e+03
18	30-Apr-2006	241.2120	167.4060	1.8013e+03	3.0318e+03	1.6473e+03	2.5434e+03	3.9275e+03
19	31-May-2006	238.5130	168.3750	1.8365e+03	2.7242e+03	1.6000e+03	2.2696e+03	3.5637e+03
20	30-Jun-2006	229.9050	169.3230	2.0223e+03	2.7113e+03	1.6347e+03	2.2949e+03	3.3095e+03
21	31-Jul-2006	231.4790	170.3290	1.9414e+03	2.7306e+03	1.6789e+03	2.2909e+03	3.3346e+03
22	31-Aug-2006	231.5700	171.3680	2.0617e+03	2.9629e+03	1.7735e+03	2.5091e+03	3.4693e+03
23	30-Sep-2006	234.7170	172.4123	2.0654e+03	2.9758e+03	1.8713e+03	2.6221e+03	3.4857e+03
24	31-Oct-2006	242.0080	173.5310	2.1482e+03	3.2839e+03	1.8758e+03	2.6156e+03	3.8818e+03
25	30-Nov-2006	244.0790	174.6580	2.1806e+03	3.4014e+03	1.9439e+03	2.7531e+03	4.1405e+03
26	31-Dec-2006	247.7530	175.8610	2.1787e+03	3.6577e+03	2.0868e+03	2.8044e+03	4.4185e+03
27	31-Jan-2007	249.4600	177.0830	2.2067e+03	3.8483e+03	2.0997e+03	2.9979e+03	4.7445e+03
28	28-Feb-2007	253.0070	178.2150	2.2897e+03	3.8645e+03	2.0755e+03	2.9458e+03	4.8181e+03
29	31-Mar-2007	251.7720	179.4910	2.5528e+03	3.9990e+03	2.0720e+03	2.9327e+03	4.9844e+03
30	30-Apr-2007	255.6270	180.7590	2.5489e+03	4.2206e+03	2.2150e+03	3.1030e+03	5.3221e+03
31	31-May-2007	252.8060	182.0850	2.7175e+03	4.2089e+03	2.2263e+03	3.1832e+03	5.2868e+03

	Time	ALBI	STEFI	J510	J520	J530	J540	J550
32	30-Jun-2007	247.6080	183.3934	2.7277e+03	4.2220e+03	2.2143e+03	3.0021e+03	5.0139e+03
33	31-Jul-2007	248.4450	184.7920	2.7419e+03	4.2225e+03	2.2957e+03	2.6715e+03	4.9259e+03
34	31-Aug-2007	250.0620	186.2270	2.7133e+03	4.2043e+03	2.3693e+03	2.7314e+03	4.9517e+03
35	30-Sep-2007	255.9130	187.6565	3.0758e+03	4.2733e+03	2.3786e+03	2.6304e+03	4.9714e+03
36	31-Oct-2007	260.0130	189.1710	3.0799e+03	4.5063e+03	2.4173e+03	2.8791e+03	5.2732e+03
37	30-Nov-2007	255.9810	190.6850	2.9679e+03	4.3021e+03	2.4224e+03	2.6807e+03	4.7960e+03
38	31-Dec-2007	258.1780	192.3010	2.7788e+03	4.3247e+03	2.3894e+03	2.7239e+03	4.6964e+03
39	31-Jan-2008	256.8120	193.9800	2.8487e+03	3.6846e+03	2.0809e+03	2.3425e+03	3.9295e+03
40	29-Feb-2008	254.6700	195.5810	3.3769e+03	4.1106e+03	2.1975e+03	2.3068e+03	4.2180e+03
41	31-Mar-2008	253.3310	197.3270	3.2739e+03	3.9778e+03	2.2551e+03	2.1880e+03	4.0014e+03
42	30-Apr-2008	251.4540	199.0540	3.4136e+03	3.9829e+03	2.2812e+03	2.1792e+03	4.2368e+03
43	31-May-2008	245.1690	200.8907	3.6250e+03	3.9665e+03	2.4446e+03	2.1504e+03	4.1756e+03
44	30-Jun-2008	240.9310	202.7250	3.6698e+03	3.5846e+03	2.1935e+03	2.0122e+03	3.9388e+03
45	31-Jul-2008	261.4270	204.6800	2.9590e+03	3.6601e+03	2.1165e+03	2.2373e+03	4.3054e+03
46	31-Aug-2008	264.7790	206.6795	2.8496e+03	3.8388e+03	2.2342e+03	2.4593e+03	4.6570e+03
47	30-Sep-2008	271.2230	208.6420	2.2296e+03	3.6902e+03	2.0220e+03	2.3016e+03	4.3204e+03
48	31-Oct-2008	269.9930	211	1.8222e+03	3.4168e+03	2.1151e+03	2.0086e+03	4.2428e+03
49	30-Nov-2008	282.4280	213	1.9204e+03	3.0381e+03	2.1808e+03	2.1897e+03	3.9092e+03
50	31-Dec-2008	301.9990	215	1.9288e+03	3.1923e+03	2.2494e+03	2.2606e+03	4.2100e+03
51	31-Jan-2009	294.8220	217	1.8685e+03	2.9844e+03	2.1582e+03	2.5625e+03	4.1216e+03
52	28-Feb-2009	286.3800	219	1.6893e+03	2.6678e+03	1.9323e+03	2516	3.7873e+03
53	31-Mar-2009	286.4840	221	1.9420e+03	2.7846e+03	1.9537e+03	2.5397e+03	3.9187e+03
54	30-Apr-2009	289.2230	222	1.9026e+03	3.0293e+03	2.0022e+03	2.6644e+03	4.1327e+03
55	31-May-2009	287.9890	224	2.2106e+03	3.1274e+03	2.2564e+03	2.7970e+03	4.4314e+03
56	30-Jun-2009	287.3190	226	2.0221e+03	3.1782e+03	2.1721e+03	2.9434e+03	4.5953e+03
57	31-Jul-2009	291.0320	227	2.2374e+03	3.4043e+03	2.5021e+03	3.2220e+03	5.0545e+03
58	31-Aug-2009	295.5940	228.7301	2.2797e+03	3.7012e+03	2.5858e+03	3.2073e+03	5.2721e+03
59	30-Sep-2009	295.8260	230.1487	2.2770e+03	3.8689e+03	2.6038e+03	3.2980e+03	5.5559e+03
60	31-Oct-2009	295.1850	231.5433	2.4398e+03	3.9232e+03	2.8993e+03	3.5946e+03	5.9167e+03
61	30-Nov-2009	295.4780	232.9687	2.5981e+03	3.7283e+03	3.0103e+03	3.6421e+03	5.7139e+03
62	31-Dec-2009	299.0130	234.3978	2.6651e+03	3.8835e+03	3.0913e+03	3.9961e+03	6.1025e+03
63	31-Jan-2010	299.8170	235.7405	2.5022e+03	3.7844e+03	3.0664e+03	3.7849e+03	5.8477e+03
64	28-Feb-2010	305.8650	237.0374	2.4909e+03	3.8350e+03	2.9680e+03	3.9097e+03	6.2339e+03

	Time	ALBI	STEFI	J510	J520	J530	J540	J550
65	31-Mar-2010	312.3060	238.5726	2.7338e+03	4.1231e+03	3.2115e+03	4.1991e+03	6.6886e+03
66	30-Apr-2010	316.2460	239.9547	2.6833e+03	4.1285e+03	3.3177e+03	4.3176e+03	6.6937e+03
67	31-May-2010	314.9270	241.3610	2.5117e+03	3.9968e+03	3.0727e+03	4.1686e+03	6.7081e+03
68	30-Jun-2010	315.7910	242.7089	2.4034e+03	3.8334e+03	3.1329e+03	4.0967e+03	6.4484e+03
69	31-Jul-2010	328.8100	244.0478	2.5609e+03	4.0601e+03	3.2807e+03	4.3283e+03	7.2305e+03
70	31-Aug-2010	338.6320	245.4669	2.3997e+03	4.0742e+03	3.2019e+03	4.3443e+03	7.0461e+03
71	30-Sep-2010	341.1960	246.7810	2.5712e+03	4.4423e+03	3.5615e+03	4.6831e+03	8.0489e+03
72	31-Oct-2010	344.7510	248.0240	2.7962e+03	4.5588e+03	3.6041e+03	4.7469e+03	8.3727e+03
73	30-Nov-2010	337.8870	249.3703	2.7776e+03	4.5598e+03	3.7467e+03	4.7870e+03	8.3134e+03
74	31-Dec-2010	343.7440	250.6353	2.9776e+03	4.8162e+03	3.8927e+03	4.8379e+03	8.7135e+03
75	31-Jan-2011	336.4010	251.8723	2.9734e+03	4.5559e+03	3.8426e+03	4.6329e+03	8.1002e+03
76	28-Feb-2011	336.7030	252.9735	3.1119e+03	4.4805e+03	3.9071e+03	4.6174e+03	8.4730e+03
77	31-Mar-2011	338.3500	254.1845	3.0418e+03	4.4012e+03	3.9564e+03	4.5710e+03	8.4060e+03
78	30-Apr-2011	345.9980	255.3145	3.0585e+03	4.5046e+03	4.1168e+03	4.6771e+03	8.8494e+03
79	31-May-2011	350.9690	256.5621	2.9847e+03	4.5311e+03	4.2118e+03	4.8494e+03	8.7677e+03
80	30-Jun-2011	351.5260	257.7353	2.8944e+03	4.5195e+03	4.1242e+03	4.8881e+03	8.6464e+03
81	31-Jul-2011	356.6690	258.8732	2.7782e+03	4.4873e+03	4.1716e+03	4.8598e+03	8.5429e+03
82	31-Aug-2011	369.1210	260.1723	2.7267e+03	4.4074e+03	4.1165e+03	4.8136e+03	8.8601e+03
83	30-Sep-2011	361.3910	261.3548	2.5808e+03	4.3113e+03	4.0374e+03	5.0099e+03	8.5394e+03
84	31-Oct-2011	371.3330	262.5788	2.8616e+03	4.5754e+03	4.6686e+03	5.2069e+03	9.2959e+03
85	30-Nov-2011	371.3830	263.7658	2.8943e+03	4.6579e+03	4.6297e+03	5.2286e+03	9.5841e+03
86	31-Dec-2011	374.0700	264.9581	2.7301e+03	4.6510e+03	4.4634e+03	5.2637e+03	9.4595e+03
87	31-Jan-2012	381.7820	266.2382	2.9675e+03	5.0282e+03	4.6847e+03	5.4439e+03	9.8836e+03
88	29-Feb-2012	382.4520	267.4030	2.9118e+03	5.2996e+03	4.8435e+03	5.6491e+03	1.0330e+04
89	31-Mar-2012	382.9110	268.6154	2.6762e+03	5.4971e+03	4.9486e+03	6.0280e+03	1.0598e+04
90	30-Apr-2012	389.8360	269.8765	2.7478e+03	5.5799e+03	5.1355e+03	6.3066e+03	1.1117e+04
91	31-May-2012	389.7830	271.1448	2.5423e+03	5.3987e+03	5.0318e+03	6.2797e+03	1.0846e+04
92	30-Jun-2012	402.8020	272.3362	2.5988e+03	5.4741e+03	5.0448e+03	6.6829e+03	1.1194e+04
93	31-Jul-2012	418.7450	273.6514	2.5355e+03	5.7757e+03	5.3411e+03	7.2664e+03	1.1862e+04
94	31-Aug-2012	419.0430	274.9015	2.4915e+03	5.7888e+03	5.7389e+03	7.2644e+03	1.2438e+04
95	30-Sep-2012	422.9590	276.0126	2.6464e+03	5.8369e+03	5.5672e+03	7.2970e+03	1.2692e+04
96	31-Oct-2012	420.4150	277.3019	2.8223e+03	5.9540e+03	5.8758e+03	7802	1.3399e+04
97	30-Nov-2012	424.1300	278.4666	2.7642e+03	5.9602e+03	6.5284e+03	7.7129e+03	1.3608e+04

	Time	ALBI	STEFI	J510	J520	J530	J540	J550
98	31-Dec-2012	433.8990	279.6686	2.8766e+03	6.3502e+03	6.3799e+03	8.3971e+03	1.4088e+04
99	31-Jan-2013	434.1860	280.8690	2.9699e+03	6.4584e+03	7.0024e+03	8.2872e+03	1.3461e+04
100	28-Feb-2013	437.0600	281.9545	2.7684e+03	6.7963e+03	6.9615e+03	8.4143e+03	1.3344e+04

⋮

```
allDataReturnsTable = fillmissing(allDataReturnsTable,'constant',0)
```

allDataReturnsTable = 145×11 timetable

...

	Time	ALBI	STEFI	J510	J520	J530	J540	J550
1	31-Dec-2004	0.0306	0.0062	0.0409	0.0464	0.0601	-0.0052	0.0616
2	31-Jan-2005	0.0145	0	-0.0379	-0.0147	-0.0025	-0.0064	-0.0033
3	28-Feb-2005	0.0199	0	0.0411	0.0259	-0.0067	0.0345	0.0051
4	31-Mar-2005	-0.0367	0.0061	-0.0288	-0.0534	0.0479	0.0134	-0.0505
5	30-Apr-2005	0.0204	0.0059	-0.0997	-0.0479	-0.0607	-0.0582	-0.0177
6	31-May-2005	-0.0010	0.0059	0.0418	0.0282	0.1151	0.1321	0.0429
7	30-Jun-2005	0.0269	0.0056	0.0306	0.0508	0.0978	0.0131	0.0312
8	31-Jul-2005	0.0096	0.0057	0.0540	0.0876	0.0456	0.1106	0.1241
9	31-Aug-2005	0.0006	0.0057	-0.0047	0.0308	0.0346	-0.0026	0.0286
10	30-Sep-2005	0.0008	0.0055	0.1259	0.1044	0.0832	0.0972	0.0475
11	31-Oct-2005	0.0063	0.0057	-0.0315	-0.0458	-0.0048	-0.0018	-0.0472
12	30-Nov-2005	0.0237	0.0055	0.0473	0.0044	-0.0058	-0.0411	0.0203
13	31-Dec-2005	0.0197	0.0058	0.0488	0.0825	0.0803	0.0381	0.0932
14	31-Jan-2006	0.0096	0.0058	0.1228	0.0950	0.0470	0.1567	0.1024
15	28-Feb-2006	0.0078	0.0052	-0.0614	0.0370	-0.0257	-0.0148	0.0080
16	31-Mar-2006	-0.0023	0.0058	0.0881	0.0896	0.0558	0.0935	0.0463
17	30-Apr-2006	0.0116	0.0056	0.0749	-0.0001	0.0567	0.0519	0.0245
18	31-May-2006	-0.0112	0.0058	0.0195	-0.1015	-0.0287	-0.1076	-0.0926
19	30-Jun-2006	-0.0361	0.0056	0.1012	-0.0047	0.0216	0.0111	-0.0713
20	31-Jul-2006	0.0068	0.0059	-0.0400	0.0071	0.0271	-0.0018	0.0076
21	31-Aug-2006	0.0004	0.0061	0.0619	0.0851	0.0564	0.0952	0.0404
22	30-Sep-2006	0.0136	0.0061	0.0018	0.0043	0.0552	0.0450	0.0047
23	31-Oct-2006	0.0311	0.0065	0.0401	0.1036	0.0024	-0.0024	0.1136
24	30-Nov-2006	0.0086	0.0065	0.0151	0.0358	0.0363	0.0525	0.0666
25	31-Dec-2006	0.0151	0.0069	-0.0008	0.0754	0.0735	0.0186	0.0671

	Time	ALBI	STEFI	J510	J520	J530	J540	J550
26	31-Jan-2007	0.0069	0.0069	0.0128	0.0521	0.0062	0.0690	0.0738
27	28-Feb-2007	0.0142	0.0064	0.0376	0.0042	-0.0115	-0.0174	0.0155
28	31-Mar-2007	-0.0049	0.0072	0.1150	0.0348	-0.0017	-0.0044	0.0345
29	30-Apr-2007	0.0153	0.0071	-0.0015	0.0554	0.0690	0.0581	0.0678
30	31-May-2007	-0.0110	0.0073	0.0661	-0.0028	0.0051	0.0258	-0.0066
31	30-Jun-2007	-0.0206	0.0072	0.0038	0.0031	-0.0054	-0.0569	-0.0516
32	31-Jul-2007	0.0034	0.0076	0.0052	0.0001	0.0367	-0.1101	-0.0176
33	31-Aug-2007	0.0065	0.0078	-0.0104	-0.0043	0.0321	0.0224	0.0053
34	30-Sep-2007	0.0234	0.0077	0.1336	0.0164	0.0039	-0.0370	0.0040
35	31-Oct-2007	0.0160	0.0081	0.0013	0.0545	0.0163	0.0945	0.0607
36	30-Nov-2007	-0.0155	0.0080	-0.0364	-0.0453	0.0021	-0.0689	-0.0905
37	31-Dec-2007	0.0086	0.0085	-0.0637	0.0053	-0.0136	0.0161	-0.0208
38	31-Jan-2008	-0.0053	0.0087	0.0252	-0.1480	-0.1291	-0.1400	-0.1633
39	29-Feb-2008	-0.0083	0.0083	0.1854	0.1156	0.0560	-0.0153	0.0734
40	31-Mar-2008	-0.0053	0.0089	-0.0305	-0.0323	0.0262	-0.0515	-0.0513
41	30-Apr-2008	-0.0074	0.0088	0.0427	0.0013	0.0116	-0.0040	0.0588
42	31-May-2008	-0.0250	0.0092	0.0619	-0.0041	0.0716	-0.0132	-0.0145
43	30-Jun-2008	-0.0173	0.0091	0.0124	-0.0963	-0.1027	-0.0643	-0.0567
44	31-Jul-2008	0.0851	0.0096	-0.1937	0.0211	-0.0351	0.1118	0.0931
45	31-Aug-2008	0.0128	0.0098	-0.0370	0.0488	0.0556	0.0992	0.0817
46	30-Sep-2008	0.0243	0.0095	-0.2176	-0.0387	-0.0949	-0.0641	-0.0723
47	31-Oct-2008	-0.0045	0.0113	-0.1827	-0.0741	0.0460	-0.1273	-0.0180
48	30-Nov-2008	0.0461	0.0095	0.0538	-0.1108	0.0311	0.0901	-0.0786
49	31-Dec-2008	0.0693	0.0094	0.0044	0.0508	0.0314	0.0324	0.0769
50	31-Jan-2009	-0.0238	0.0093	-0.0313	-0.0651	-0.0406	0.1336	-0.0210
51	28-Feb-2009	-0.0286	0.0092	-0.0959	-0.1061	-0.1046	-0.0182	-0.0811
52	31-Mar-2009	0.0004	0.0091	0.1496	0.0438	0.0111	0.0094	0.0347
53	30-Apr-2009	0.0096	0.0045	-0.0203	0.0878	0.0248	0.0491	0.0546
54	31-May-2009	-0.0043	0.0090	0.1619	0.0324	0.1270	0.0497	0.0723
55	30-Jun-2009	-0.0023	0.0089	-0.0853	0.0162	-0.0374	0.0524	0.0370
56	31-Jul-2009	0.0129	0.0044	0.1065	0.0711	0.1519	0.0946	0.0999
57	31-Aug-2009	0.0157	0.0076	0.0189	0.0872	0.0335	-0.0046	0.0431
58	30-Sep-2009	0.0008	0.0062	-0.0011	0.0453	0.0070	0.0283	0.0538

	Time	ALBI	STEFI	J510	J520	J530	J540	J550
59	31-Oct-2009	-0.0022	0.0061	0.0715	0.0140	0.1135	0.0900	0.0649
60	30-Nov-2009	0.0010	0.0062	0.0649	-0.0497	0.0383	0.0132	-0.0343
61	31-Dec-2009	0.0120	0.0061	0.0258	0.0416	0.0269	0.0972	0.0680
62	31-Jan-2010	0.0027	0.0057	-0.0611	-0.0255	-0.0081	-0.0529	-0.0418
63	28-Feb-2010	0.0202	0.0055	-0.0045	0.0134	-0.0321	0.0330	0.0660
64	31-Mar-2010	0.0211	0.0065	0.0975	0.0751	0.0821	0.0740	0.0729
65	30-Apr-2010	0.0126	0.0058	-0.0185	0.0013	0.0331	0.0282	0.0008
66	31-May-2010	-0.0042	0.0059	-0.0640	-0.0319	-0.0739	-0.0345	0.0022
67	30-Jun-2010	0.0027	0.0056	-0.0431	-0.0409	0.0196	-0.0173	-0.0387
68	31-Jul-2010	0.0412	0.0055	0.0655	0.0591	0.0472	0.0565	0.1213
69	31-Aug-2010	0.0299	0.0058	-0.0629	0.0035	-0.0240	0.0037	-0.0255
70	30-Sep-2010	0.0076	0.0054	0.0715	0.0904	0.1123	0.0780	0.1423
71	31-Oct-2010	0.0104	0.0050	0.0875	0.0262	0.0120	0.0136	0.0402
72	30-Nov-2010	-0.0199	0.0054	-0.0066	0.0002	0.0396	0.0085	-0.0071
73	31-Dec-2010	0.0173	0.0051	0.0720	0.0562	0.0390	0.0106	0.0481
74	31-Jan-2011	-0.0214	0.0049	-0.0014	-0.0540	-0.0129	-0.0424	-0.0704
75	28-Feb-2011	0.0009	0.0044	0.0466	-0.0165	0.0168	-0.0033	0.0460
76	31-Mar-2011	0.0049	0.0048	-0.0225	-0.0177	0.0126	-0.0100	-0.0079
77	30-Apr-2011	0.0226	0.0044	0.0055	0.0235	0.0405	0.0232	0.0527
78	31-May-2011	0.0144	0.0049	-0.0241	0.0059	0.0231	0.0368	-0.0092
79	30-Jun-2011	0.0016	0.0046	-0.0303	-0.0026	-0.0208	0.0080	-0.0138
80	31-Jul-2011	0.0146	0.0044	-0.0401	-0.0071	0.0115	-0.0058	-0.0120
81	31-Aug-2011	0.0349	0.0050	-0.0186	-0.0178	-0.0132	-0.0095	0.0371
82	30-Sep-2011	-0.0209	0.0045	-0.0535	-0.0218	-0.0192	0.0408	-0.0362
83	31-Oct-2011	0.0275	0.0047	0.1088	0.0613	0.1563	0.0393	0.0886
84	30-Nov-2011	0.0001	0.0045	0.0114	0.0180	-0.0083	0.0042	0.0310
85	31-Dec-2011	0.0072	0.0045	-0.0567	-0.0015	-0.0359	0.0067	-0.0130
86	31-Jan-2012	0.0206	0.0048	0.0870	0.0811	0.0496	0.0342	0.0448
87	29-Feb-2012	0.0018	0.0044	-0.0188	0.0540	0.0339	0.0377	0.0451
88	31-Mar-2012	0.0012	0.0045	-0.0809	0.0373	0.0217	0.0671	0.0260
89	30-Apr-2012	0.0181	0.0047	0.0268	0.0151	0.0378	0.0462	0.0489
90	31-May-2012	-0.0001	0.0047	-0.0748	-0.0325	-0.0202	-0.0043	-0.0243
91	30-Jun-2012	0.0334	0.0044	0.0222	0.0140	0.0026	0.0642	0.0320

	Time	ALBI	STEFI	J510	J520	J530	J540	J550
92	31-Jul-2012	0.0396	0.0048	-0.0244	0.0551	0.0587	0.0873	0.0597
93	31-Aug-2012	0.0007	0.0046	-0.0174	0.0023	0.0745	-0.0003	0.0486
94	30-Sep-2012	0.0093	0.0040	0.0622	0.0083	-0.0299	0.0045	0.0204
95	31-Oct-2012	-0.0060	0.0047	0.0665	0.0201	0.0554	0.0692	0.0558
96	30-Nov-2012	0.0088	0.0042	-0.0206	0.0010	0.1111	-0.0114	0.0155
97	31-Dec-2012	0.0230	0.0043	0.0407	0.0654	-0.0227	0.0887	0.0353
98	31-Jan-2013	0.0007	0.0043	0.0324	0.0170	0.0976	-0.0131	-0.0445
99	28-Feb-2013	0.0066	0.0039	-0.0679	0.0523	-0.0058	0.0153	-0.0087
100	31-Mar-2013	0.0024	0.0039	-0.0371	0.0331	0.0536	0.0860	0.0153

⋮

Load pre-processed data

Clear environment and remove all plots

```
clc % clears the command window
close all % removes all figures
clear % clears the workspace
```

Load previously prepared data

```
load('PortfolioTheory.mat')
% fill the missing data in the prices
allDataTable = allDataTableFilled;
% compute the simple returns
allDataReturnsTable = tick2ret(allDataTable,'Method','Simple')
```

```
allDataReturnsTable = 145×11 timetable
```

...

	Time	ALBI	STEFI	J510	J520	J530	J540	J550
1	31-Dec-2004	0.0306	0.0062	0.0409	0.0464	0.0601	-0.0052	0.0616
2	31-Jan-2005	0.0145	0	-0.0379	-0.0147	-0.0025	-0.0064	-0.0033
3	28-Feb-2005	0.0199	0.0117	0.0411	0.0259	-0.0067	0.0345	0.0051
4	31-Mar-2005	-0.0367	0.0061	-0.0288	-0.0534	0.0479	0.0134	-0.0505
5	30-Apr-2005	0.0204	0.0059	-0.0997	-0.0479	-0.0607	-0.0582	-0.0177
6	31-May-2005	-0.0010	0.0059	0.0418	0.0282	0.1151	0.1321	0.0429
7	30-Jun-2005	0.0269	0.0056	0.0306	0.0508	0.0978	0.0131	0.0312
8	31-Jul-2005	0.0096	0.0057	0.0540	0.0876	0.0456	0.1106	0.1241
9	31-Aug-2005	0.0006	0.0057	-0.0047	0.0308	0.0346	-0.0026	0.0286

	Time	ALBI	STEFI	J510	J520	J530	J540	J550
10	30-Sep-2005	0.0008	0.0055	0.1259	0.1044	0.0832	0.0972	0.0475
11	31-Oct-2005	0.0063	0.0057	-0.0315	-0.0458	-0.0048	-0.0018	-0.0472
12	30-Nov-2005	0.0237	0.0055	0.0473	0.0044	-0.0058	-0.0411	0.0203
13	31-Dec-2005	0.0197	0.0058	0.0488	0.0825	0.0803	0.0381	0.0932
14	31-Jan-2006	0.0096	0.0058	0.1228	0.0950	0.0470	0.1567	0.1024
15	28-Feb-2006	0.0078	0.0052	-0.0614	0.0370	-0.0257	-0.0148	0.0080
16	31-Mar-2006	-0.0023	0.0058	0.0881	0.0896	0.0558	0.0935	0.0463
17	30-Apr-2006	0.0116	0.0056	0.0749	-0.0001	0.0567	0.0519	0.0245
18	31-May-2006	-0.0112	0.0058	0.0195	-0.1015	-0.0287	-0.1076	-0.0926
19	30-Jun-2006	-0.0361	0.0056	0.1012	-0.0047	0.0216	0.0111	-0.0713
20	31-Jul-2006	0.0068	0.0059	-0.0400	0.0071	0.0271	-0.0018	0.0076
21	31-Aug-2006	0.0004	0.0061	0.0619	0.0851	0.0564	0.0952	0.0404
22	30-Sep-2006	0.0136	0.0061	0.0018	0.0043	0.0552	0.0450	0.0047
23	31-Oct-2006	0.0311	0.0065	0.0401	0.1036	0.0024	-0.0024	0.1136
24	30-Nov-2006	0.0086	0.0065	0.0151	0.0358	0.0363	0.0525	0.0666
25	31-Dec-2006	0.0151	0.0069	-0.0008	0.0754	0.0735	0.0186	0.0671
26	31-Jan-2007	0.0069	0.0069	0.0128	0.0521	0.0062	0.0690	0.0738
27	28-Feb-2007	0.0142	0.0064	0.0376	0.0042	-0.0115	-0.0174	0.0155
28	31-Mar-2007	-0.0049	0.0072	0.1150	0.0348	-0.0017	-0.0044	0.0345
29	30-Apr-2007	0.0153	0.0071	-0.0015	0.0554	0.0690	0.0581	0.0678
30	31-May-2007	-0.0110	0.0073	0.0661	-0.0028	0.0051	0.0258	-0.0066
31	30-Jun-2007	-0.0206	0.0072	0.0038	0.0031	-0.0054	-0.0569	-0.0516
32	31-Jul-2007	0.0034	0.0076	0.0052	0.0001	0.0367	-0.1101	-0.0176
33	31-Aug-2007	0.0065	0.0078	-0.0104	-0.0043	0.0321	0.0224	0.0053
34	30-Sep-2007	0.0234	0.0077	0.1336	0.0164	0.0039	-0.0370	0.0040
35	31-Oct-2007	0.0160	0.0081	0.0013	0.0545	0.0163	0.0945	0.0607
36	30-Nov-2007	-0.0155	0.0080	-0.0364	-0.0453	0.0021	-0.0689	-0.0905
37	31-Dec-2007	0.0086	0.0085	-0.0637	0.0053	-0.0136	0.0161	-0.0208
38	31-Jan-2008	-0.0053	0.0087	0.0252	-0.1480	-0.1291	-0.1400	-0.1633
39	29-Feb-2008	-0.0083	0.0083	0.1854	0.1156	0.0560	-0.0153	0.0734
40	31-Mar-2008	-0.0053	0.0089	-0.0305	-0.0323	0.0262	-0.0515	-0.0513
41	30-Apr-2008	-0.0074	0.0088	0.0427	0.0013	0.0116	-0.0040	0.0588
42	31-May-2008	-0.0250	0.0092	0.0619	-0.0041	0.0716	-0.0132	-0.0145

	Time	ALBI	STEFI	J510	J520	J530	J540	J550
43	30-Jun-2008	-0.0173	0.0091	0.0124	-0.0963	-0.1027	-0.0643	-0.0567
44	31-Jul-2008	0.0851	0.0096	-0.1937	0.0211	-0.0351	0.1118	0.0931
45	31-Aug-2008	0.0128	0.0098	-0.0370	0.0488	0.0556	0.0992	0.0817
46	30-Sep-2008	0.0243	0.0095	-0.2176	-0.0387	-0.0949	-0.0641	-0.0723
47	31-Oct-2008	-0.0045	0.0113	-0.1827	-0.0741	0.0460	-0.1273	-0.0180
48	30-Nov-2008	0.0461	0.0095	0.0538	-0.1108	0.0311	0.0901	-0.0786
49	31-Dec-2008	0.0693	0.0094	0.0044	0.0508	0.0314	0.0324	0.0769
50	31-Jan-2009	-0.0238	0.0093	-0.0313	-0.0651	-0.0406	0.1336	-0.0210
51	28-Feb-2009	-0.0286	0.0092	-0.0959	-0.1061	-0.1046	-0.0182	-0.0811
52	31-Mar-2009	0.0004	0.0091	0.1496	0.0438	0.0111	0.0094	0.0347
53	30-Apr-2009	0.0096	0.0045	-0.0203	0.0878	0.0248	0.0491	0.0546
54	31-May-2009	-0.0043	0.0090	0.1619	0.0324	0.1270	0.0497	0.0723
55	30-Jun-2009	-0.0023	0.0089	-0.0853	0.0162	-0.0374	0.0524	0.0370
56	31-Jul-2009	0.0129	0.0044	0.1065	0.0711	0.1519	0.0946	0.0999
57	31-Aug-2009	0.0157	0.0076	0.0189	0.0872	0.0335	-0.0046	0.0431
58	30-Sep-2009	0.0008	0.0062	-0.0011	0.0453	0.0070	0.0283	0.0538
59	31-Oct-2009	-0.0022	0.0061	0.0715	0.0140	0.1135	0.0900	0.0649
60	30-Nov-2009	0.0010	0.0062	0.0649	-0.0497	0.0383	0.0132	-0.0343
61	31-Dec-2009	0.0120	0.0061	0.0258	0.0416	0.0269	0.0972	0.0680
62	31-Jan-2010	0.0027	0.0057	-0.0611	-0.0255	-0.0081	-0.0529	-0.0418
63	28-Feb-2010	0.0202	0.0055	-0.0045	0.0134	-0.0321	0.0330	0.0660
64	31-Mar-2010	0.0211	0.0065	0.0975	0.0751	0.0821	0.0740	0.0729
65	30-Apr-2010	0.0126	0.0058	-0.0185	0.0013	0.0331	0.0282	0.0008
66	31-May-2010	-0.0042	0.0059	-0.0640	-0.0319	-0.0739	-0.0345	0.0022
67	30-Jun-2010	0.0027	0.0056	-0.0431	-0.0409	0.0196	-0.0173	-0.0387
68	31-Jul-2010	0.0412	0.0055	0.0655	0.0591	0.0472	0.0565	0.1213
69	31-Aug-2010	0.0299	0.0058	-0.0629	0.0035	-0.0240	0.0037	-0.0255
70	30-Sep-2010	0.0076	0.0054	0.0715	0.0904	0.1123	0.0780	0.1423
71	31-Oct-2010	0.0104	0.0050	0.0875	0.0262	0.0120	0.0136	0.0402
72	30-Nov-2010	-0.0199	0.0054	-0.0066	0.0002	0.0396	0.0085	-0.0071
73	31-Dec-2010	0.0173	0.0051	0.0720	0.0562	0.0390	0.0106	0.0481
74	31-Jan-2011	-0.0214	0.0049	-0.0014	-0.0540	-0.0129	-0.0424	-0.0704
75	28-Feb-2011	0.0009	0.0044	0.0466	-0.0165	0.0168	-0.0033	0.0460

	Time	ALBI	STEFI	J510	J520	J530	J540	J550
76	31-Mar-2011	0.0049	0.0048	-0.0225	-0.0177	0.0126	-0.0100	-0.0079
77	30-Apr-2011	0.0226	0.0044	0.0055	0.0235	0.0405	0.0232	0.0527
78	31-May-2011	0.0144	0.0049	-0.0241	0.0059	0.0231	0.0368	-0.0092
79	30-Jun-2011	0.0016	0.0046	-0.0303	-0.0026	-0.0208	0.0080	-0.0138
80	31-Jul-2011	0.0146	0.0044	-0.0401	-0.0071	0.0115	-0.0058	-0.0120
81	31-Aug-2011	0.0349	0.0050	-0.0186	-0.0178	-0.0132	-0.0095	0.0371
82	30-Sep-2011	-0.0209	0.0045	-0.0535	-0.0218	-0.0192	0.0408	-0.0362
83	31-Oct-2011	0.0275	0.0047	0.1088	0.0613	0.1563	0.0393	0.0886
84	30-Nov-2011	0.0001	0.0045	0.0114	0.0180	-0.0083	0.0042	0.0310
85	31-Dec-2011	0.0072	0.0045	-0.0567	-0.0015	-0.0359	0.0067	-0.0130
86	31-Jan-2012	0.0206	0.0048	0.0870	0.0811	0.0496	0.0342	0.0448
87	29-Feb-2012	0.0018	0.0044	-0.0188	0.0540	0.0339	0.0377	0.0451
88	31-Mar-2012	0.0012	0.0045	-0.0809	0.0373	0.0217	0.0671	0.0260
89	30-Apr-2012	0.0181	0.0047	0.0268	0.0151	0.0378	0.0462	0.0489
90	31-May-2012	-0.0001	0.0047	-0.0748	-0.0325	-0.0202	-0.0043	-0.0243
91	30-Jun-2012	0.0334	0.0044	0.0222	0.0140	0.0026	0.0642	0.0320
92	31-Jul-2012	0.0396	0.0048	-0.0244	0.0551	0.0587	0.0873	0.0597
93	31-Aug-2012	0.0007	0.0046	-0.0174	0.0023	0.0745	-0.0003	0.0486
94	30-Sep-2012	0.0093	0.0040	0.0622	0.0083	-0.0299	0.0045	0.0204
95	31-Oct-2012	-0.0060	0.0047	0.0665	0.0201	0.0554	0.0692	0.0558
96	30-Nov-2012	0.0088	0.0042	-0.0206	0.0010	0.1111	-0.0114	0.0155
97	31-Dec-2012	0.0230	0.0043	0.0407	0.0654	-0.0227	0.0887	0.0353
98	31-Jan-2013	0.0007	0.0043	0.0324	0.0170	0.0976	-0.0131	-0.0445
99	28-Feb-2013	0.0066	0.0039	-0.0679	0.0523	-0.0058	0.0153	-0.0087
100	31-Mar-2013	0.0024	0.0039	-0.0371	0.0331	0.0536	0.0860	0.0153

⋮

```
% save the data
save("PortfolioTheory.mat",'allDataTable','allDataReturnsTable')
```

Different kinds of returns when considering compounding and means

We consider three different type of returns and there compunding. Here we use the ALBI index to demonstrate these. Prices at time t given by P_t

```
prc = allDataTable.ALBI;
```

The price relatives from time $t-1$ to t are x_t

1. Relative

$$x_t = \frac{P_t}{P_{t-1}}$$

The returns R_t (Simple) are then

2. Simple

$$R_t = \frac{P_t - P_{t-1}}{P_{t-1}} = x_t - 1$$

The continous returns are μ_t (Continous)

3. Continuous

$$\mu_t = \ln(P_r) - \ln(P_{t-1}) \iff P_t = e^{\mu_t} P_{t-1} \iff x_t = e^{\mu_t}$$

This is mathematically useful because it implies that: $P_t = P_0 e^{\mu_t t}$ given an initial price P_0.

It is computational convenient because returns can be efficientily computed over vectors

```
mut = diff(log(prc))
```

```
mut = 145×1
    0.0301
    0.0144
    0.0197
   -0.0374
    0.0202
   -0.0010
    0.0265
    0.0096
    0.0006
    0.0008
     ⋮
```

The arithmetic average of this gives the mean continous returns: $\bar{\mu} = \frac{1}{n}\sum_{t=1}^{n} \mu_t$

```
mu =mean(mut)
```

```
mu = 0.0066
```

If the prices are log-normally distributed then the continous-returns will be normally distributed (they aren't on real world historic data).

Geometric returns are geometrically compounding over discrete time-intervals for times $t = 0, 1, 2, \ldots, t, \ldots, n$ where there are $n+1$ prices and n returns:

$$(R_1+1)(R_2+1)\ldots(R_t+1)\ldots(R_n+1) = \frac{P_1}{P_0}\frac{P_2}{P_1}\cdots\frac{P_t}{P_{t-1}}\cdots\frac{P_n}{P_{n-1}} = \frac{P_n}{P_0}$$

```
R = (prc(2:end)./prc(1:end-1))-1
```

```
R = 145×1
    0.0306
    0.0145
```

```
 0.0199
-0.0367
 0.0204
-0.0010
 0.0269
 0.0096
 0.0006
 0.0008
  ⋮
```

The mean geometric returns are: $\mu_G = \left(\prod_{t=1}^{n} x_t\right)^{\frac{1}{n}} - 1 = \left(\prod_{t=1}^{n} (R_t + 1)\right)^{\frac{1}{n}} - 1$

```
muG = exp(mean(log(R+1)))-1
```

```
muG = 0.0066
```

The mean arithmetic returns are: $\mu_A = \frac{1}{n}\sum_{t=1}^{n} R_t$

```
muA = mean(R)
```

```
muA = 0.0068
```

These are not the same. Note that because $\mu = \ln(\mu_G + 1)$over the entire period we have

```
prc(1) * exp((length(prc)-1)*mu)
```

```
ans = 533.4670
```

```
prc(1) * (muG+1)^(length(prc)-1)
```

```
ans = 533.4670
```

```
prc(end)
```

```
ans = 533.4670
```

We will typically use geometric returns compound monthly because we will hold the portfolio for a single month. These will be annualised when presented i.e. NACM (Nominally Annualised Compounded Monthly). Returns are usual quote as CAGR (Compound Annual Growth Rate) returns. This can be managed by a portfolio object, however we should make sure that the returns being used are those correct for the use case. Not all portfolio optimisations use monthly geometrically compounded returns. For example, when modelling a long-short hedge fund we will use arithmetically linked returns, and arithmetic averaging (because we will assume the fund has fixed leverage through multiple rebalances), while for a mutual (pension) fund we will using geometric compounding to link the returns across the months, weeks or days. Another example is when holding a balanced fund from futures close-out to future close-out and then only rebalancing every 3 months, the returns should be quarterly returns, they are then compounded 4 times over a year to get the appropriate annual returns. If the continous-time returns μ_t are Gaussian then the prices will be log-normally distributed.

Create a Portfolio Object

When creating software applications, it is important to organize the various building blocks of your software into related groups.Object-oriented programming (OOP) allows you to group the solver's configuration parameters (properties) with its functions (methods) into a single definition, or *class*. Everything a user will need to properly execute this solver is defined in this class.

An object is an instance of a class. When a program executes, the object is created based on its class definition and behaves in the way defined by the class. The properties of an object represent its state, and its methods represent all the actions a user may perform. In this way, a code author can easily group all the related data and functions for a software system and a user can easily find and use all the capabilities the code author has developed.

Learn more about OOP:

- Object-Oriented Programming in MATLAB
- Working with Objects in MATLAB
- MATLAB self-paced OOP onramp

The `Portfolio` object implements mean-variance portfolio optimization. Every property and function of the `Portfolio` object is public, although some properties and functions are hidden. See `Portfolio` for the properties and functions of the `Portfolio` object.

First create a "standard" `Portfolio` object with Portfolio to incorporate the list of assets.

```
q = Portfolio('AssetList',allDataTable.Properties.VariableNames)
```

```
q =
  Portfolio with properties:

          BuyCost: []
         SellCost: []
     RiskFreeRate: []
        AssetMean: []
       AssetCovar: []
    TrackingError: []
     TrackingPort: []
         Turnover: []
      BuyTurnover: []
     SellTurnover: []
             Name: []
        NumAssets: 11
        AssetList: {'ALBI'  'STEFI'  'J510'  'J520'  'J530'  'J540'  'J550'  'J560'  'J580'  'J590'  'ALSI'}
         InitPort: []
      AInequality: []
      bInequality: []
        AEquality: []
        bEquality: []
       LowerBound: []
       UpperBound: []
      LowerBudget: []
      UpperBudget: []
      GroupMatrix: []
       LowerGroup: []
       UpperGroup: []
           GroupA: []
           GroupB: []
       LowerRatio: []
```

```
      UpperRatio: []
    MinNumAssets: []
    MaxNumAssets: []
       BoundType: []
```

Once the initial portfolio is created, the `estimateAssetMoments` function estimates the mean and standard deviation of portfolio returns.

```
q = estimateAssetMoments(q,allDataReturnsTable{:,:});
```

Compare this with the direct computations of Arithmetic mean

```
q.AssetMean'
```

```
ans = 1×11
    0.0068    0.0059    0.0079    0.0121    0.0191    0.0166    0.0190    0.0131 ···
```

```
mean(allDataReturnsTable{:,:})
```

```
ans = 1×11
    0.0068    0.0059    0.0079    0.0121    0.0191    0.0166    0.0190    0.0131 ···
```

```
% q.AssetCov
% cov(allDataReturnsTable{:,:})
```

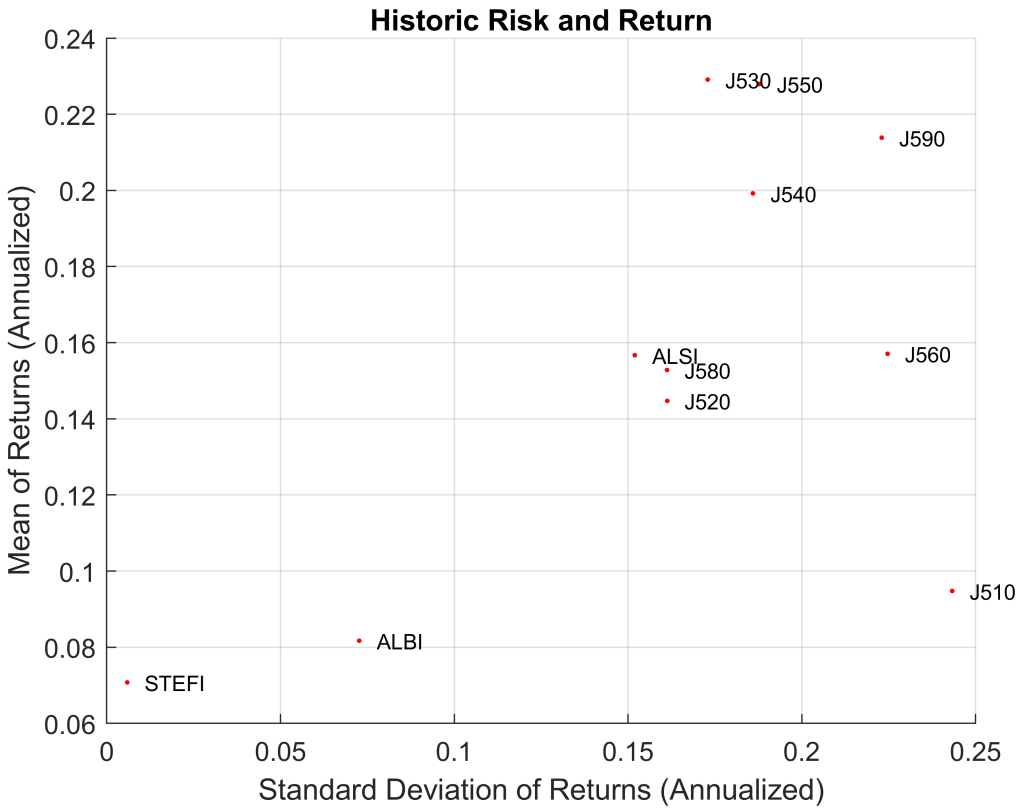
Visualise the Risk-Return relationship

A specialized "helper" function `portfolioexamples_plot` in Local Functions makes it possible to plot all results to be developed here. The first plot shows the distribution of individual assets according to their means and standard deviations of returns.

Note that the `portfolioexamples_plot` function converts monthly total returns into annualized total returns.

```
clf; %clear figure
portfolioexamples_plot('Historic Risk and Return', ...
    {'scatter', sqrt(diag(q.AssetCovar)), q.AssetMean, q.AssetList, '.r'});
```

When you plot normally you must annualise manuall
1. 12*q.AssetMean
2. sqrt(12*diag(q.assetcovar)))




```
12*q.AssetMean'
```

```
ans = 1×11
    0.0817    0.0708    0.0948    0.1447    0.2291    0.1992    0.2279    0.1571 ···
```

```
sqrt(12*diag(q.AssetCovar))'
```

```
ans = 1×11
    0.0726    0.0059    0.2432    0.1613    0.1729    0.1859    0.1877    0.2247 ···
```

Portfolio weights for 2 Assets 2 asset portfolio optimization

Tickers to be considered

```
tickers = {'J510','J550'};
```

Calculating the Risk-Return trade-off for two-risky assets

Create a new Portfolio objects that incorporates the specified tickers only and estimate the asset moments.

```
p = Portfolio('AssetList',allDataTable.Properties.VariableNames(tickers));
Returns2 = allDataReturnsTable{:,tickers};
p = estimateAssetMoments(p,Returns2);
```

Number of rows of weights

```
rowNum = 60;
```

Generate a weight vector for 2 assets (with size 61 x 2)

```
Wts = (0:rowNum)/rowNum;
PortWts = [Wts',(1-Wts)'];
```

Risk-Return relationship for 2 Assets

Compute return and risk for each weight vector (How are the returns calculated?)

```
ret = estimatePortReturn(p, PortWts')
```

```
ret = 61×1
    0.0190
    0.0188
    0.0186
    0.0184
    0.0183
    0.0181
    0.0179
    0.0177
    0.0175
    0.0173
     ⋮
```

```
rsk = estimatePortRisk(p,PortWts')
```

```
rsk = 61×1
    0.0542
    0.0536
    0.0531
    0.0525
    0.0520
    0.0516
    0.0511
    0.0507
    0.0503
    0.0500
     ⋮
```

We can demonstrate how this is computed

```
mu = mean(Returns2);
Sigma = cov(Returns2);
```

The portfolio return: $R_p = \omega' \mu$

```
PortWts(1,:) * mu'
```

```
ans = 0.0190
```

The portfolio risk: $\sigma_p = \omega' \Sigma \omega$

```
sqrt(PortWts(1,:) * Sigma * PortWts(1,:)')
```

```
ans = 0.0542
```

To vectorise and compute the entire vector of returns

```
PortWts * mu'
```

```
ans = 61×1
    0.0190
    0.0188
    0.0186
    0.0184
    0.0183
    0.0181
    0.0179
    0.0177
    0.0175
    0.0173
     ⋮
```

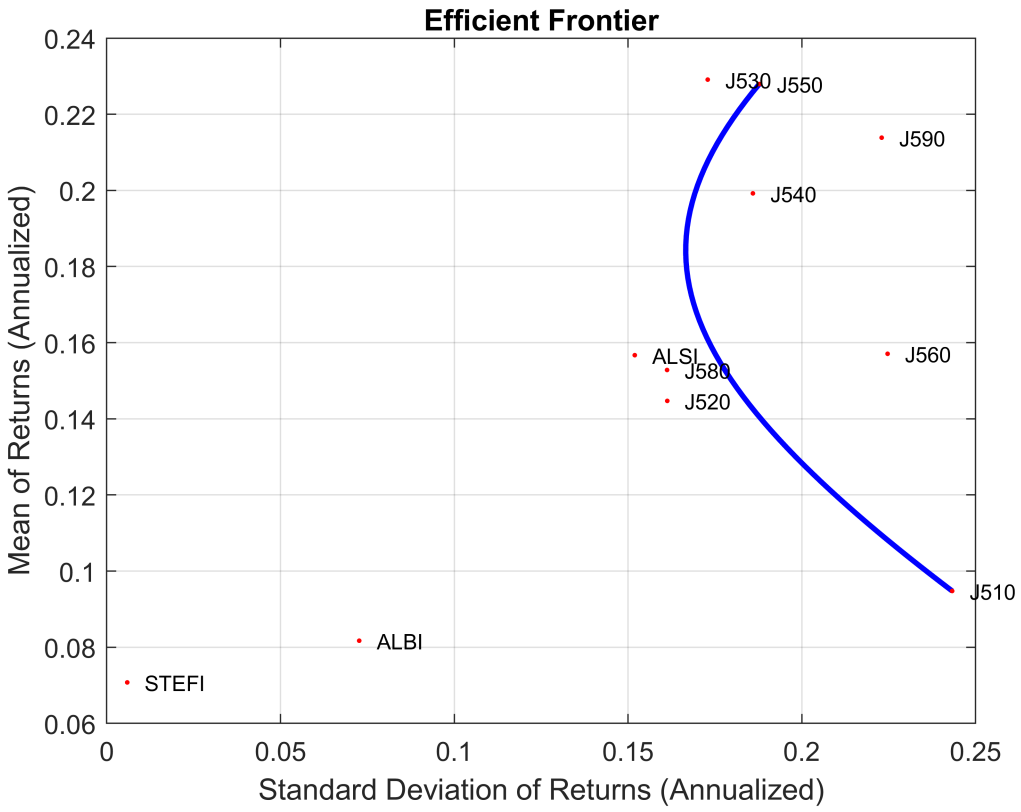
```
sqrt(diag(PortWts * Sigma * PortWts'))
```

```
ans = 61×1
    0.0542
    0.0536
    0.0531
    0.0525
    0.0520
    0.0516
    0.0511
    0.0507
    0.0503
    0.0500
     ⋮
```

Add an Efficient Frontier

Add risk return curves to plot

```
clf;
portfolioexamples_plot('Efficient Frontier', ...
    {'line', rsk, ret}, ...
    {'scatter', sqrt(diag(q.AssetCovar)), q.AssetMean, q.AssetList, '.r'});
```

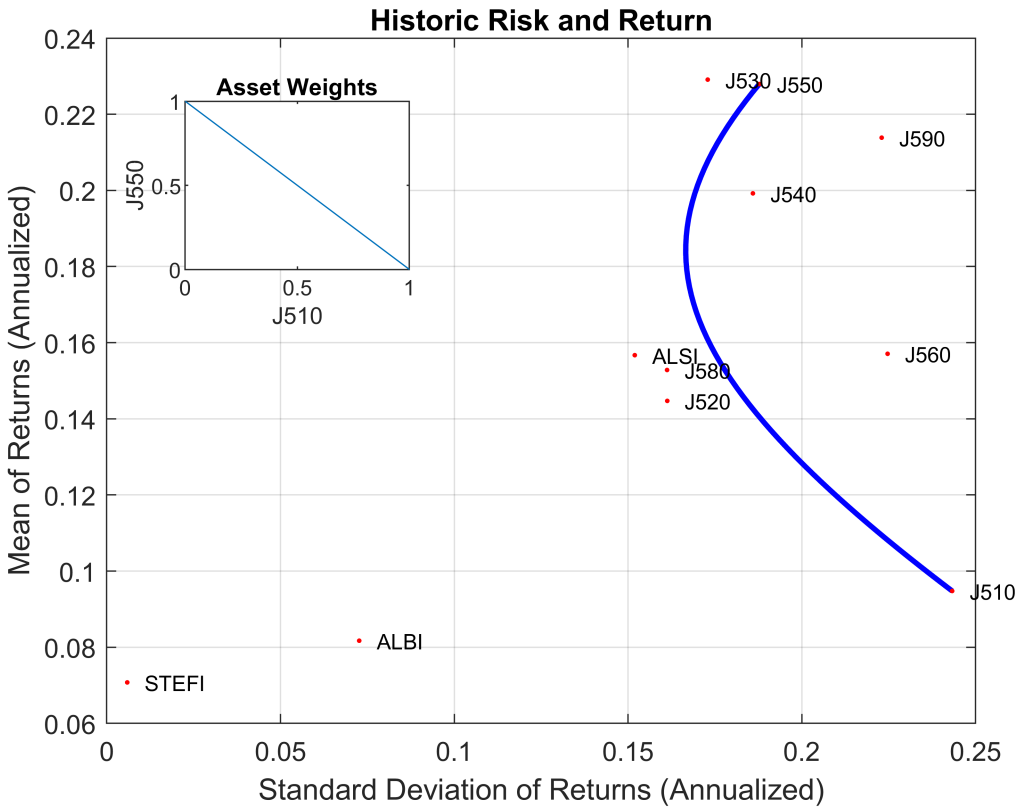


Visualise the weights

Plot the control simplex as an inset plot

```
portfolioexamples_plot('Historic Risk and Return', ...
    {'line', rsk, ret}, ...
    {'scatter', sqrt(diag(q.AssetCovar)), q.AssetMean, q.AssetList, '.r'});
axes('Position',[.2 .65 .2 .2])
box on

plot(PortWts(:,1),PortWts(:,2));
% plot(PortWts)
title('Asset Weights')
xlabel('J510')
ylabel('J550')
hold off
```



The effect of correlations

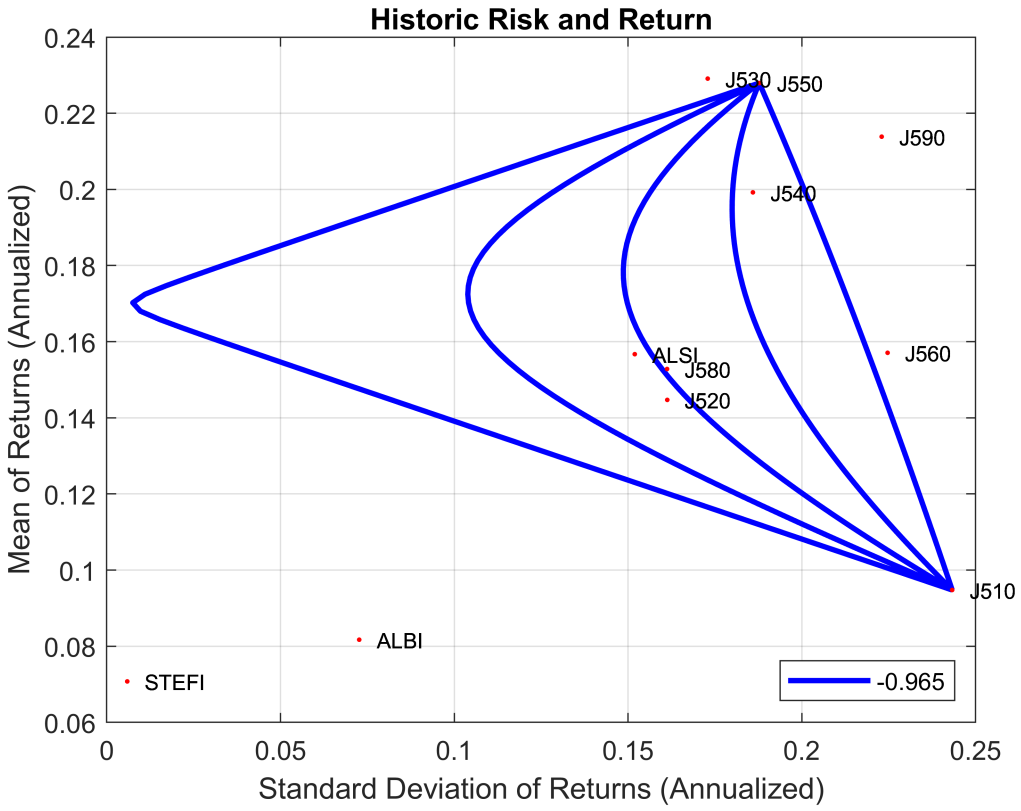
Calculating the Risk-Return trade-off for two-risky assets

```
% With different correlation
rho2ij = [1,0.5,0,-0.5,-0.965];
% We use the original standard deviations
s1ij = sqrt(diag(p.AssetCovar));
```

Compute the portfolio volatility

```
%loop over different correlations
figure;
for j = 1:length(rho2ij)
    %create the family of correlation matrices
    Rho2 = eye(2); %initialise correlations
    Rho2(2,1) = rho2ij(j);
    Rho2(1,2) = Rho2(2,1);%symmetric
    Sigma2 = s1ij .* Rho2 .* s1ij;
    % Compute return and risk
    ret = PortWts * p.AssetMean;
    rsk = sqrt(diag(PortWts * Sigma2 * PortWts'));
    % Matrix to plot (plot without portfolio object)
    portfolioexamples_plot('Historic Risk and Return', ...
        {'line', rsk, ret,{string(rho2ij(j))}}, ...
        {'scatter', sqrt(diag(q.AssetCovar)), q.AssetMean, q.AssetList, '.r'});
```

```
    hold on
end
```



N Asset optimization (fully invested portfolio)

Fully invested portfolio with varying risk aversion

Plot Efficient frontier by varying risk aversion by solving the risk adjusted return maximisation:

$$\max_{\omega} \left\{ \mu\omega - \frac{\gamma}{2}\omega\Sigma\omega \right\} s.t. \omega' \mathbf{1} = 1$$

for the vector of portfolio weights ω where these sum to one. This is done by instead solving:

$$\min_{x} \left\{ \frac{1}{2} x' H x + f' x \right\} s.t. Ax \leq b$$

for an appropriate choice of H, f, A and x. Where we implement the equality constraint as two inequality constraints or use the equality constraint functionality. Here $H \rightarrow \Sigma$ and $f \rightarrow -\mu$ with the correct constraint convention.

```
% Create the range of risk aversion parameters
lambda = linspace(-0.25,0.25,45);
% Fully Invested
q.AEquality = ones(1,length(q.AssetMean));
q.bEquality = 1;
```

Initialise the weights

```
PortWts = NaN(length(lambda),length(q.AssetMean));
```

Find the weight vector for each return level using that $H = \Sigma$ (the asset covariance matrix) and that $f = -\lambda\mu$ for the asset returns μ and the parameter λ that selects the return level at which the risk is minimised.

```
for i = 1:length(lambda)
    f = - lambda(i) * q.AssetMean; % This moves the solution up along the efficient frontier
    H = q.AssetCovar; %the covariance matrix
    PortWts(i,:) = quadprog(H,f,[],[],q.AEquality,q.bEquality) ;
end
```

```
Minimum found that satisfies the constraints.

Optimization completed because the objective function is non-decreasing in
feasible directions, to within the value of the optimality tolerance,
and constraints are satisfied to within the value of the constraint tolerance.

<stopping criteria details>

Minimum found that satisfies the constraints.

Optimization completed because the objective function is non-decreasing in
feasible directions, to within the value of the optimality tolerance,
and constraints are satisfied to within the value of the constraint tolerance.

<stopping criteria details>

Minimum found that satisfies the constraints.

Optimization completed because the objective function is non-decreasing in
feasible directions, to within the value of the optimality tolerance,
and constraints are satisfied to within the value of the constraint tolerance.

<stopping criteria details>

Minimum found that satisfies the constraints.

Optimization completed because the objective function is non-decreasing in
feasible directions, to within the value of the optimality tolerance,
and constraints are satisfied to within the value of the constraint tolerance.

<stopping criteria details>

Minimum found that satisfies the constraints.

Optimization completed because the objective function is non-decreasing in
feasible directions, to within the value of the optimality tolerance,
and constraints are satisfied to within the value of the constraint tolerance.

<stopping criteria details>

Minimum found that satisfies the constraints.

Optimization completed because the objective function is non-decreasing in
feasible directions, to within the value of the optimality tolerance,
and constraints are satisfied to within the value of the constraint tolerance.

<stopping criteria details>

Minimum found that satisfies the constraints.
```

```
Optimization completed because the objective function is non-decreasing in
feasible directions, to within the value of the optimality tolerance,
and constraints are satisfied to within the value of the constraint tolerance.

<stopping criteria details>

Minimum found that satisfies the constraints.

Optimization completed because the objective function is non-decreasing in
feasible directions, to within the value of the optimality tolerance,
and constraints are satisfied to within the value of the constraint tolerance.

<stopping criteria details>

Minimum found that satisfies the constraints.

Optimization completed because the objective function is non-decreasing in
feasible directions, to within the value of the optimality tolerance,
and constraints are satisfied to within the value of the constraint tolerance.

<stopping criteria details>

Minimum found that satisfies the constraints.

Optimization completed because the objective function is non-decreasing in
feasible directions, to within the value of the optimality tolerance,
and constraints are satisfied to within the value of the constraint tolerance.

<stopping criteria details>

Minimum found that satisfies the constraints.

Optimization completed because the objective function is non-decreasing in
feasible directions, to within the value of the optimality tolerance,
and constraints are satisfied to within the value of the constraint tolerance.

<stopping criteria details>

Minimum found that satisfies the constraints.

Optimization completed because the objective function is non-decreasing in
feasible directions, to within the value of the optimality tolerance,
and constraints are satisfied to within the value of the constraint tolerance.

<stopping criteria details>

Minimum found that satisfies the constraints.

Optimization completed because the objective function is non-decreasing in
feasible directions, to within the value of the optimality tolerance,
and constraints are satisfied to within the value of the constraint tolerance.

<stopping criteria details>

Minimum found that satisfies the constraints.

Optimization completed because the objective function is non-decreasing in
feasible directions, to within the value of the optimality tolerance,
and constraints are satisfied to within the value of the constraint tolerance.

<stopping criteria details>

Minimum found that satisfies the constraints.
```

```
Optimization completed because the objective function is non-decreasing in
feasible directions, to within the value of the optimality tolerance,
and constraints are satisfied to within the value of the constraint tolerance.

<stopping criteria details>

Minimum found that satisfies the constraints.

Optimization completed because the objective function is non-decreasing in
feasible directions, to within the value of the optimality tolerance,
and constraints are satisfied to within the value of the constraint tolerance.

<stopping criteria details>

Minimum found that satisfies the constraints.

Optimization completed because the objective function is non-decreasing in
feasible directions, to within the value of the optimality tolerance,
and constraints are satisfied to within the value of the constraint tolerance.

<stopping criteria details>

Minimum found that satisfies the constraints.

Optimization completed because the objective function is non-decreasing in
feasible directions, to within the value of the optimality tolerance,
and constraints are satisfied to within the value of the constraint tolerance.

<stopping criteria details>

Minimum found that satisfies the constraints.

Optimization completed because the objective function is non-decreasing in
feasible directions, to within the value of the optimality tolerance,
and constraints are satisfied to within the value of the constraint tolerance.

<stopping criteria details>

Minimum found that satisfies the constraints.

Optimization completed because the objective function is non-decreasing in
feasible directions, to within the value of the optimality tolerance,
and constraints are satisfied to within the value of the constraint tolerance.

<stopping criteria details>

Minimum found that satisfies the constraints.

Optimization completed because the objective function is non-decreasing in
feasible directions, to within the value of the optimality tolerance,
and constraints are satisfied to within the value of the constraint tolerance.

<stopping criteria details>

Minimum found that satisfies the constraints.

Optimization completed because the objective function is non-decreasing in
feasible directions, to within the value of the optimality tolerance,
and constraints are satisfied to within the value of the constraint tolerance.

<stopping criteria details>

Minimum found that satisfies the constraints.
```

```
Optimization completed because the objective function is non-decreasing in
feasible directions, to within the value of the optimality tolerance,
and constraints are satisfied to within the value of the constraint tolerance.

<stopping criteria details>

Minimum found that satisfies the constraints.

Optimization completed because the objective function is non-decreasing in
feasible directions, to within the value of the optimality tolerance,
and constraints are satisfied to within the value of the constraint tolerance.

<stopping criteria details>

Minimum found that satisfies the constraints.

Optimization completed because the objective function is non-decreasing in
feasible directions, to within the value of the optimality tolerance,
and constraints are satisfied to within the value of the constraint tolerance.

<stopping criteria details>

Minimum found that satisfies the constraints.

Optimization completed because the objective function is non-decreasing in
feasible directions, to within the value of the optimality tolerance,
and constraints are satisfied to within the value of the constraint tolerance.

<stopping criteria details>

Minimum found that satisfies the constraints.

Optimization completed because the objective function is non-decreasing in
feasible directions, to within the value of the optimality tolerance,
and constraints are satisfied to within the value of the constraint tolerance.

<stopping criteria details>

Minimum found that satisfies the constraints.

Optimization completed because the objective function is non-decreasing in
feasible directions, to within the value of the optimality tolerance,
and constraints are satisfied to within the value of the constraint tolerance.

<stopping criteria details>

Minimum found that satisfies the constraints.

Optimization completed because the objective function is non-decreasing in
feasible directions, to within the value of the optimality tolerance,
and constraints are satisfied to within the value of the constraint tolerance.

<stopping criteria details>

Minimum found that satisfies the constraints.

Optimization completed because the objective function is non-decreasing in
feasible directions, to within the value of the optimality tolerance,
and constraints are satisfied to within the value of the constraint tolerance.

<stopping criteria details>

Minimum found that satisfies the constraints.
```

```
Optimization completed because the objective function is non-decreasing in
feasible directions, to within the value of the optimality tolerance,
and constraints are satisfied to within the value of the constraint tolerance.

<stopping criteria details>

Minimum found that satisfies the constraints.

Optimization completed because the objective function is non-decreasing in
feasible directions, to within the value of the optimality tolerance,
and constraints are satisfied to within the value of the constraint tolerance.

<stopping criteria details>

Minimum found that satisfies the constraints.

Optimization completed because the objective function is non-decreasing in
feasible directions, to within the value of the optimality tolerance,
and constraints are satisfied to within the value of the constraint tolerance.

<stopping criteria details>

Minimum found that satisfies the constraints.

Optimization completed because the objective function is non-decreasing in
feasible directions, to within the value of the optimality tolerance,
and constraints are satisfied to within the value of the constraint tolerance.

<stopping criteria details>

Minimum found that satisfies the constraints.

Optimization completed because the objective function is non-decreasing in
feasible directions, to within the value of the optimality tolerance,
and constraints are satisfied to within the value of the constraint tolerance.

<stopping criteria details>

Minimum found that satisfies the constraints.

Optimization completed because the objective function is non-decreasing in
feasible directions, to within the value of the optimality tolerance,
and constraints are satisfied to within the value of the constraint tolerance.

<stopping criteria details>

Minimum found that satisfies the constraints.

Optimization completed because the objective function is non-decreasing in
feasible directions, to within the value of the optimality tolerance,
and constraints are satisfied to within the value of the constraint tolerance.

<stopping criteria details>

Minimum found that satisfies the constraints.

Optimization completed because the objective function is non-decreasing in
feasible directions, to within the value of the optimality tolerance,
and constraints are satisfied to within the value of the constraint tolerance.

<stopping criteria details>

Minimum found that satisfies the constraints.
```

```
Optimization completed because the objective function is non-decreasing in
feasible directions, to within the value of the optimality tolerance,
and constraints are satisfied to within the value of the constraint tolerance.

<stopping criteria details>

Minimum found that satisfies the constraints.

Optimization completed because the objective function is non-decreasing in
feasible directions, to within the value of the optimality tolerance,
and constraints are satisfied to within the value of the constraint tolerance.

<stopping criteria details>

Minimum found that satisfies the constraints.

Optimization completed because the objective function is non-decreasing in
feasible directions, to within the value of the optimality tolerance,
and constraints are satisfied to within the value of the constraint tolerance.

<stopping criteria details>

Minimum found that satisfies the constraints.

Optimization completed because the objective function is non-decreasing in
feasible directions, to within the value of the optimality tolerance,
and constraints are satisfied to within the value of the constraint tolerance.

<stopping criteria details>

Minimum found that satisfies the constraints.

Optimization completed because the objective function is non-decreasing in
feasible directions, to within the value of the optimality tolerance,
and constraints are satisfied to within the value of the constraint tolerance.

<stopping criteria details>

Minimum found that satisfies the constraints.

Optimization completed because the objective function is non-decreasing in
feasible directions, to within the value of the optimality tolerance,
and constraints are satisfied to within the value of the constraint tolerance.

<stopping criteria details>
```

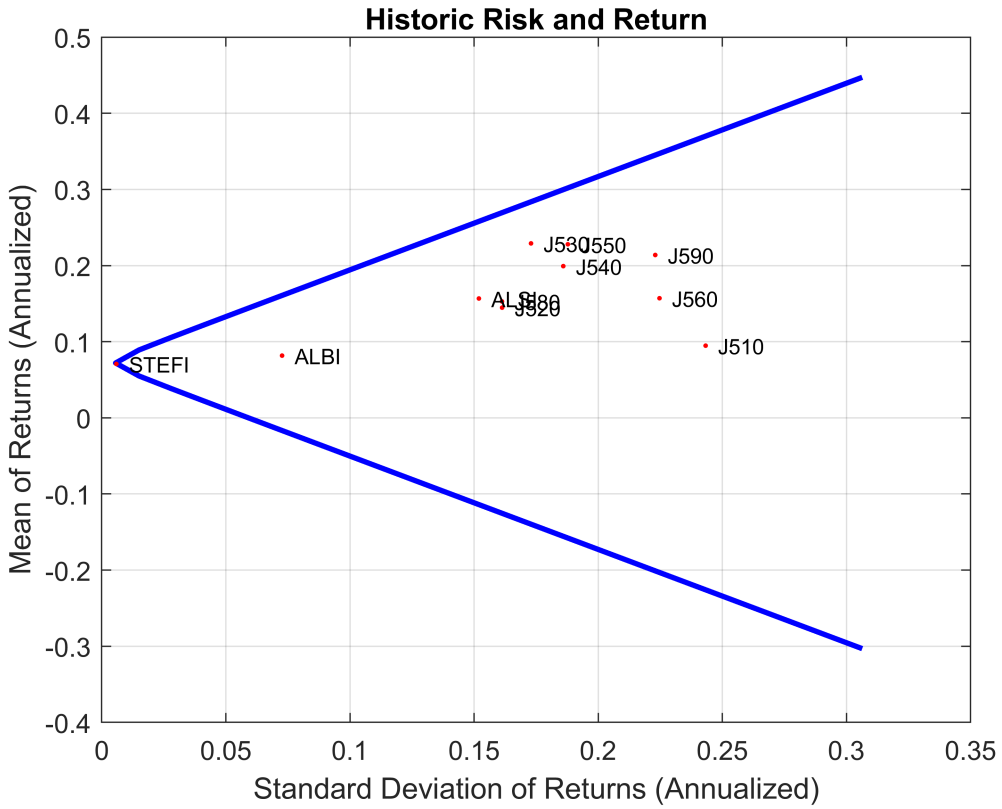
Compute Portfolio Risk and Return

Compute return and risk for each weight vector

```
ret = estimatePortReturn(q, PortWts');
rsk = estimatePortRisk(q,PortWts');
```

Add risk return curves to plot

```
clf;
portfolioexamples_plot('Historic Risk and Return', ...
    {'line', rsk, ret}, ...
    {'scatter', sqrt(diag(q.AssetCovar)), q.AssetMean, q.AssetList, '.r'});
```



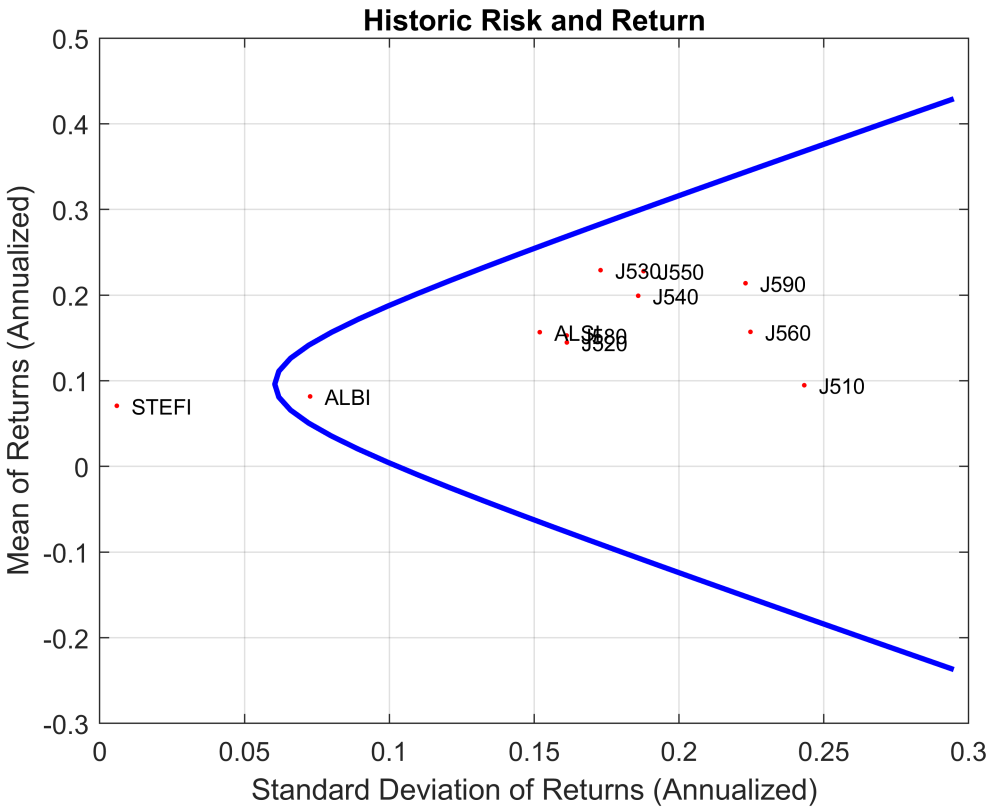
Exclude the Cash and Market Indices

```
tickers = setdiff(allDataTable.Properties.VariableNames,{'STEFI','ALSI'})
```

```
tickers = 1×9 cell
'ALBI'      'J510'      'J520'      'J530'      'J540'      'J550'      'J ···
```

```
Returns = allDataReturnsTable{:,tickers};
mu = mean(Returns);
Sigma = cov(Returns);
% Create the range of risk aversion parameters
lambda = linspace(-0.25,0.25,45);
% Fully Invested
Aeq = ones(1,length(mu));
beq = 1;
% initialise the weights
PortWts = NaN(length(lambda),length(mu));
% set the QP options
optionsQP = optimset('quadprog')';
```

```
optionsQP= optimset(optionsQP,'Display','off');
% generate the efficient frontier
for i = 1:length(lambda)
    f = - lambda(i) * mu; % This moves the solution up along the efficient frontier
    H = Sigma; %the covariance matrix
    [PortWts(i,:),fVal,exitFlag(i)] = quadprog(H,f,[],[],Aeq,beq,[],[],[],optionsQP);
end
% compute risk and return
ret0 = PortWts * mu';
rsk0 = sqrt(diag(PortWts * Sigma * PortWts'));
% plot
clf;portfolioexamples_plot('Historic Risk and Return', ...
    {'line', rsk0, ret0}, ...
    {'scatter', sqrt(diag(q.AssetCovar)), q.AssetMean, q.AssetList, '.r'});
```



Include the no short selling constraint (as an inequality constraint)

We again want to solve an optimisation the maximises the risk adjusted returns. However, now we need to include both the full investment constraint: $\sum_i \omega_i = 1$ (wihich in vector notation is $\omega' \mathbf{1} = 1$ and is an equality constraint), and the inequality constraints $\omega_i \geq 0$ (or in vector notation $\omega \geq 0$). We need to solve the following quadratic problem

$$\max_{\omega} \left\{ \mu\omega - \frac{\gamma}{2}\omega\Sigma\omega \right\} s.t. \omega' \mathbf{1} = 1 \text{and} \omega \geq 0.$$

We can set up the range of the hyper-parameter that controls the risk aversion

```
% Create the range of risk aversion parameters
lambda = 12*linspace(-1,1,45);
```

Change the lambda range later on

The full investment constraints

```
% Fully Invested (equality constraint)
Aeq = ones(1,length(mu));
beq = 1;
```

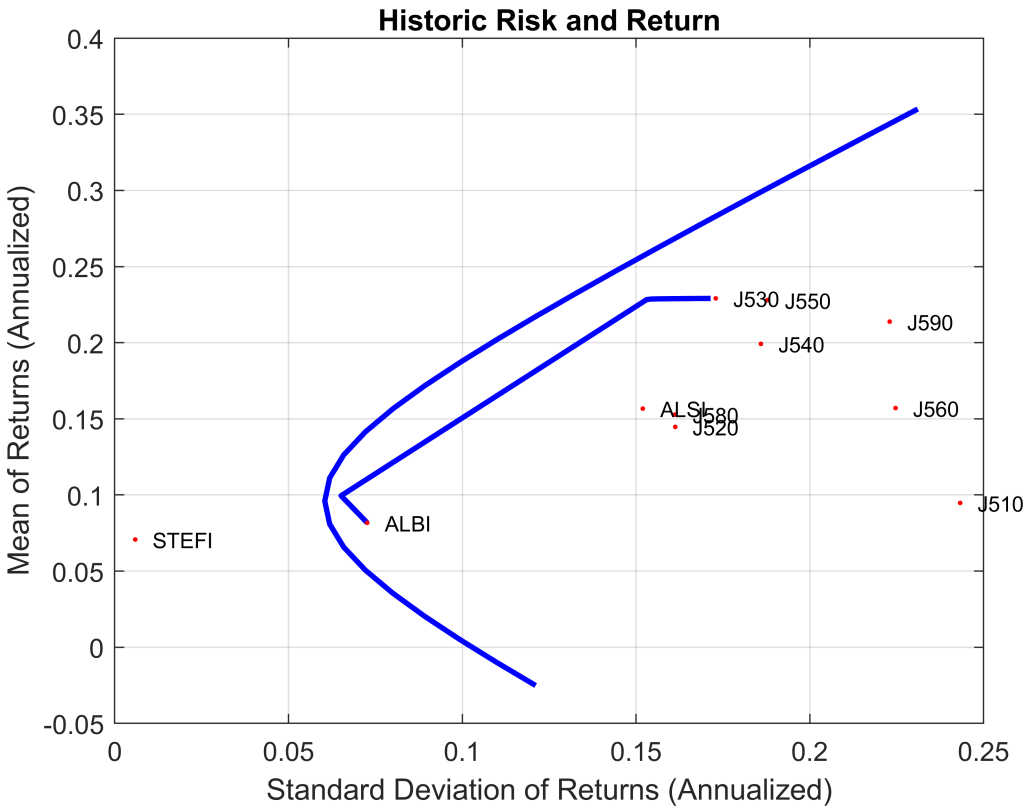
We implement no short-selling (using that $Ax \leq b$). We have that A = - I for the identity marix I, and b = 0.

```
% No Short-selling (inequality constraint)
A = -eye(length(mu));
b = zeros(length(mu),1);
```

Method 1

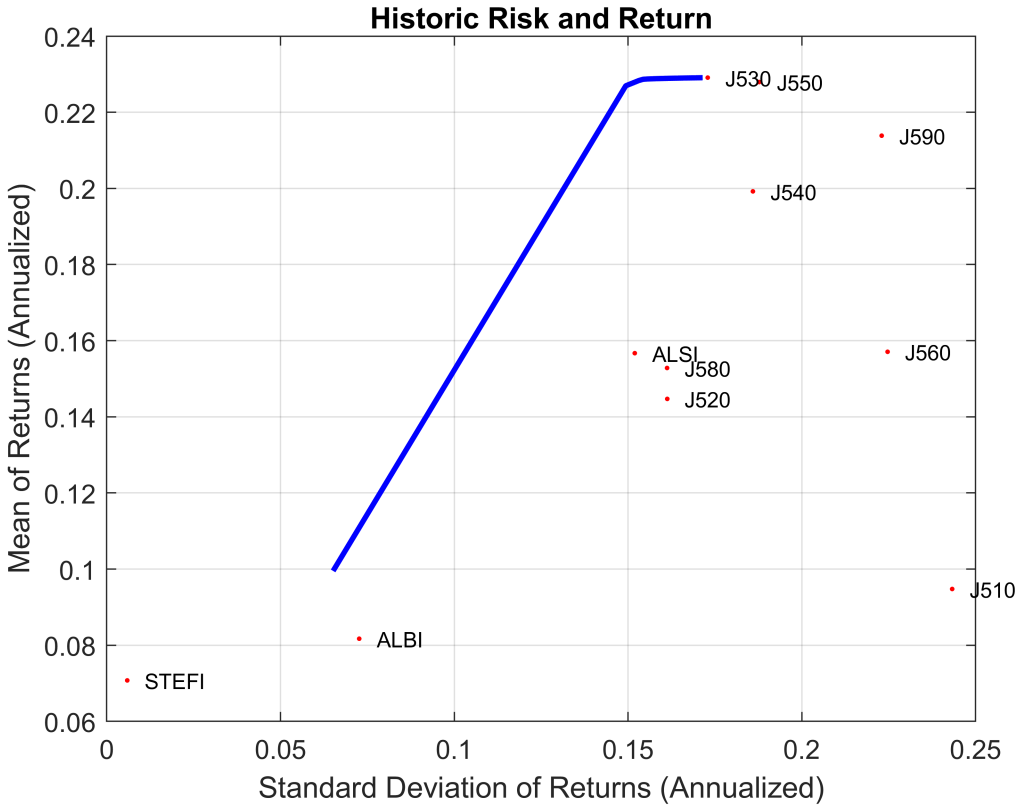
Initialise the weights

```
% initialise the weights
PortWts = NaN(length(lambda),length(mu));
% set the QP options
optionsQP = optimset('quadprog')';
optionsQP= optimset(optionsQP,'Display','off');
% generate the efficient frontier
for i = 1:length(lambda)
    f = - lambda(i) * mu; % This moves the solution up along the efficient frontier
    H = Sigma; %the covariance matrix
    [PortWts(i,:),fVal,exitFlag(i)] = quadprog(H,f,A,b,Aeq,beq,[],[],[],optionsQP);
end
% compute risk and return
ret1 = PortWts * mu';
rsk1 = sqrt(diag(PortWts * Sigma * PortWts'));
% plot
clf;portfolioexamples_plot('Historic Risk and Return', ...
    {'line', rsk0(15:end-5), ret0(15:end-5)}, ...
    {'line', rsk1, ret1}, ...
    {'scatter', sqrt(diag(q.AssetCovar)), q.AssetMean, q.AssetList, '.r'});
```



Change the lambda range

```
% Create the range of risk aversion parameters
lambda = 12*linspace(0,1,45);
% No Short-selling (upper and lower bounds)
ub = ones(length(mu),1);
lb = zeros(length(mu),1);
% initialise the weights
PortWts = NaN(length(lambda),length(mu));
% set the QP options
optionsQP = optimset('quadprog')';
optionsQP= optimset(optionsQP,'Display','off');
% generate the efficient frontier
for i = 1:length(lambda)
    f = - lambda(i) * mu; % This moves the solution up along the efficient frontier
    H = Sigma; %the covariance matrix
    [PortWts(i,:),fVal,exitFlag(i)] = quadprog(H,f,[],[],Aeq,beq,lb,ub,[],optionsQP);
end
% compute risk and return
ret = PortWts * mu';
rsk = sqrt(diag(PortWts * Sigma * PortWts'));
% plot
clf;portfolioexamples_plot('Historic Risk and Return', ...
    {'line', rsk, ret}, ...
    {'scatter', sqrt(diag(q.AssetCovar)), q.AssetMean, q.AssetList, '.r'});
```



Mutual/Pension Fund Efficient Frontier (QP) return targeting

Again only consider Bonds and Equities.

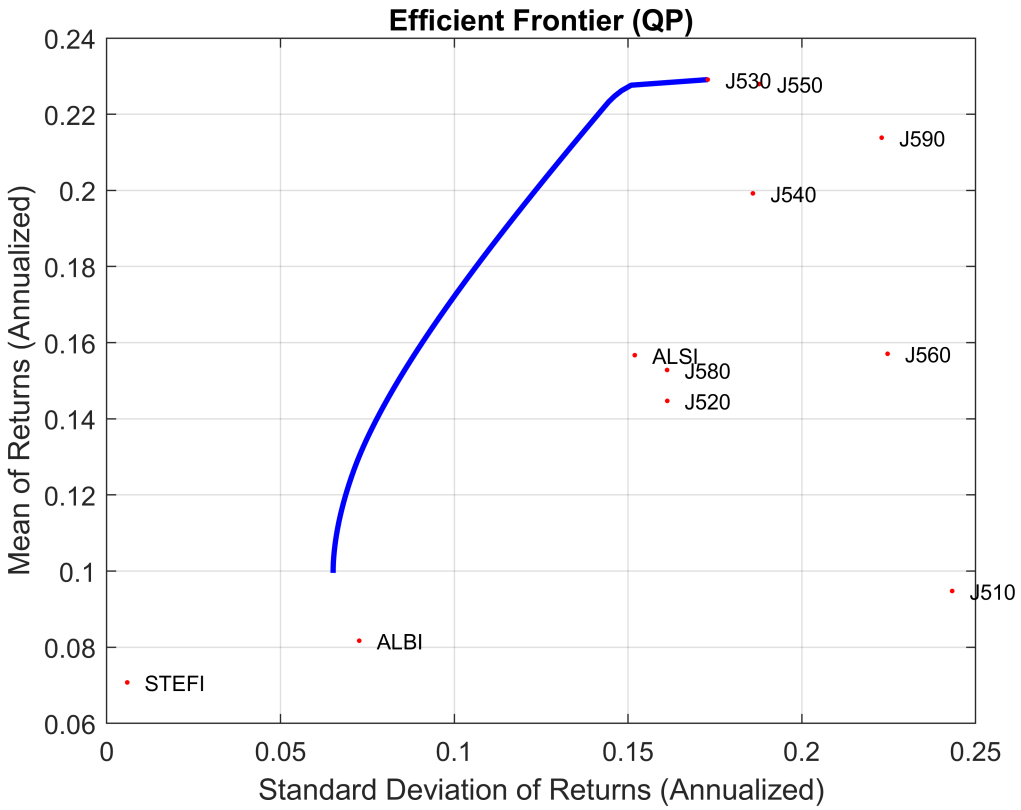
Plot efficient frontier by varying the return target directly

$$\min_{\omega}\left\{\frac{1}{2}\omega'\Sigma\omega\right\} s.t. \quad \omega'\mathbf{1}=1, \omega'\mu=\mu_i \text{and} \omega \geq 0.$$

Method 2

```
% Create the range of risk aversion parameters
% retTarget = min(mu):((max(mu)-min(mu))/90):max(mu);
retTarget = min(ret):((max(ret)-min(ret))/90):max(ret);
% No Short-selling (upper and lower bounds)
ub = ones(length(mu),1);
lb = zeros(length(mu),1);
% Equality constraint (fully invested)
Aeq = ones(1,length(mu));
beq = 1;
% Equality constraint (return target)
Aeq = [Aeq; mu];
beq = [beq; 0];
% initialise the weights
PortWts = NaN(length(retTarget),length(mu));
% set the QP options
optionsQP = optimset('quadprog')';
optionsQP= optimset(optionsQP,'Display','off');
```

```
% generate the efficient frontier
for i = 1:length(retTarget)
    beq(2) = retTarget(i); % This moves the solution up along the efficient frontier
    H = Sigma; %the covariance matrix
    [PortWts(i,:),fVal,exitFlag(i)] = quadprog(H,zeros(size(mu))',[],[],Aeq,beq,lb,ub,[],optio
end
ret = PortWts * mu';
rsk = sqrt(diag(PortWts * Sigma * PortWts'));
clf;
portfolioexamples_plot('Efficient Frontier (QP)', ...
    {'line', rsk, ret}, ...
    {'scatter', sqrt(diag(q.AssetCovar)), q.AssetMean, q.AssetList, '.r'});
```



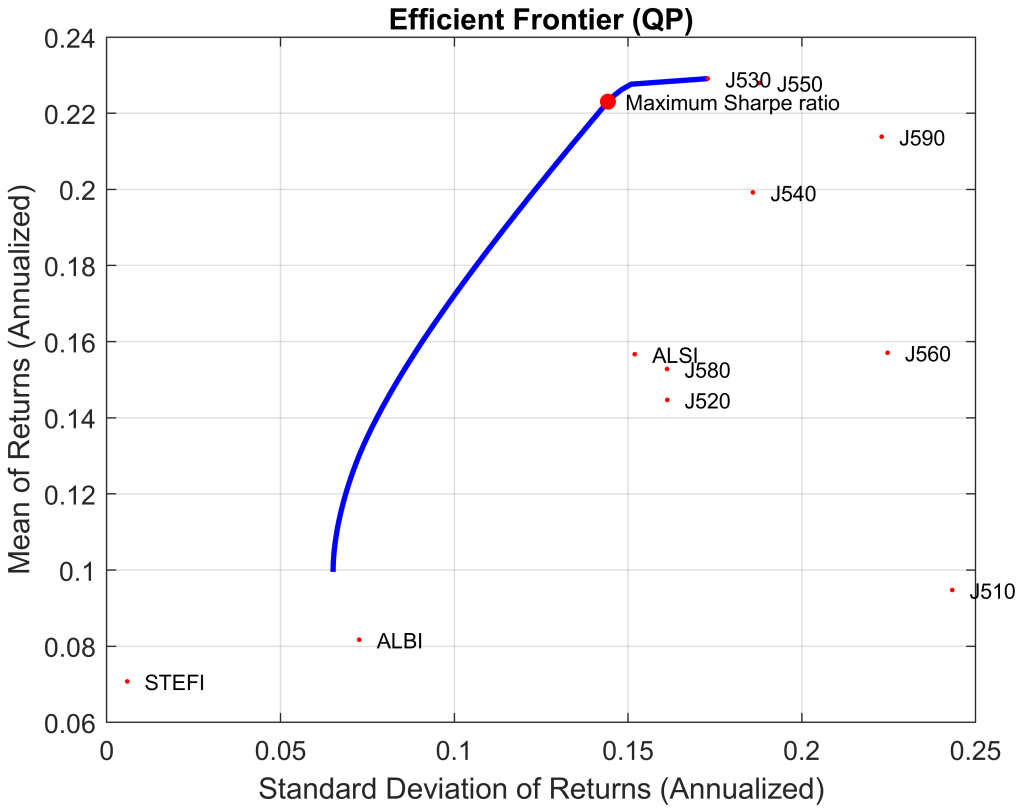
Add the Sharpe Ratio Maximising Portfolio (SQP)

The Sharpe ratio maximising portfolio is the solution too:

$$\max_{\omega} \left\{ \frac{(\omega' \mu - R_{\text{rfr}})}{\sqrt{\omega' \Sigma \omega}} \right\} s.t. \quad \omega' \mathbf{1} = 1 \text{and} \omega \geq 0.$$

```
Returns = allDataReturnsTable{:,tickers};
% risk free rate
RFR = allDataReturnsTable{:,{'STEFI'}};
% monthy sampled arithmetic mean of simple returns
mu = mean(Returns);
% average risk free rate (when to use geometric average)
```

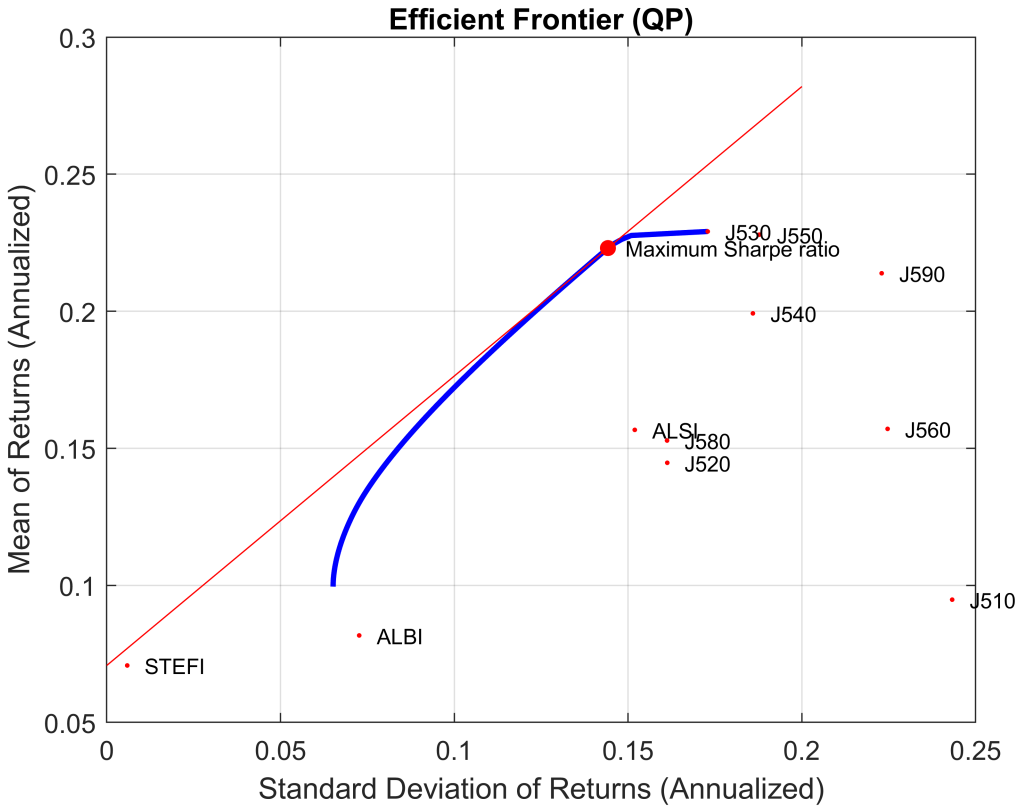
```
ERFR = mean(RFR);
% covariance
Sigma = cov(Returns);
% Equally weighted portfolio
x0 = ones(size(mu))/length(mu);
%initialise the weights
Wts = NaN(1,length(mu));
% objective function to maximise the SR
fn0 = @(x) (-(x*mu' - ERFR)/sqrt(x*Sigma*x'));
% No Short-selling (upper and lower bounds)
ub = ones(length(mu),1);
lb = zeros(length(mu),1);
% Equality constraint (fully invested)
Aeq = ones(1,length(mu));
beq = 1;
% Use SQP to solve for the tangency portfolio
options = optimoptions(@fmincon,'Algorithm','sqp','OptimalityTolerance',1e-8,'Display','off');
x = fmincon(fn0,x0,[],[],Aeq,beq,lb,ub,[],options);
% plot the maximum SR portfolio on EF
retSR = x*mu';
rskSR = sqrt(x*Sigma*x');
clf; portfolioexamples_plot('Efficient Frontier (QP)', ...
    {'line', rsk, ret}, ...
    {'scatter',rskSR,retSR,{'Maximum Sharpe ratio'},'r'}, ...
    {'scatter', sqrt(diag(q.AssetCovar)), q.AssetMean, q.AssetList, '.r'});
```



Fully invested, no short selling and positive risk-aversion

Plot SML (Security Market Line)

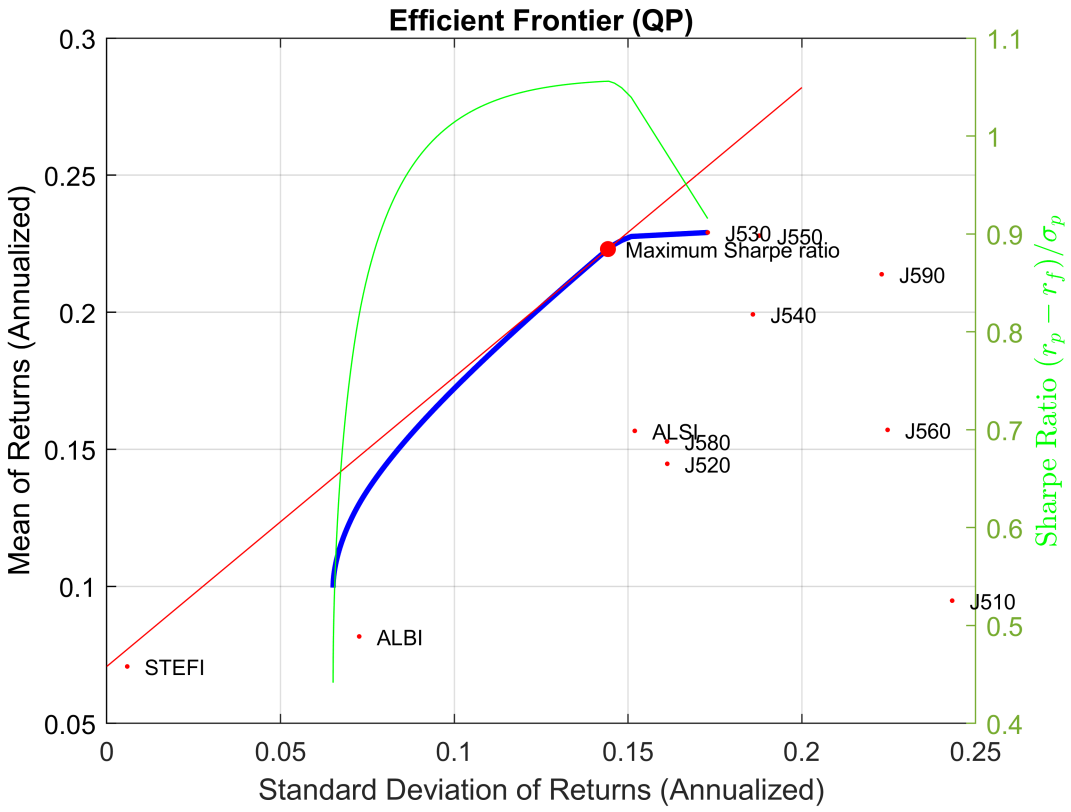
```
hold on
% SML through the tangency portfolio
SML = @(x) (12*ERFR + sqrt(12)*(retSR-ERFR)/rskSR*x);
x = linspace(0,0.20,20);
% plot market line
plot(x,SML(x),'r')
% legend("Market Line")
hold off;
```



Visualise the Sharpe Ratio values

Plot the Sharpe Ratio against risk levels

```
%plot the Sharpe Ratio against risk levels
yyaxis right
ylabel('Sharpe Ratio ${ ( r_p -r_{f} ) / \sigma_p }$','interpreter','latex','Color','g')
% plot the Sharpe Ratio against risk level
hold on
plot(sqrt(12)*rsk,sqrt(12)*(ret-ERFR)./rsk,'-g')
hold off
```


Efficient Frontier (QP)
Mean of Returns (Annualized)
Standard Deviation of Returns (Annualized)
Sharpe Ratio $(r_p - r_f)/\sigma_p$
Maximum Sharpe ratio
J530
J550
J590
J540
ALSI
J580
J520
J560
J510
ALBI
STEFI
0.3
0.25
0.2
0.15
0.1
0.05
0
0.05
0.1
0.15
0.2
0.25
1.1
1
0.9
0.8
0.7
0.6
0.5
0.4