EXCEL PROJECT 2: EFFICIENT FRONTIER

A SKILLS SHOWCASE PROJECT FOR EXCEL

Chinh X. Mai

Last update: June 23, 2022, v.01

CONTENTS

1. PROJECT DESCRIPTION

1.1. WHAT IS TO BE ACHIEVED WITH THIS PROJECT

In this project, I construct many efficient frontiers from various stocks in the DAX30 index to showcase my familiarity with Excel. Spreadsheet is an intuitive way to work with data due to its tabular form of presentation and working with spreadsheet does not require as much knowledge as other analytical tools do. However, the latest version of Excel included in the Microsoft 365 subscription is very sophisticated with many cool features that greatly accelerates the analyzing process. In this project, I will present the implementation of such features in the optimization of a stock portfolio.

Moreover, a short but explanatory introduction to portfolio optimization is also given as the foundation for all the analyses taken in this project. The approach presented in this project might not be the best way to optimize a portfolio or construct an efficient frontier, because the implemented method focuses on finding the analytical solution to the efficient frontier, and it is more efficient to use the numerical approach to achieve the same result. In this project, two approaches are discussed and presented in two separate Excel files. Besides the technical details, some aesthetic elements are also taken into consideration because Excel also provides many features for customizing the spreadsheet to make it more user friendly.

1.2. DETAILS OF THE SHOWCASE SKILLS

For me, Excel is like an inclusive set of tools with a canvas and lots of other packed instruments ready for the analyst to paint a picture or articulate a comprehensive story from raw data, a manifestation of the chaotic real world. Yet Excel does not do that automatically following its user's will, just like all the other tools, but it requires the user to master certain skills to be able to control these tools efficiently. The purpose of this project is to demonstrate some of these skills and my knowledge of them.

In details, these skills include

- Downloading data of chosen stocks from Yahoo! Finance
- Importing multiple data files from a folder using Power Query
- Using MATCH and INDEX functions to transform data from long to wide format
- Illustrating stock data using line chart in Excel
- Utilizing named range and table to simplify the implementation of Excel functions and make these functions more intuitive
- Summarizing information of chosen stocks with statistical functions and the Data Analysis Tools of Excel
- Calculating stock and portfolio statistics using Excel functions of matrix algebra
- Solving optimization problems numerically with Solver
- Generating customizable series with Fill tool
- Constructing complex scatter plots to illustrate different types of efficient frontier
- Generating the table of random values using Data Table tool in What-If Analysis

These are mostly the tools used for calculation and data transformation. The tools used for data aggregation are not implemented due to the nature of this project.

2. THEORETICAL BACKGROUND AND TECHNICAL DETAILS OF IMPLEMENTED TOOLS

2.1. THEORETICAL BACKGROUND: AN INTRODUCTION TO MODERN PORTFOLIO THEORY

Markowitz's modern portfolio theory

Figure 1: Harry Markowitz (source: t.ly/-Wcg)

The modern portfolio theory (MPT) is a practical method for selecting securities of investment to optimize their general return within a desired level of risk. This theory was firstly presented in a paper called "Portfolio Selection" of the American economist Harry Markowitz published in the Journal of Finance in 1952, which brought him a Nobel Prize later.

The key idea of the theory is the diversification of risks of many related stocks when put together in a portfolio. The theory assumes that investors will always want to minimize the risk for a given level of expected return when choosing a security for investment. As a result, the investor must be compensated for bearing a greater risk by receiving higher expected returns. With diversification, any collection of securities can be used to construct a portfolio that fulfill the specific desired expected returns of the investor while minimizing the risk the agent must bear.

What is a portfolio?

Generally, a financial asset is usually characterized by its expected return $E(r)$ or μ and risk σ. When many financial assets are bought together following certain ratios, we have a portfolio. A portfolio can consist of a wide range of assets including stocks, bonds, commodities, cash and cash equivalents, and even real estate, art, and private investments. A portfolio is also characterized by expected return μ_p and risk σ_p.

Diversification and how it works

The expected return of related financial assets can be correlated with each other, which means that their returns tend to change together. The changes can, however, be different in magnitude and direction, so if two correlated stocks are put together in a portfolio, the investor can adjust the ratio of the two stocks to construct a portfolio that has the characteristics that he wants. Moreover, the difference in magnitude of changes can be utilized to minimize the risk so that the portfolio is less risky than it should be for a certain level of return. This phenomenon is regarded as diversification and the core mechanism of MPT.

Portfolio optimization

Constructing a portfolio of certain characteristics is an art because it requires the investor to combine different financial assets and choose an optimal ratio. Portfolio optimization is the process of adjusting a portfolio to make it more efficient in terms of its return over the risk that the portfolio imposes. While the MPT is available to all the investors, each agent might have his or her own adjustments to the approach to include their beliefs of the market as well as the advantage of information in the portfolio. Regardless of which adjustments are added, all investors must start at the same spot, so being able to optimize a portfolio following Markowitz's MPT is an important step in the the process of creating one's own way of portfolio optimization.

2.2. TECHNICAL INTRODUCTION TO THE IMPLEMENTED TOOLS

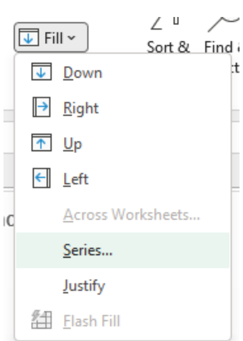


Figure 2: customizable data series with Fill

Brooks (2019) has presented a brief instruction to the implementation of MPT in Excel both theoretically and technically. In this project, I follow basic steps of this short guide, but also improve the implementation by using more advanced and better tools provided by the current version of Excel. The basic operations mostly involve in matrix calculation using provided Excel functions for matrix algebra, so the enhancement of these techniques will not be too different from this root.

The fundamental technique that I use a lot in this project is naming a range or a table by a comprehensive term, so that I can use this term in further calculations in a more intuitive way. This makes the formula become much more understandable and simplify the debugging process if there were any mistake in the implemented formulas. This function can be accessed easily by many ways in Excel, one can name a range easily by changing the name of that range directly from the name box 

on the left side of the Formula Bar and then call this name directly in the formula bar to address the named range.

The ability to generate customized data series is very powerful when plotting. Since the project will focus on the analytical solution of the efficient frontier, this technique is very useful to generate any series to plot the shape of the deduced function. Series creation can be accessed under the Fill drop-down window as shown in Figure 2.

These are just techniques that I find important in this project. The other skills are already listed in the previous section. Further technical details will be discussed along with the analysis in the next section.

3. PMT IN EXCEL I: ASSETS SELECTION FROM DAX30 AND DATA IMPORT

DAX30 index is a stock market index that consists of the 30 major German blue-chip companies trading on the Frankfurt Stock Exchange. The data of DAX30 can be fetched directly from Yahoo! Finance. For a direct view, it can be inspected using the following links:

1. Google
2. Yahoo! Finance

The first link provides the description and explanation of the index, and the second presents the up-to-date DAX30 data together with a downloadable link for viewer the data in other programs. Moreover, Yahoo! Finance also provides a list of components for DAX30. In this project, I will optimize a portfolio of 8 stocks chosen from this list and optimize this portfolio. The list of chosen stocks is given in the tab Schema of the Excel file, the short version of the table is given below

Table 1: Information of chosen stocks

Company	Ticker	Date	Link
Covestro AG	1COV.DE	6/21/2022	Yahoo! Finance - 1COV.DE
adidas AG	ADS.DE	6/21/2022	Yahoo! Finance - ADS.DE
Deutsche Börse AG	DB1.DE	6/21/2022	Yahoo! Finance - BD1.DE
Fresenius Medical Care AG & Co. KGaA	FME.DE	6/21/2022	Yahoo! Finance - FME.DE
MERCK Kommanditgesellschaft auf Aktien	MRK.DE	6/21/2022	Yahoo! Finance - MRK.DE
PUMA SE	PUM.DE	6/21/2022	Yahoo! Finance - PUM.DE
Siemens Aktiengesellschaft	SIE.DE	6/21/2022	Yahoo! Finance - SIE.DE
Zalando SE	ZAL.DE	6/21/2022	Yahoo! Finance - ZAL.DE

These stocks are chosen by random, and they are from different industries, so it might be expected that there would be a diversification effect on the portfolio. The data of daily prices of the chosen stocks for the last 5 years is downloaded and saved in many csv files with the tickers are used as file names (Figure 3)

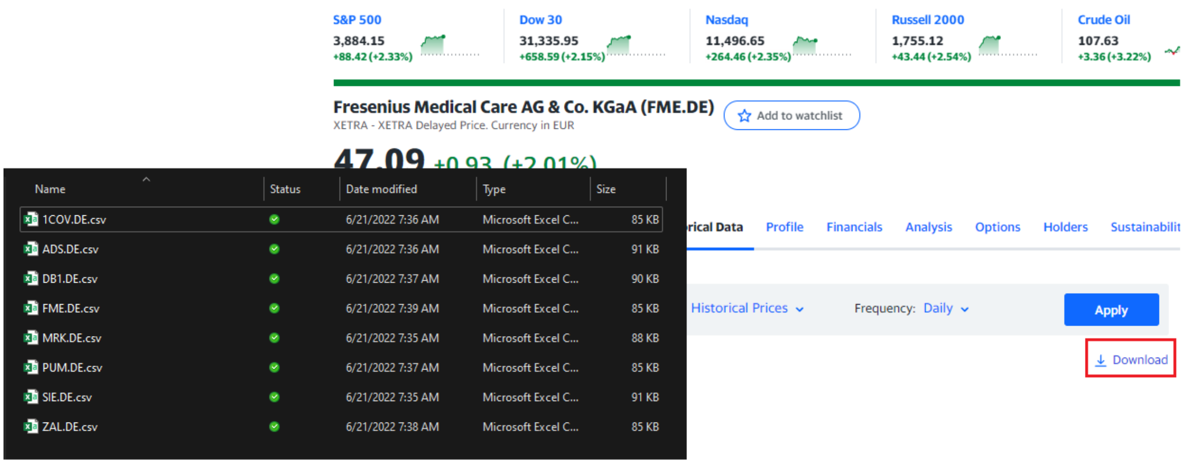


Figure 3: Getting stock data from Yahoo! Finance

Then these csv files are imported into Excel using the Get Data function as shown in the Figure 4. When the data is loaded to the Power Query Editor, it can be adjusted before being loaded into an Excel sheet.

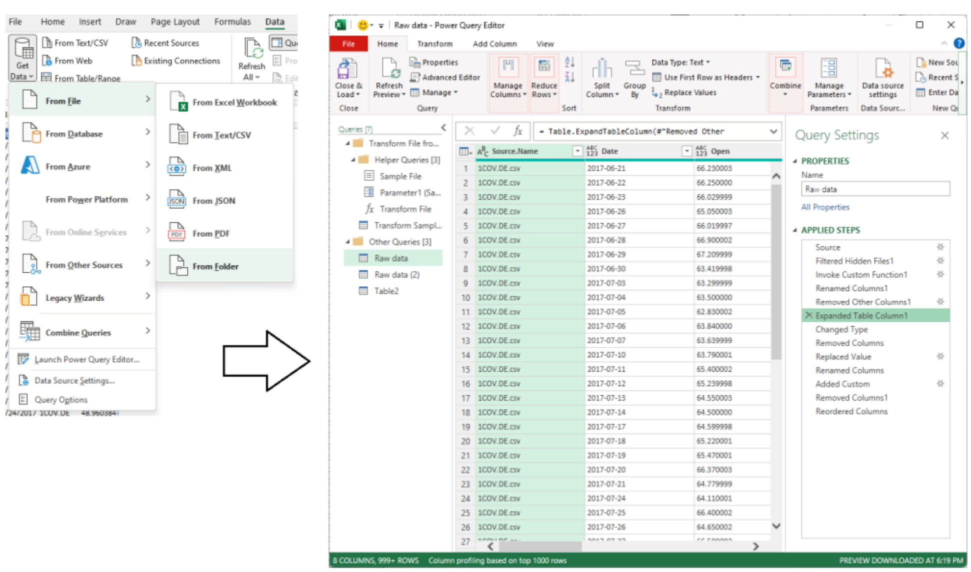
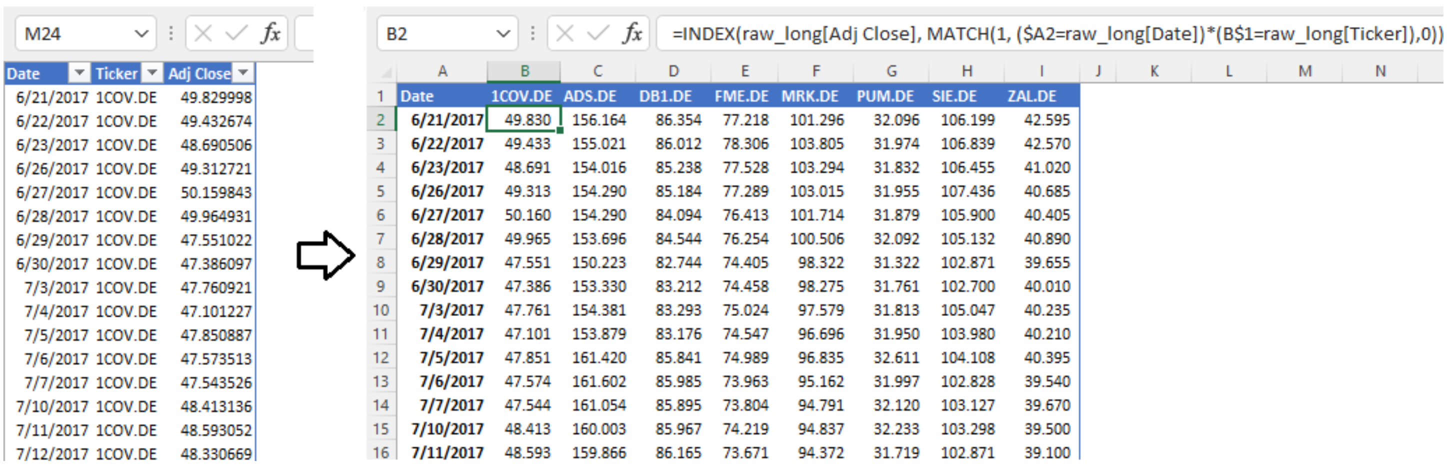


Figure 4: Loading and editing data with Power Query

When data is loaded into Power Query in this way, the data files should have the same structure as the loaded data is the data from all the files stacked on each other. Power query allows for very quick modifications of a huge number of rows. In this case, even though the data only has over ten thousand rows in total, transforming data in Power Query is still more efficient than in Excel. With some modifications, I only take the Adjusted Closed Price, Ticker, and the Date of the price. I prefer Adjusted Closed Price to Close Price since the Adjusted Closed Price is the closing price after having adjusted for all applicable splits and dividend distributions adhering to the standards of CRSP. The data is then loaded into Excel and has a long table form as illustrated in the Figure 5. This form looks neat but is not comfortable for the analysis I will do in this project, so I must transform it into a wide table form, each column of which provides the information of the adjusted closing price of one stock with the date as the index of the table. In Excel, this operation is regarded as unpivoting a table. There are many ways to do this, and I will use MATCH and INDEX functions to perform this task.



M24

Date	Ticker	Adj Close
6/21/2017	1COV.DE	49.829998
6/22/2017	1COV.DE	49.432674
6/23/2017	1COV.DE	48.690506
6/26/2017	1COV.DE	49.312721
6/27/2017	1COV.DE	50.159843
6/28/2017	1COV.DE	49.964931
6/29/2017	1COV.DE	47.551022
6/30/2017	1COV.DE	47.386097
7/3/2017	1COV.DE	47.760921
7/4/2017	1COV.DE	47.101227
7/5/2017	1COV.DE	47.850887
7/6/2017	1COV.DE	47.573513
7/7/2017	1COV.DE	47.543526
7/10/2017	1COV.DE	48.413136
7/11/2017	1COV.DE	48.593052
7/12/2017	1COV.DE	48.330669

B2 =INDEX(raw_long[Adj Close], MATCH(1, ($A2=raw_long[Date])*(B$1=raw_long[Ticker]),0))

	A	B	C	D	E	F	G	H	I
1	Date	1COV.DE	ADS.DE	DB1.DE	FME.DE	MRK.DE	PUM.DE	SIE.DE	ZAL.DE
2	6/21/2017	49.830	156.164	86.354	77.218	101.296	32.096	106.199	42.595
3	6/22/2017	49.433	155.021	86.012	78.306	103.805	31.974	106.839	42.570
4	6/23/2017	48.691	154.016	85.238	77.528	103.294	31.832	106.455	41.020
5	6/26/2017	49.313	154.290	85.184	77.289	103.015	31.955	107.436	40.685
6	6/27/2017	50.160	154.290	84.094	76.413	101.714	31.879	105.900	40.405
7	6/28/2017	49.965	153.696	84.544	76.254	100.506	32.092	105.132	40.890
8	6/29/2017	47.551	150.223	82.744	74.405	98.322	31.322	102.871	39.655
9	6/30/2017	47.386	153.330	83.212	74.458	98.275	31.761	102.700	40.010
10	7/3/2017	47.761	154.381	83.293	75.024	97.579	31.813	105.047	40.235
11	7/4/2017	47.101	153.879	83.176	74.547	96.696	31.950	103.980	40.210
12	7/5/2017	47.851	161.420	85.841	74.989	96.835	32.611	104.108	40.395
13	7/6/2017	47.574	161.602	85.985	73.963	95.162	31.997	102.828	39.540
14	7/7/2017	47.544	161.054	85.895	73.804	94.791	32.120	103.127	39.670
15	7/10/2017	48.413	160.003	85.967	74.219	94.837	32.233	103.298	39.500
16	7/11/2017	48.593	159.866	86.165	73.671	94.372	31.719	102.871	39.100

Figure 5: Unpivoting table in Excel

MATCH and INDEX is a very powerful combination of functions specially used for lookup. They are more flexible than VLOOKUP or even XLOOKUP as they allow for a match of data with multiple conditions. In this case, I use this combination to look for the price of (1) a specific ticker in (2) a specific day, which are two conditions and cannot be done by the normal VLOOKUP and XLOOKUP functions. Moreover, I also named the long table as **raw_long** to make it more convenient to call it in the formula. The result is a long table with date as the index and closing price of each stock stored in each column named by the stock ticker ready for further analysis.

Before going to the detailed analysis, it would be better to do some visual inspection by looking at a line chart of the stock prices to see how the daily prices change in the last five years. This is done easily in Excel by the Charts function in the Insert tab. The result is presented below

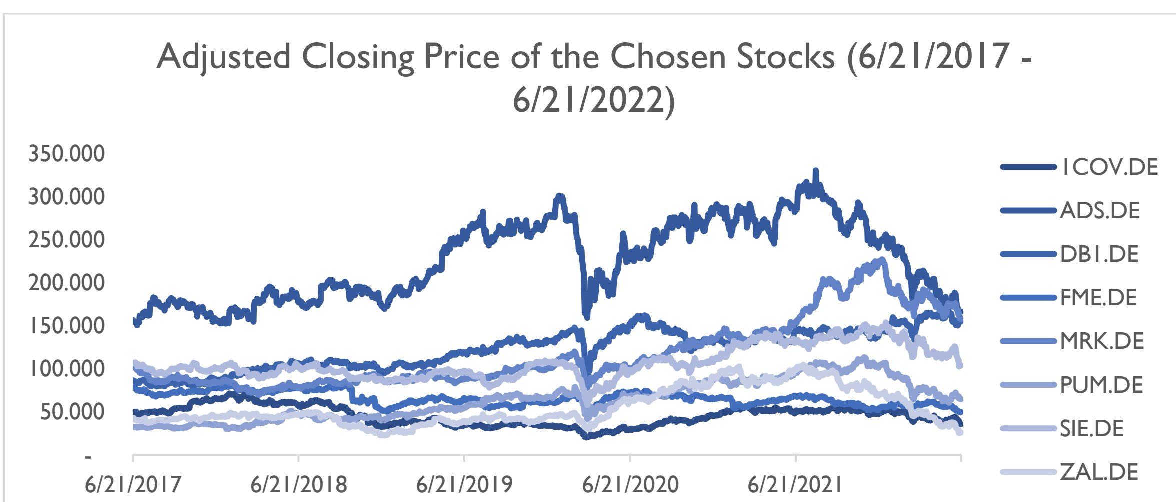


Figure 6: Adjusted closing price of the chosen stocks

Some of the stocks follow a similar trend while the others do not. The magnitudes of change are also varied depending on the stock so a diversification effect can be expected from this data.

4. PMT IN EXCEL 2: EXPECTED RETURNS AND VARIANCE-COVARIANCE MATRIX

Returns is used instead of closed price to eliminate the day-to-day autocorrelation and can be simply calculated by simple or log returns. In this spreadsheet, the simple return is used:

$$r_t = \frac{p_t - p_{(t-1)}}{p_t} \times 100\%$$

where r_t denotes the return at time t and p_t denotes the asset price at time t.

The result is stored in a table named as **return**. This matrix is then used to calculate the statistics that describe the characteristics of the chosen stocks such as mu and sigma respectively with the functions AVERAGE and STDEV.S, the results are stored in the two respective vectors regarded as **v.mu** and **v.sigma** to make further operations more convenient (Figure 7).

Returns Matrix (r)

Date	1COV.DE	ADS.DE	DB1.DE	FME.DE	MRK.DE	PUM.DE	SIE.DE	ZAL.DE
6/22/2017	-0.80%	-0.73%	-0.40%	1.41%	2.48%	-0.38%	0.60%	-0.06%
6/23/2017	-1.50%	-0.65%	-0.90%	-0.99%	-0.49%	-0.44%	-0.36%	-3.64%
6/26/2017	1.28%	0.18%	-0.06%	-0.31%	-0.27%	0.39%	0.92%	-0.82%
6/27/2017	1.72%	0.00%	-1.28%	-1.13%	-1.26%	-0.24%	-1.43%	-0.69%
6/28/2017	-0.39%	-0.39%	0.54%	-0.21%	-1.19%	0.67%	-0.73%	1.20%
6/29/2017	-4.83%	-2.26%	-2.13%	-2.42%	-2.17%	-2.40%	-2.15%	-3.02%
6/30/2017	-0.35%	2.07%	0.57%	0.07%	-0.05%	1.40%	-0.17%	0.90%
7/3/2017	0.79%	0.69%	0.10%	0.76%	-0.71%	0.16%	2.28%	0.56%
7/4/2017	-1.38%	-0.33%	-0.14%	-0.64%	-0.90%	0.43%	-1.02%	-0.06%

Portfolio size (N)	8
No. Observations (T)	1,266

Expected Return (mu')

	1COV.DE	ADS.DE	DB1.DE	FME.DE	MRK.DE	PUM.DE	SIE.DE	ZAL.DE
Expected Return	-0.0036%	0.0255%	0.0568%	-0.0212%	0.0473%	0.0802%	0.0136%	-0.0046%

Standard Deviation (sigma')

	1COV.DE	ADS.DE	DB1.DE	FME.DE	MRK.DE	PUM.DE	SIE.DE	ZAL.DE
Standard Deviation	0.0213280	0.0202348	0.0146662	0.0162896	0.0156702	0.0222735	0.0176520	0.0264066

Figure 7: Returns matrix and vectors of stock statistics

Generating a Variance - Covariance Matrix in Excel

There are many ways to generate a Variance - Covariance matrix in Excel.

1. The first matrix is generated using the Data Analysis function in the tab Data Data Analysis
2. The second matrix is generated using the COVARIANCE.S function, this function uses the sample covariance formula to calculate the covariance/variance
3. The third matrix is generated using the matrix formula. Firstly, the matrix of actual returns (R) is calculated. With T as the number of observations, the Variance - Covariance matrix (V) is calculated using the following formula

$$V = R'R \times (T-1)^{-1}$$

As can be seen from the result (Figure 8), the second and third way of calculation result in the same matrix, which might be caused by the fact that the Data Analysis tool uses another formula to calculate the covariances. The second matrix is named **m.V** and used for further calculation in this spreadsheet.

Var-Cov Matrix (V)

	1COV.DE	ADS.DE	DB1.DE	FME.DE	MRK.DE	PUM.DE	SIE.DE	ZAL.DE
1COV.DE	0.0004545	0.0001923	0.0000988	0.0001172	0.0001081	0.0001965	0.0002202	0.0001839
ADS.DE	0.0001923	0.0004091	0.0001213	0.0000963	0.0001180	0.0003010	0.0001972	0.0002144
DB1.DE	0.0000988	0.0001213	0.0002149	0.0000781	0.0000986	0.0001315	0.0001138	0.0001378
FME.DE	0.0001172	0.0000963	0.0000781	0.0002651	0.0000792	0.0000861	0.0001025	0.0000941
MRK.DE	0.0001081	0.0001180	0.0000986	0.0000792	0.0002454	0.0001380	0.0001267	0.0001363
PUM.DE	0.0001965	0.0003010	0.0001315	0.0000861	0.0001380	0.0004957	0.0001888	0.0002613
SIE.DE	0.0002202	0.0001972	0.0001138	0.0001025	0.0001267	0.0001888	0.0003113	0.0001671
ZAL.DE	0.0001839	0.0002144	0.0001378	0.0000941	0.0001363	0.0002613	0.0001671	0.0006968

	1COV.DE	ADS.DE	DB1.DE	FME.DE	MRK.DE	PUM.DE	SIE.DE	ZAL.DE
1COV.DE	0.0004549	0.0001924	0.0000988	0.0001173	0.0001082	0.0001966	0.0002204	0.0001840
ADS.DE	0.0001924	0.0004094	0.0001214	0.0000963	0.0001181	0.0003013	0.0001974	0.0002146
DB1.DE	0.0000988	0.0001214	0.0002151	0.0000782	0.0000986	0.0001317	0.0001139	0.0001379
FME.DE	0.0001173	0.0000963	0.0000782	0.0002654	0.0000793	0.0000862	0.0001026	0.0000942
MRK.DE	0.0001082	0.0001181	0.0000986	0.0000793	0.0002456	0.0001381	0.0001268	0.0001364
PUM.DE	0.0001966	0.0003013	0.0001317	0.0000862	0.0001381	0.0004961	0.0001889	0.0002616
SIE.DE	0.0002204	0.0001974	0.0001139	0.0001026	0.0001268	0.0001889	0.0003116	0.0001672
ZAL.DE	0.0001840	0.0002146	0.0001379	0.0000942	0.0001364	0.0002616	0.0001672	0.0006973

	1COV.DE	ADS.DE	DB1.DE	FME.DE	MRK.DE	PUM.DE	SIE.DE	ZAL.DE
1COV.DE	0.0004549	0.0001924	0.0000988	0.0001173	0.0001082	0.0001966	0.0002204	0.0001840
ADS.DE	0.0001924	0.0004094	0.0001214	0.0000963	0.0001181	0.0003013	0.0001974	0.0002146
DB1.DE	0.0000988	0.0001214	0.0002151	0.0000782	0.0000986	0.0001317	0.0001139	0.0001379
FME.DE	0.0001173	0.0000963	0.0000782	0.0002654	0.0000793	0.0000862	0.0001026	0.0000942
MRK.DE	0.0001082	0.0001181	0.0000986	0.0000793	0.0002456	0.0001381	0.0001268	0.0001364
PUM.DE	0.0001966	0.0003013	0.0001317	0.0000862	0.0001381	0.0004961	0.0001889	0.0002616
SIE.DE	0.0002204	0.0001974	0.0001139	0.0001026	0.0001268	0.0001889	0.0003116	0.0001672
ZAL.DE	0.0001840	0.0002146	0.0001379	0.0000942	0.0001364	0.0002616	0.0001672	0.0006973

Matrix of actual Returns (R)

Date	1COV.DE	ADS.DE	DB1.DE	FME.DE	MRK.DE	PUM.DE	SIE.DE	ZAL.DE
6/22/2017	-0.7937%	-0.7571%	-0.4531%	1.4303%	2.4298%	-0.4626%	0.5891%	-0.0541%
6/23/2017	-1.4978%	-0.6741%	-0.9571%	-0.9729%	-0.5396%	-0.5231%	-0.3730%	-3.6365%
6/26/2017	1.2815%	0.1526%	-0.1202%	-0.2869%	-0.3172%	0.3053%	0.9083%	-0.8121%
6/27/2017	1.7215%	-0.0255%	-1.3358%	-1.1119%	-1.3102%	-0.3165%	-1.4433%	-0.6837%
6/28/2017	-0.3850%	-0.4106%	0.4785%	-0.1872%	-1.2350%	0.5860%	-0.7388%	1.2049%
6/29/2017	-4.8276%	-2.2853%	-2.1868%	-2.4034%	-2.2202%	-2.4776%	-2.1646%	-3.0157%
6/30/2017	-0.3432%	2.0433%	0.5090%	0.0925%	-0.0945%	1.3213%	-0.1795%	0.8998%
7/3/2017	0.7946%	0.6601%	0.0406%	0.7815%	-0.7565%	0.0833%	2.2714%	0.5669%
7/4/2017	-1.3776%	-0.3511%	-0.1974%	-0.6155%	-0.9520%	0.3501%	-1.0290%	-0.0576%
7/5/2017	1.5952%	4.8750%	3.1473%	0.6145%	0.0969%	1.9880%	0.1095%	0.4646%
7/6/2017	-0.5760%	0.0878%	0.1110%	-1.3472%	-1.7747%	-1.9618%	-1.2431%	-2.1120%
7/7/2017	-0.0594%	-0.3648%	-0.1615%	-0.1941%	-0.4379%	0.3033%	0.2769%	0.3333%
7/10/2017	1.8327%	-0.6781%	0.0270%	0.5845%	0.0018%	0.2725%	0.1519%	-0.4240%
7/11/2017	0.3752%	-0.1112%	0.1736%	-0.7178%	-0.5372%	-1.6763%	-0.4266%	-1.0081%
7/12/2017	-0.5363%	2.0329%	0.5283%	2.2786%	2.2669%	0.2918%	1.8529%	4.0711%
7/13/2017	-0.2446%	1.0390%	0.3587%	0.2091%	-0.6247%	0.2460%	-1.8866%	1.8478%
7/14/2017	0.4545%	0.0854%	0.0259%	0.7479%	-0.0473%	0.6144%	-0.1796%	0.5837%
7/17/2017	1.1801%	0.6943%	-1.4110%	-0.3046%	-1.0637%	1.1672%	-1.0942%	2.5597%
7/18/2017	0.0648%	-1.0425%	-0.8428%	-1.8934%	-1.2698%	-2.6748%	-0.8959%	-8.2654%
7/19/2017	0.8293%	-0.5531%	-0.8173%	-0.5144%	-0.5423%	0.3662%	-1.1581%	-3.5149%
7/20/2017	-1.4522%	0.3374%	-0.5464%	-1.0917%	0.1020%	-1.4876%	1.2729%	2.7537%
7/21/2017	-1.5815%	-1.4441%	-1.2975%	-1.8665%	-1.5475%	-0.8465%	-1.8341%	-3.2627%
7/24/2017	2.1303%	-0.2794%	-1.0098%	0.2432%	-0.7029%	0.4952%	0.3314%	-1.6178%
7/25/2017	-0.9151%	-0.1103%	1.4630%	0.7103%	-0.0574%	0.3413%	0.7599%	0.2614%
7/26/2017	3.1098%	0.1727%	-0.1861%	-0.4799%	-3.3982%	0.3246%	0.0291%	0.9483%

Figure 8: Different ways of generating the Variance-Covariance matrix

The matrix of actual return is named **X** and calculating the deviations of daily returns from the respective expected return of each stock. Then the following formula can be used to calculate the Variance-Covariance matrix easily:

=MMULT(TRANSPOSE(X), X)/(T-1)

The function TRANSPOSE creates a transposed matrix of X, and then the matrix multiplication is done by the function MMULT. **T** is the number of observations, which is the name assigned to this number in another calculation. Therefore, naming range is so powerful because it makes the Excel formula more comprehensive, it is like assigning a name to a variable in other programming languages then use this variable in calculation by calling its name.

5. PMT IN EXCEL 3: EFFICIENT FRONTIER WITH SHORT SELLING

With the expected return column vector $\boldsymbol{\mu}$, variance-covariance matrix $\boldsymbol{V}$, define $\boldsymbol{w}$ as the column vector of the weights of all assets in the to-be-constructed portfolio and I as the column vector 1, the minimum variance portfolio for each given μ_p is determined by solving the minimization problem

$$\min_{\boldsymbol{w}} \boldsymbol{w}'\boldsymbol{V}\,\boldsymbol{w} \text{ subjected to } \boldsymbol{w}'\boldsymbol{\mu} = \mu_p, \boldsymbol{w}'\mathbf{1} = 1$$

The resultant optimal weight is

$$w^* = \frac{C\mu_p - B}{AC - B^2}\boldsymbol{V}^{-1}\boldsymbol{\mu} + \frac{A - B\mu_p}{AC - B^2}\boldsymbol{V}^{-1}\mathbf{1}$$

Where $A = \boldsymbol{\mu}'\boldsymbol{V}^{-1}\boldsymbol{\mu}, B = \boldsymbol{\mu}'\boldsymbol{V}^{-1}\mathbf{1}, C = \mathbf{1}'\boldsymbol{V}^{-1}\mathbf{1}$. Rearranging the formula to make it tidier:

$$w^* = \frac{A\boldsymbol{V}^{-1}\mathbf{1} - B\boldsymbol{V}^{-1}\boldsymbol{\mu}}{AC - B^2} + \frac{C\boldsymbol{V}^{-1}\boldsymbol{\mu} - B\boldsymbol{V}^{-1}\mathbf{1}}{AC - B^2}\mu_p = D + E\mu_p$$

Without the constraint $\boldsymbol{w}'\boldsymbol{\mu} = \mu_p$, the desired portfolio return μ_p is not given, so the solution to the minimization problem is the weight of the minimum variance portfolio (MVP), which is derived by the formula

$$w_{MVP} = \frac{V^{-1}1}{1'V^{-1}1}$$

The corresponding portfolio expected return and standard deviation of the MVP are $\mu_{MVP} = \frac{B}{C}$, $\sigma_{MVP} = \frac{1}{\sqrt{C}}$.

When the desired portfolio return μ_p varies and this problem is solved accordingly for each chosen value μ_p then calculating the corresponding values of portfolio risk σ_p, the collection of all the pairs (μ_p, σ_p) is the minimum variance frontier (MVF). Letting the technical details aside, the function of the MVF is expressed as follows

$$\sigma_p = \left(\frac{C\mu_p^2 - 2B\mu_p + A}{AC - B^2}\right)^{\frac{1}{2}}$$

Efficient Frontier (EF) is the upper part of the MVF, which consists of all the portfolios that have the higher risks and higher returns than the MVP and the portfolios below it on this line. This is the analytical solution to the optimization problem that I want to seek. In Excel, I just need to simply calculate the core values A, B and C as the inputs for these values are already available. The corresponding Excel formulas are given in the table below

A	=MMULT(v.mu,MMULT(MINVERSE(m.V),TRANSPOSE(v.mu)))
B	=MMULT(v.mu,MMULT(MINVERSE(m.V),TRANSPOSE(one)))
C	=MMULT(one,MMULT(MINVERSE(m.V),TRANSPOSE(one)))

Table 2: Excel formulas for the values A, B, and C

With these values, the weight vector of the MVP is calculated following the formula deduced above and so are the characteristics of this portfolio. It should be noticed that the format of the vectors (row vs column) in the spreadsheet is different from how it should be following the formulas. As a result, the Excel formulas must be adjusted accordingly so that they will give correct results. This explains why some might find that the actual Excel formulas might be different from those in the original analytical solutions. Moreover, the values of $A, B,$ and C are also named **c.A, c.B**, and **c.C** in the spreadsheet for the later operations. The Excel formula for the weight is then

=MMULT(MINVERSE(m.V), TRANSPOSE(one))/c.C

The vector **one** in this formula is the vector of 1s mentioned above. The next step is generating the Frontiers, as the function of σ_p requires a reasonable range of μ_p as input, so this series must be generated first. This is easily done by utilizing the Fill feature discussed above. After entering the first value of the series in a cell and choosing Series in the Fill dropdown panel, the Series window will appear as in Figure 9. The Figure shows that I would like to create a series of values from -0.001 to 0.0015 in the column below the firstly entered value of -0.001.

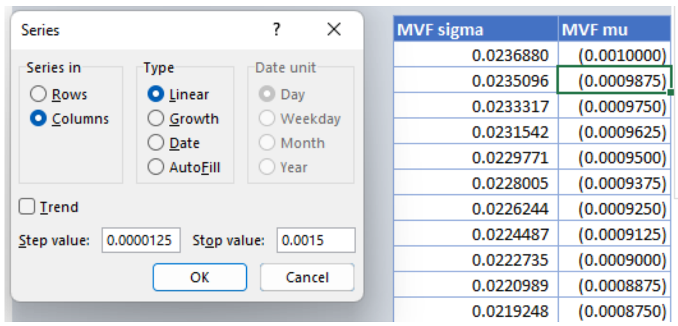


Figure 9: Generating a series with Fill Series

Then the corresponding series of σ_p can be generated accordingly using the function of MVF discussed above. This is how the next column beside the column **MVF mu**, **MVF sigma**, is generated. With these two columns, the statistics of the MVP and the chosen stocks, the MVF and EF can be generated using the scatter plot feature of Excel (Figure 10).

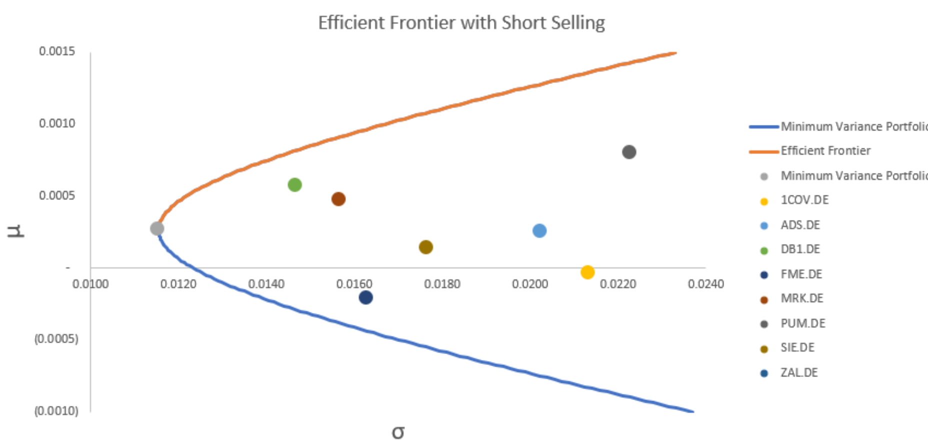


Figure 10: Efficient frontier with short selling

6. PMT IN EXCEL 4: EFFICIENT FRONTIER WITHOUT SHORT SELLING

It is often the case that we want to place additional constraints on the optimization, such as

- Restrict the weights so that none is greater than 10% of overall wealth
- Restrict them to all be positive (i.e. long positions only with no short selling)

In such cases, the Markowitz portfolio allocation problem cannot be solved analytically, so a numerical procedure must be used. If the procedure above is followed repeatedly for different return targets, it will trace out the efficient frontier.

With the expected return column vector **μ**, variance-covariance matrix $\boldsymbol{V}$, define $\boldsymbol{w}$ as the column vector of the weights of all assets in the to-be-constructed portfolio and **I** as the column vector I, when sort selling is not allowed, every element w_i of the weight vector w must be non-negative.

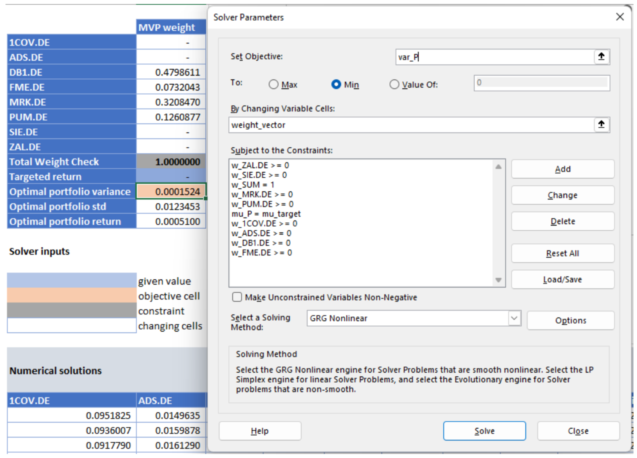


Figure 11: Solver setup

Using the Solver in Excel with the parameters shown in Figure 11, a table of the minimum variance portfolios is generated by repeating this process with different desired expected returns shown in the Table below as the input to construct the EF numerically. When the constraint **mu_P = mu_target** is not given in the constraints list, the result is the weights of the MVP.

Numerical solutions

1COV.DE	ADS.DE	DB1.DE	FME.DE	MRK.DE	PUM.DE	SIE.DE	ZAL.DE	Optimal portfolio std	Optimal portfolio return
0.0951825	0.0149635	0.0785783	0.5152105	0.1109404	-	0.1110031	0.0741218	0.0126801	(0.0000000)
0.0936007	0.0159878	0.0881221	0.5060292	0.1162552	-	0.1086160	0.0713890	0.0126001	0.0000100
0.0917790	0.0161290	0.0978639	0.4969133	0.1216699	-	0.1068078	0.0688371	0.0125228	0.0000200
0.0902547	0.0163396	0.1075328	0.4877780	0.1271860	-	0.1048090	0.0660998	0.0124480	0.0000300
0.0880588	0.0190373	0.1165111	0.4786511	0.1322833	-	0.1022683	0.0631901	0.0123759	0.0000400
0.0863162	0.0191265	0.1262362	0.4695100	0.1376809	(0.0000010)	0.1006290	0.0605022	0.0123065	0.0000500
0.0846672	0.0193177	0.1359233	0.4603689	0.1431206	-	0.0988156	0.0577875	0.0122400	0.0000600
0.0830241	0.0196337	0.1455588	0.4512249	0.1485613	-	0.0969682	0.0550290	0.0121762	0.0000700
0.0813550	0.0201036	0.1551540	0.4421057	0.1539844	-	0.0950773	0.0522200	0.0121153	0.0000800
0.0796607	0.0207697	0.1646782	0.4329884	0.1593967	-	0.0931357	0.0493706	0.0120573	0.0000900
0.0771440	0.0251125	0.1733417	0.4239080	0.1643919	-	0.0892879	0.0468139	0.0120022	0.0001000
0.0751222	0.0261526	0.1827514	0.4147688	0.1696802	-	0.0874658	0.0440590	0.0119501	0.0001100
0.0734081	0.0262661	0.1924411	0.4056310	0.1751398	-	0.0857257	0.0413882	0.0119011	0.0001200
0.0717149	0.0266376	0.2020585	0.3965044	0.1805729	-	0.0838890	0.0386227	0.0118551	0.0001300
0.0699891	0.0273029	0.2115952	0.3873929	0.1859718	-	0.0819511	0.0357969	0.0118122	0.0001400
0.0681956	0.0282799	0.2210608	0.3782987	0.1913257	-	0.0798865	0.0329529	0.0117725	0.0001500
0.0663239	0.0295210	0.2304714	0.3692028	0.1966367	-	0.0776957	0.0301486	0.0117359	0.0001600
0.0643977	0.0308941	0.2398515	0.3600829	0.2019286	-	0.0754245	0.0274207	0.0117026	0.0001700
0.0621233	0.0332612	0.2489828	0.3508897	0.2070603	-	0.0727592	0.0249234	0.0116724	0.0001800

Security

	sigma	mu
MVP	0.0115498	0.0002640
1COV.DE	0.0213280	(0.0000362)
ADS.DE	0.0202348	0.0002548
DB1.DE	0.0146662	0.0005684
FME.DE	0.0162896	(0.0002119)
MRK.DE	0.0156702	0.0004726
PUM.DE	0.0222735	0.0008021
SIE.DE	0.0176520	0.0001359
ZAL.DE	0.0264066	(0.0000456)

Figure 12: Inputs for EF without short selling

The result is an EF without short selling presented in the Figure 13.

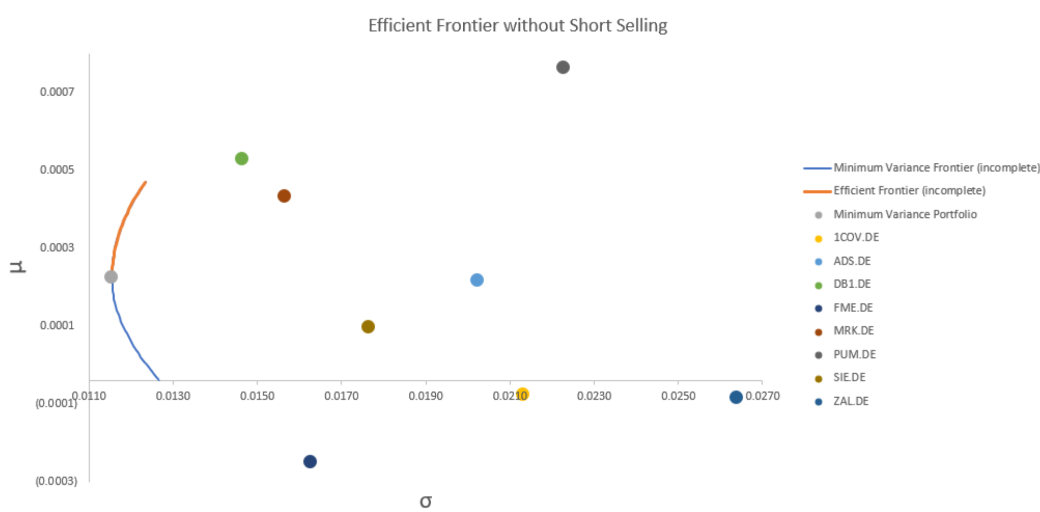


Figure 13: Efficient frontier without short selling

7. PMT IN EXCEL 5: EFFICIENT FRONTIER WITH SHORT SELLING AND RISK-FREE ASSET

In the case there exists a risk-free asset and short selling is allowed, the analytical solution for the efficient frontier with risk-free asset of rate r_f is

$$\mu_p = r_f + \sigma_p \sqrt{A - 2B \times r_f + C \times r_f^2}$$

With A, B, C defined similarly to the case of efficient frontier with short selling.

Tangency portfolio between the efficient frontier with the risk-free asset and the efficient frontier without the risk-free asset is another special portfolio that is extremely relevant, whose weight vector $\boldsymbol{w_{TP}}$ is calculated as follows

$$w_{TP} = \frac{V^{-1}(\mu - r_f)}{1'V^{-1}(\mu - r_f)}$$

These formulas only require one new input, the risk-free rate r_f. For each chosen level of risk-free rate, following the same steps presented in the previous section, the scatter plot of the EF with short selling and the tangent portfolio as in the Figure 14.

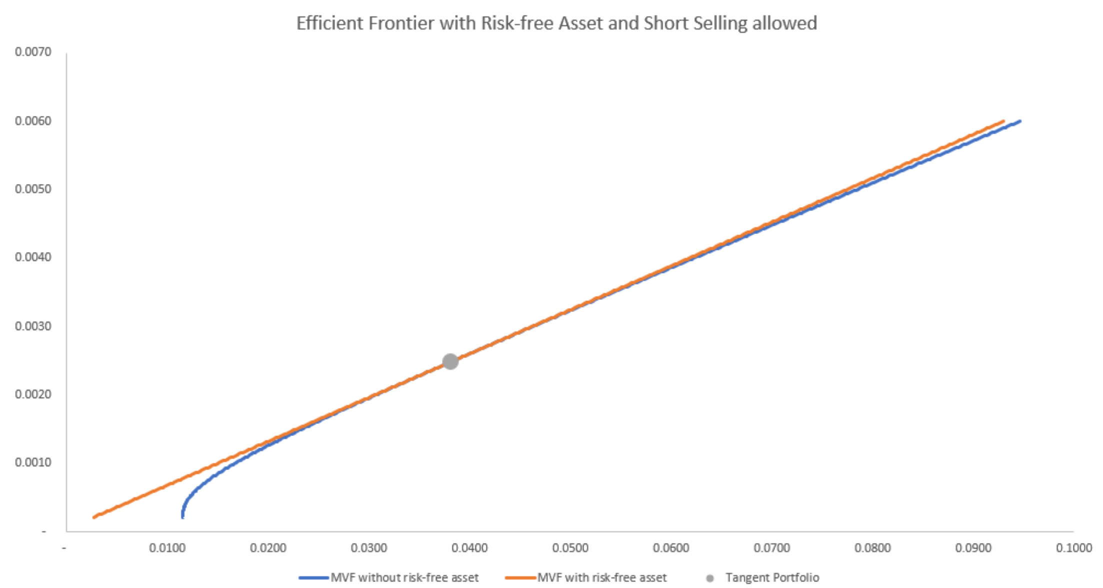


Figure 14: Efficient frontier with risk-free asset and short selling

8. RELATED INFORMATION AND DATA

8.1. DATASET

The raw data files are included together with the Excel file and the documentation

8.2. CHANGE LOG

24/06/2022: v0.1 first establishment of the documentation

APPENDIX AND REFERENCES

REFERENCES

Brooks, C. (2019). Portfolio Theory Using Matrix Algebra. In *Introductory Econometrics for Finance* (4th ed., pp. 82–90). Cambridge University Press.