Report of Lab 2: Decision theory

Alberto Nieto Sandino

November 2, 2018

Corrections for re-submission

All the parts that have been improved based on the comments given or new sections that have been added are written in blue. The Assignment 2 has been added since last resubmission.

1 Introduction

Lab 2 centers around portfolio optimization. The aim of the lab is to divide a certain capital between 7 different stocks (i.e. create a portfolio) in such a way that the profit is maximize while taking into account the risk.

2 Assignment 1.1, Assumptions about the data

2.1 Problem

This first assignment involves an analysis of the data that will be used to create the portfolio. The aim is to get an idea of how accurate the assumptions that are being taken are when observed carefully.

2.2 Theory and implementation

We are given an array of data containing the prices of stocks for different time points. In order to be able to model the behaviour of the stocks, the log-returns $X(t)$ of these prices over time are calculated as

$$X(t) = log(S(t)) - log(S(t-1)), \tag{1}$$

where S(t) corresponds to the price of a stock S at a time t. These log-returns are assumed to be time-independent and identically normally distributed.

In order to check the identically normally distributed assumption, we display the histograms, the qq-plots, and perform a formal goodness-of-fit test (e.g. Chi-squared test) for the log-returns. The histograms (command `histogram`) give us a graphical representation of the sampling distribution, while the qq-plots (command `qqplot`) compares the the log-returns (ith ordered value of $X(i)$) against the theoretical ($\frac{i-0.5}{n}$th) quantile values of a normal distribution. On the other side, the

Chi-squared test (`chi2gof`) gives us a quantitative answer to the null hypothesis of whether the data comes from a normal distribution. It computes the chi-square test statistic

$$\chi^2 = \sum_{i=1}^{N} \frac{(O_i - E_i)^2}{E_i}, \tag{2}$$

where O_i are the observed counts and E_i are the expected counts.

In order to check the time independence assumption, we can try to observe that dependence with the autocorrelation function. This function only allows us to detect linear dependence, thus, non linear dependency cannot be assessed in this method. We implement the function `autocorr` which computes the sample autocorrelation for our stochastic time series, the log-returns. The function measures the correlation between $X(t)$ and $X(t+h)$ with h being the time lag as $r_h = c_h/c_o$, where

$$c_h = \frac{1}{T} \sum_{t=1}^{T-h} (X(i) - \overline{X})(X(i+h) - \overline{X}), \tag{3}$$

with $\overline{X}$ being the sample mean and c_0, the sample variance of the data.

The sample mean (`mean`) is used to estimate the mean for the log-returns as

$$\mu_j = \frac{1}{N} \sum_{i=1}^{N} X_i, \tag{4}$$

where $j = 0, 1, .., N$ is the number of stocks. The sample standard deviation (`std`) is used to estimate the standard deviation of the stocks as

$$\sigma_j = \sqrt{\frac{1}{N-1} \sum_{i=1}^{N} |X_i - \mu_j|^2}. \tag{5}$$

If we want to know if the log-returns are dependent between each others, we can compute the correlation structure between them. The correlation (`corrcoef`), ρ, of two stocks (A, B) measures their linear dependence as

$$\rho(A, B) = \frac{1}{N-1} \sum_{i=1}^{N} \left(\overline{\frac{A_i - \mu_A}{\sigma_A}} \right) \left(\overline{\frac{B_i - \mu_B}{\sigma_B}} \right), \tag{6}$$

with a correlation coefficient matrix as

$$R = \begin{bmatrix} \rho(A, A) & \rho(A, B) \\ \rho(B, A) & \rho(B, B) \end{bmatrix}. \tag{7}$$

The same formulation applies for the correlation of the total amount of stocks.

2.3 Results and discussion

Observing Figure 1, one could argue that the shape of the histograms is approximately similar to that of a normal distribution. The qq-plots in Figure 2 show, however, "thin tails" for all the stocks which point to a distribution with more data concentrated around the mean value and less

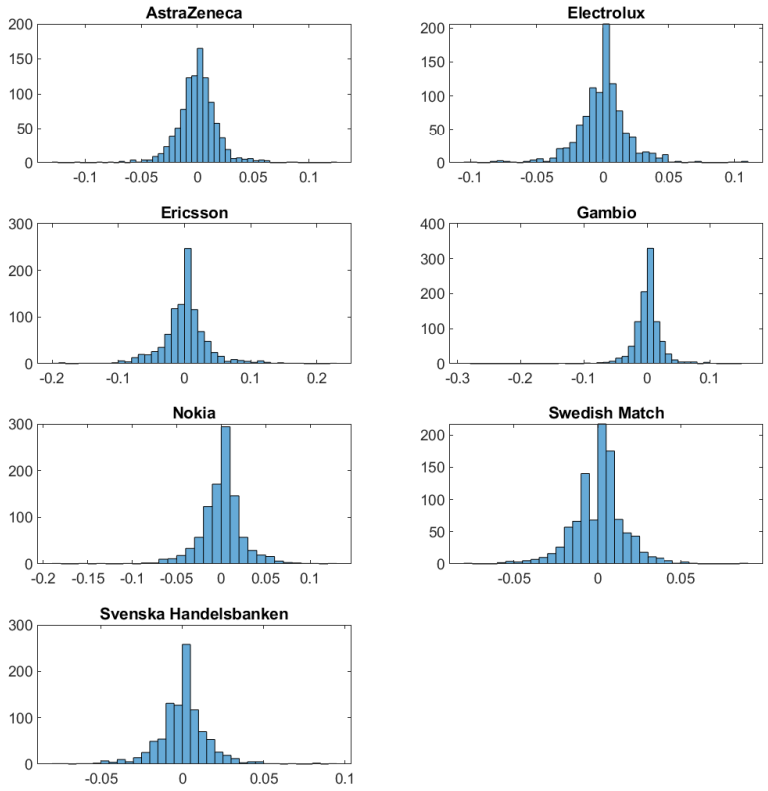

Figure 1: Histogram for the log-returns of the stocks

in the tails compared to that of a normal distribution. These shows that the first quantiles occur at higher than expected values, while the last quantiles occur at lower than expected values.

Looking at the Chi-square goodness-of-fit test results, we get the following p-values seen in Table 1. The values are non-significant by a large margin ($p << 0.05$), thus rejecting the null hypothesis (i.e. data following a normal distribution) with high confidence. However, this measure is not of how far from a normal distribution it is, but of how sure this conclusion is.

	AstraZeneca	Electrolux	Ericsson	Gambio	Nokia	Swedish Match	Svenska Handelsbanken
p	$2.05 \cdot 10^{-20}$	$5.15 \cdot 10^{-27}$	$3.48 \cdot 10^{-47}$	$8.35 \cdot 10^{-28}$	$1.70 \cdot 10^{-26}$	$1.45 \cdot 10^{-13}$	$6.12 \cdot 10^{-34}$

Table 1: p-values of the Chi-square goodness-of-fit test for the stock values

In conclusion, the formal goodness-of-fit test rejects our hypothesis, however by inspecting the histograms and the qq-plots we can see the log-returns can be considered approximately normally distributed as they do not differ greatly from the shape of a normal distribution.

The sample autocorrelation for the log-returns can be seen in Figure 3. It shows little to no degree of autocorrelation between the samples. This validates our assumption of a time-independent

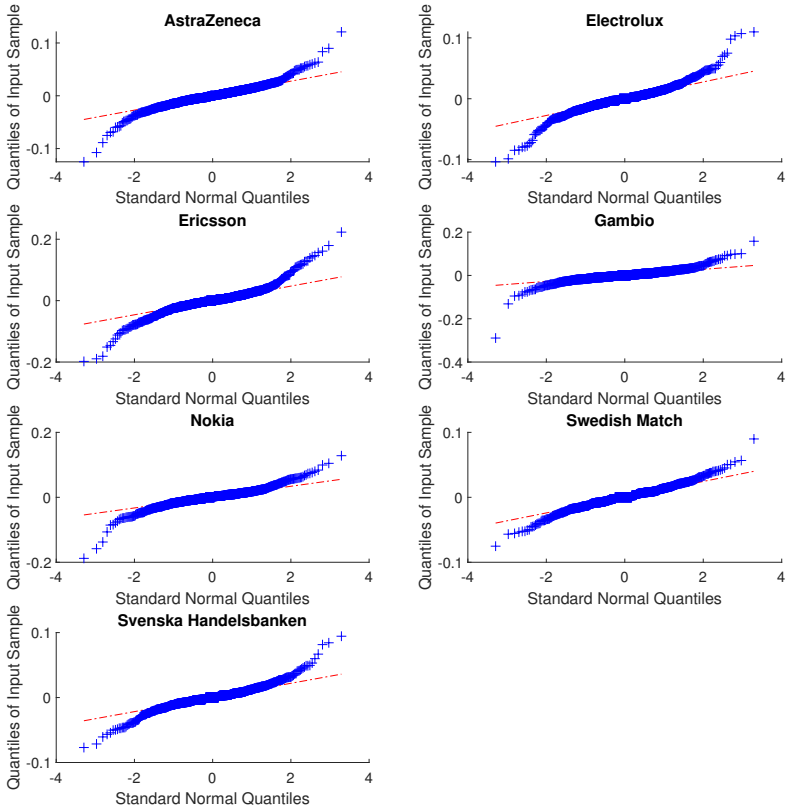

Figure 2: QQ-plots for the log-returns of the stocks

data which can be approximated as a stationary process.

The sample autocorrelation for the absolute value of the log-returns (seen in Figure 4) shows a different picture. The figure shows a significant positive autocorrelation which decays slowly with time. This is a well-known property of financial data called volatility clustering, meaning that high returns are likely followed by high returns while low returns are likely followed by low returns.

The sample mean and sample standard deviation can be found in Table 2. The correlation structure for the seven stocks can be found in equation 8. When inspecting the matrix, it is considered that there is a significant correlation between the stocks since the values in the correlation matrix are non-zero. Thus, it can be concluded they are dependent of each other unlike it was assumed initially. This information should be taken into account when analysing the final results of the portfolio.

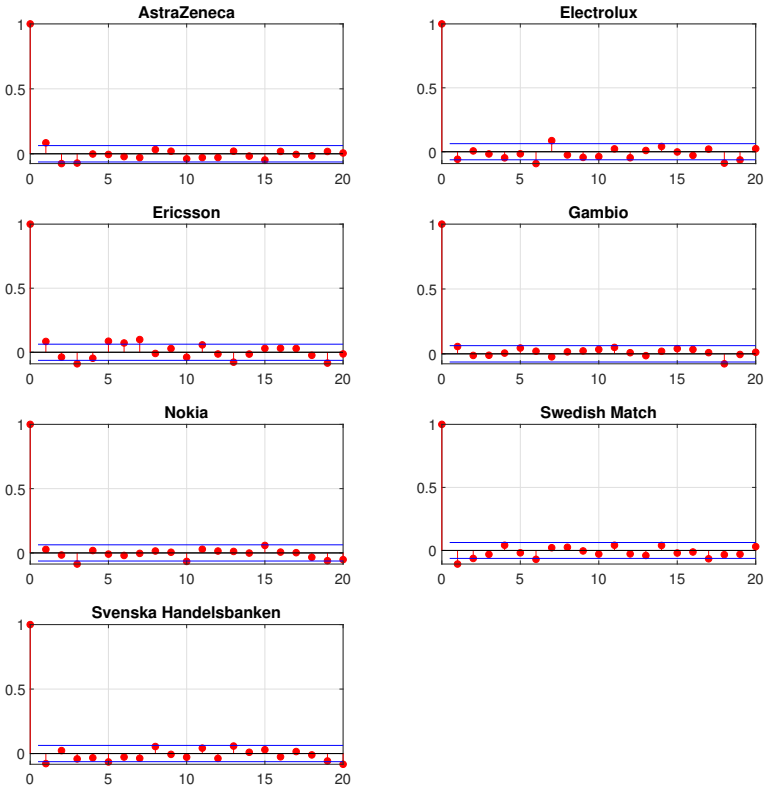

Figure 3: Sample autocorrelation function computed for the log-returns of the stocks

$$R = \begin{bmatrix} 1.000 & 0.265 & 0.276 & 0.191 & 0.363 & 0.126 & 0.203 \\ 0.265 & 1.000 & 0.367 & 0.223 & 0.375 & 0.120 & 0.451 \\ 0.276 & 0.367 & 1.000 & 0.231 & 0.523 & 0.074 & 0.409 \\ 0.191 & 0.223 & 0.231 & 1.000 & 0.199 & 0.061 & 0.217 \\ 0.363 & 0.375 & 0.523 & 0.199 & 1.000 & 0.064 & 0.386 \\ 0.126 & 0.120 & 0.074 & 0.061 & 0.064 & 1.000 & 0.185 \\ 0.203 & 0.451 & 0.409 & 0.217 & 0.386 & 0.185 & 1.000 \end{bmatrix} \tag{8}$$

All in all, it is reasonable to assume that the log-returns are i.i.d. normal *in time*. The data does not fit perfect to the assumptions, but they are close enough to be approximated as such. This discrepancies should be kept in mind when extracting conclusions from the portfolio optimization since the results will be subjected to a certain small error that can, for the most part, be ignored.

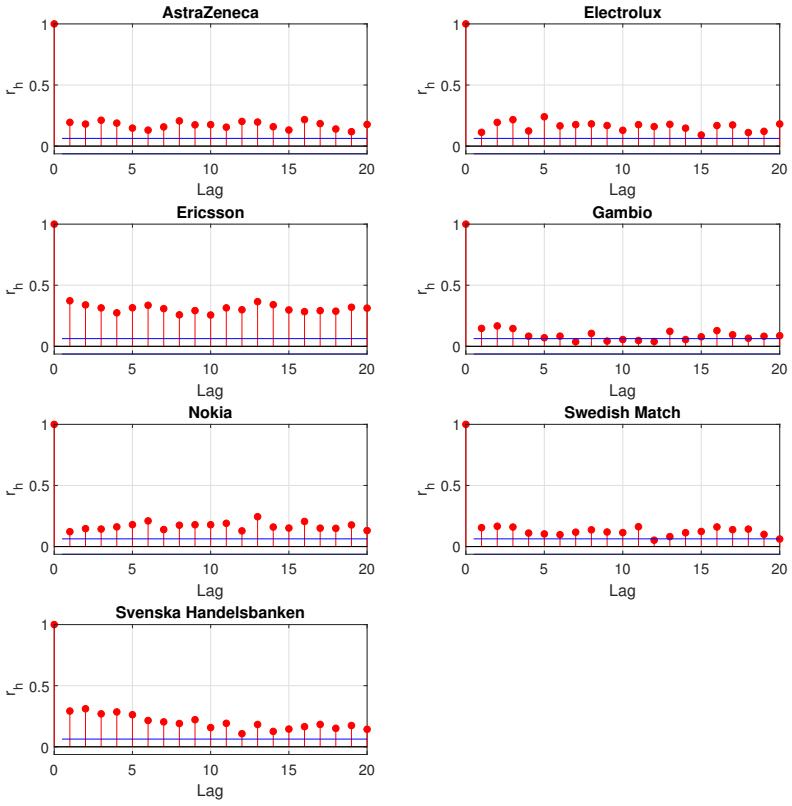

Figure 4: Sample autocorrelation function computed for the log-returns of the stocks

3 Assignment 1.2, Utility and expected utility

3.1 Problem

In this section, we explore how the chosen utility function behaves for different risk behaviours and we take a first look at how the expected utility for investing in each stock looks.

3.2 Theory and implementation

The utility function $u(y)$ is a way of defining the gain we will obtain for a certain outcome. We can take different approaches based on the level of risk we want to accept: risk averse, risk neutral and risk seeking. In our case, we focus our study in the utility functions of the form

$$u(y) = 1 - e^{-ky}, \tag{9}$$

with $k > 0$. The expected utility function $U_\pi(w)$ will give the expected gain from the utility function for a given probability distribution $\pi(y)$ describing the possible states (i.e. the stock values), that

	AstraZeneca	Electrolux	Ericsson	Gambio	Nokia	Swedish Match	Svenska Handelsbanken
μ	$-9.58 \cdot 10^{-05}$	0.000112	0.000390	0.000527	0.000178	0.000386	0.000311
σ	0.0188	0.0202	0.0378	0.0238	0.0250	0.0156	0.0158

Table 2: Sample mean (μ) and standard deviation (σ) for the stock values

is

$$U_\pi(w) = \mathbf{E}_Y[u(y)] = \int u(y)\pi(y)dy = \int (1 - e^{-ky}) \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-(y-w^T\mu)^2}{2\sigma^2}} dy, \tag{10}$$

where $\pi(y)$ is modelling the joint multivariate normal distribution of the seven stocks, based on the assumptions of them being time-independent and normally distributed. This distribution comes from the distribution $X \sim N(\mu, \Sigma)$ where $\mu = \left[\mu_1, ..., \mu_7\right]^T$ (command `mean`) and Σ is the covariance matrix of the stocks with $\sigma_{ij} = \text{Cov}(X_i, X_j)$ (command `cov`). This distribution is linearly transformed to obtain the one used in our portfolio $Y \sim N(w^T\mu, w^T\Sigma w)$. Finally, the integral is solved in Matlab by Quadrature (`quad`) which is a numerical method that computes the definite integral for the given function.

3.3 Results and discussion

The utility function is plotted in Figure 5 for different risk behaviours. It can be seen that as the smaller the k value, the more neutral the behaviour is towards risk. However, as k increases, the attitude becomes more risk averse.

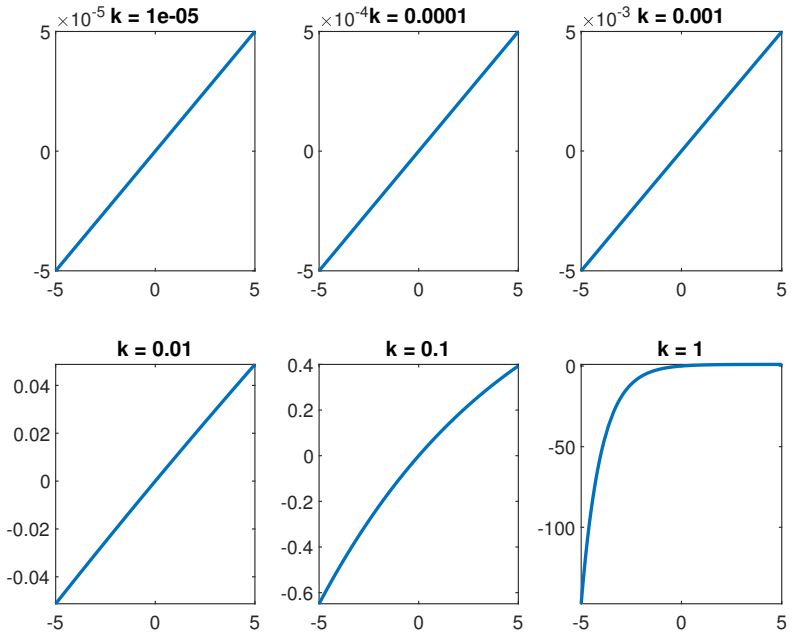

Figure 5: Utility function for different risk behaviours (k values)

Now, the expected return is calculated in the case of investing all the money in one stock for different levels of risk aversion. The result is plotted in Figure 6. For the most neutral behaviour ($k = 0.0001$), the best choice is to invest in the Gambio stock (number 4). As the behaviour becomes more risk averse, Gambio keeps being the best option for $k = [0.0001, 0.01, 0.1]$. Except for $k = 0.001$ where Ericsson (number 3) is a better option, although it would not be recommended to invest since the expected utility is negative. For the most risk averse case ($k = 1$), Swedish Match (number 6) would be the first option followed closely by Gambio.

Considering the results, it seems Gambio is the likely candidate to choose no matter the risk that is chosen. This can be explained by looking at the expected values and covariances matrices for each stock investment. It has the highest expected value, while not having a big variance which makes it a very promising candidate for a good investment.

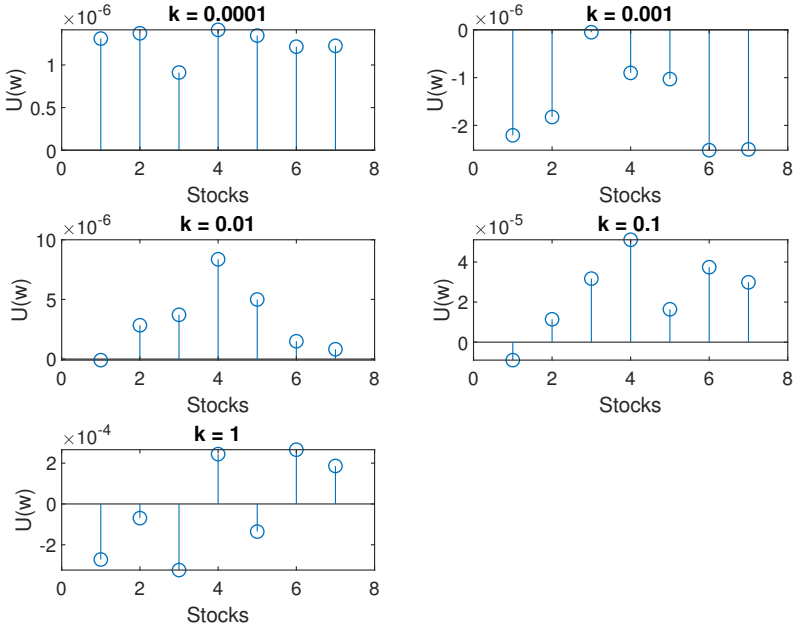

Figure 6: Expected utility when investing only in one stock each time for different behaviours towards risk

4 Assignment 2

4.1 Problem

The expected utility of the portfolio,

$$U_\pi(w) = \int_{-\infty}^{\infty} (1 - e^{-kx}) \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-z)^2}{2\sigma^2}} dx, \tag{11}$$

with the constraints of $w_1, ..., w_n \geq 0$ and $\sum_{i=1}^{n} w_i = 1$ where $\sigma^2 = w_T \sum w$ and $z = \mu^T w$, can be maximized to simplify it. We can show that maximizing the equation above is in fact equivalent

to maximizing the following equation,

$$\mu^T w - \frac{k}{2} w^k \sum w. \tag{12}$$

4.2 Solution

We start with multiplying the utility function with the probability distribution function,

$$U_\pi(w) = \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-z)^2}{2\sigma^2}} dx - \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}\sigma} e^{-kx} e^{-\frac{(x-z)^2}{2\sigma^2}} dx. \tag{13}$$

The first integral is over a normally distributed function, thus it is equal to one. We can then re-write the remaining integral as:

$$-kx - \frac{(x-z)^2}{2\sigma^2} = -\frac{-(x^2 + 2kx\sigma^2 - 2xz + z^2)}{2\sigma^2}, \tag{14}$$

If we expand the equation, we get that

$$\frac{-(x-(z-k\sigma^2))^2 - 2kz\sigma^2 + k^2\sigma^2}{2\sigma^2} = -\frac{(x-(z-k\sigma^2))^2}{2\sigma^2} - k\Big(z - \frac{k}{2}\sigma^2\Big). \tag{15}$$

Finally, we arrive at:

$$U_\pi(w) = 1 - e^{-k(z-\frac{k}{2}\sigma^2} \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{(z-(z_k\sigma^2))^2}{2\sigma^2}} dx, \tag{16}$$

where the integral is over a normally distributed function so it is equal to one, except for the term $z - k\sigma^2$. Thus,

$$1 - e^{-k\left(z-\frac{k}{2}\sigma^2\right)}. \tag{17}$$

Thus, it has been proven that maximizing equation (11) is equivalent to maximizing equation (12).

5 Assignment 3 and 4. Optimization with stocks

5.1 Problem

In this part of the assignment, we examine the optimization of the portfolio (i.e. of the weights) for either two (Assignment 3) or all of the stocks (Assignment 4). The aim is to find the optima weights that will optimize the expected utility and observe the conclusions that can be extracted.

5.2 Theory and implementation

In this case, we only make use of the stocks from Gambio and Ericsson, thus the mean and covariance matrix of the log-returns only involve these two stocks. Using this two stocks, we can observe where the maximum expected value lies for a combination of different weights for the two stocks. Then,

the expected utility is computed for every pair of weights by integrating (`quad`) expression (10) over a finite interval in order to see where it reaches a maximum value.

This process can be automated and calculated more precisely by `fmincon`. This function is able to solve the maximization problem concerning the expected value which is equivalent to maximizing the quadratic problem

$$\mu^T w - \frac{k}{2} w^T \Sigma w, \tag{18}$$

with the constraints such as $w_1, ..., w_7 \geq 0$ and $\sum_{i=1}^{7} = 10$ as is proven in Assignment 2 (missing from this version).

5.3 Results and discussion

Figure 7 shows the values of the expected utility for a pair of stocks given different combinations of weights for $k = 1$. The maximum expected utility can be found approximately where $w_1 = 0.14$ and $w_2 = 0.86$, that is when most of the money is invested in Gambio.

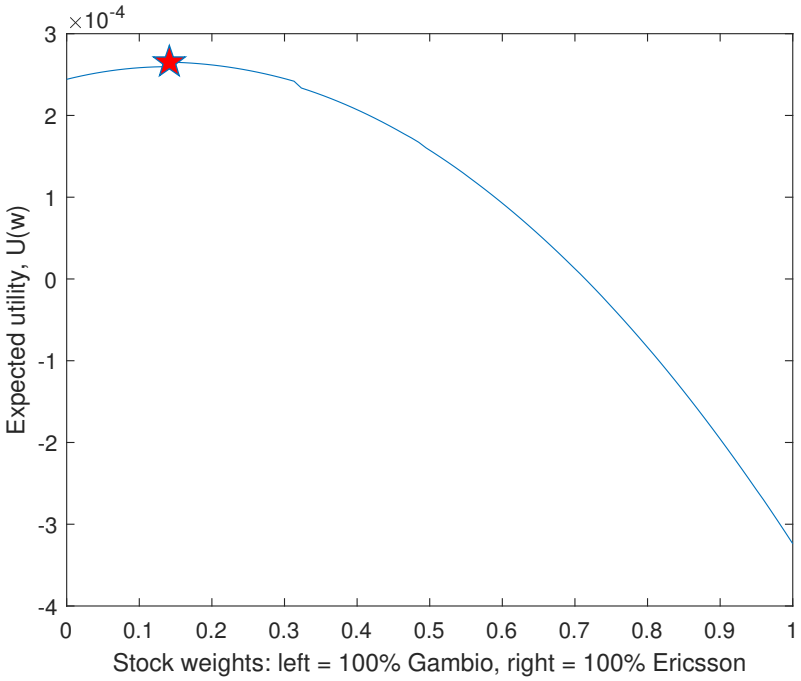

Figure 7: Evolution of the expected utility for two stocks

In Figure 8, the optimal weights for the two stocks are shown as a function of k. The most risk neutral the behaviour is (i.e. smaller k), the bigger weight for Gambio (2nd stock). This can be explained by looking at expression (18): the first term represents the mean of the expected returns which is given more relative importance by scaling down (i.e. small k's) the second term which constitutes the risk (i.e. covariance). The bigger k gets, the more risk averse the estimation gets. This leads to the increase on the Ericsson weights (1st stock). By doing a small sacrifice in the expected returns, the risk is decreased because of the diversification of the stocks.

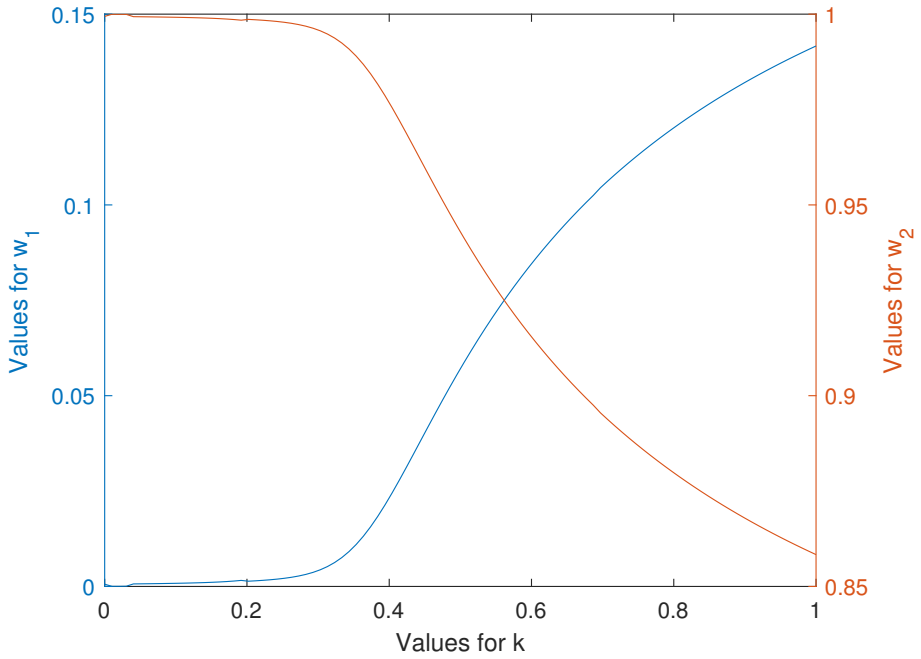

Figure 8: Evolution of the optimal weights for two stocks as a function of k

Now that the solutions for one pair of stocks have been detailed, we can proceed to the optimization of the whole portfolio with all seven stocks. The optimal weights for each k can be seen in Figure 9, while the expected utility for each value of k with the optimal weights is found in Figure 10. In Figure 10, it can also be found how the expected utility would look for the case of a naive investment (i.e. equally divided) or going for putting everything into one stock (in this case Gambio, since it has been seen to be the most profitable in previous sections).

As happened in the optimization of two stocks, in Figure 9 we see how most of the weight for a neutral behaviour goes to the stocks with the biggest expected returns. While as k increases, it gets a bit more spread throughout the different stocks to reduce the risk.

This can also be seen in Figure 10, since as k increases we see that the optimal weights have a bigger expected value because of the risk consideration. We can see, as well, how investing naively will be the worst possible scenario since any sort of criteria was used to decide the investment.

Appendix - Matlab code

```
%% Lab 2, created by Alberto Nieto Sandino

%% Assignment 1.1 - Assumptions about the data
clear all; close all; clc

% Read the data with stock values
load('stockdata.tsv');
```

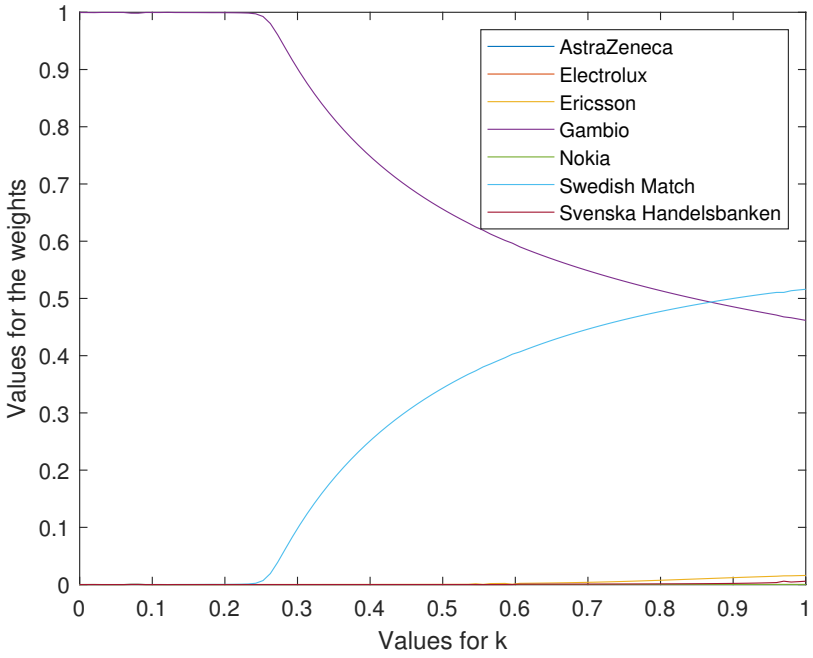

Figure 9: Evolution of the optimal weights for the stocks as a function of k

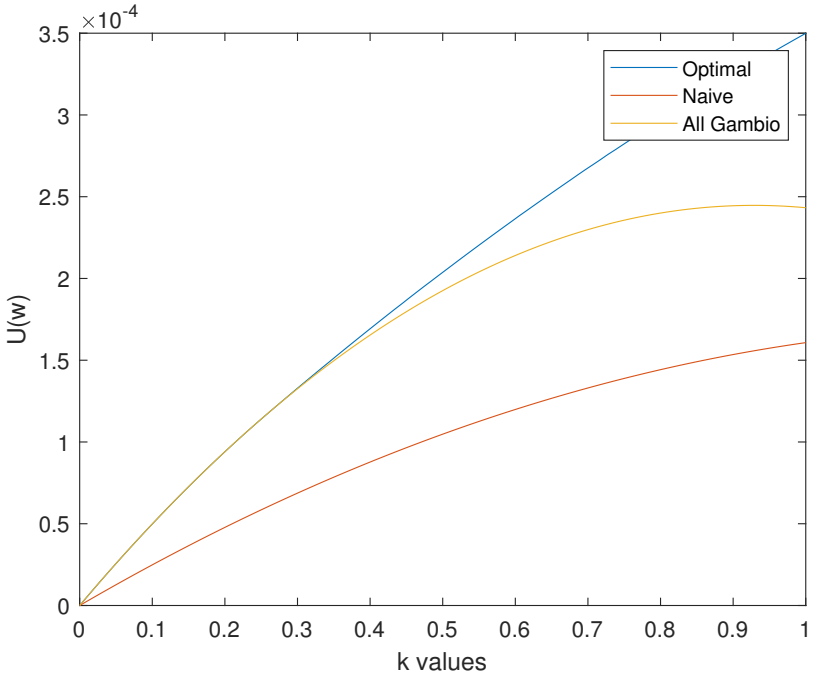

Figure 10: Comparison between the expected utility for the optimal weights, the "naive" weights, and opting for one stock only (Gambio) as a function of k

```
% Calculate the log-returns
X = zeros(size(stockdata,1)-1,size(stockdata,2)-1);
for i = 2:size(stockdata,1)
    X(i,:) = log(stockdata(i,2:8)) - log(stockdata(i-1,2:8));
end

% Create cell array for stock names
stocks = {'AstraZeneca','Electrolux','Ericsson','Gambio','Nokia',...
          'Swedish Match','Svenska Handelsbanken'};

% Plot log of stock values over time
figure();
for j = 1:7
    subplot(4,2,j)
    plot(stockdata(:,1), X(:,j));
    xlim([min(stockdata(:,1)) max(stockdata(:,1))])
    title(stocks(j));
    xlabel('time (days)');
    ylabel('X(t)');
end

% Plot the histograms and qq-plots for every stock value
figure(); % Histograms
for k = 1:7
    subplot(4,2,k)
    histogram(X(:,k));
    title(stocks(k));
end

figure()
for k = 1:7
    subplot(4,2,k)
    qqplot(X(:,k));
    title(stocks(k));
end
clear i j k

% Formal goodness of fit for each stock
h = zeros(7,1);
p = zeros(7,1);
for i = 1:7
    [h(i), p(i), stats(i)] = chi2gof(X(:,i),'nbins',20);
end

% Normal assumptions for the log-returns do not seem possible since the
% goodness of fit rejects the null hypothesis.

% Estimate the autocorrelation function of log-returns and its absolute
% values
figure()
for j = 1:7
    subplot(4,2,j)
    autocorr(X(:,j));
    title(stocks(j));
%     ylabel('r_h');
```

```
end

figure()
for j = 1:7
    subplot(4,2,j)
    autocorr(abs(X(:,j)));
    title(stocks(j));
    ylabel('r_h');
end
% The sample ACF has significant autocorrelation for some of the lags in
% all the stocks.

% Estimate the mean and standard deviation of the log-returns
mu = zeros(7,1);
sd = zeros(7,1);
for k = 1:7
    mu(k) = mean (X(:,k));
    sd(k) = std (X(:,k));
end

% Estimate the correlation between the log-returns
clear i j k
r_sign = 0.8; % Minimum value of the correlation coeficient to be
% considered significant
R = corrcoef(X);
for i = 1:7
    for j = 1:7
        if i~=j
            r_temp = R(i,j);
            if r_temp > r_sign
                disp (['Correlation between stocks ' num2str(i)...
                    ' and ' num2str(j) ' is likely,'...
                    'correlations coefficients are significant']);
            else
                disp (['Correlation between stocks is ' num2str(i)...
                    ' and ' num2str(j) ' unlikely,'...
                    'correlations coefficients are not significant']);
            end
        end
    end
end

% All the correlation coeficients are not significant, thus we can conclude
% it is unlikely that there is a linear dependency between the stock
% values.
% It's expected that they're not correlated since they change based on
% different basis depending on how each company is doing.


%% Assignment 1.2 - Utility and expected utility

close all;

% Explore the utility function for different k's
k = 0.00001;
figure()
```

```
for i = 1:6
    u = @(x)(1 - exp(-k*x));
    subplot(2,3,i);
    fplot(u,'linewidth',1.5);
    title(['k = ' num2str(k)]);
    k = k*10;
end

% The smaller the k, the more neutral the behaviour. The bigger the k, the
% more risk averse it is.

clc;
w = [1; zeros(6,1)]; % Considering 1st stock at a time
mu = mean (X).'; % Mean vector of log-returns
cov_mat = cov(X); % Covariance matrix of log-returns

z = mu.'*w; % Mean of 1st stock
sigma = sqrt(w.'*cov_mat*w); % Standard deviation for 1st stock

clear x
% Expected utility for stock nbr 1
% x = X(:,1); % utility
k = 1;
u = @(x)(1 - exp(-k*x));
pi_x = @(x)(1/(sqrt(2*pi)*sigma)).*exp(-((x-z).^2/(2*sigma.^2)));
y = @(x)((1 - exp(-k*x)).*(1/(sqrt(2*pi)*sigma)).*exp(-((x-z).^2/(2*sigma.^2))));

figure();
subplot(2,2,1:2)
    fplot(pi_x,[-1 1],'g');
    title('\pi_x(y)');
subplot(2,2,3)
    fplot(u,[-1 1],'b');
    title('u(y)');
subplot(2,2,4)
    fplot(y,[-1 1], 'r');
    title('\pi_x(y)*u(y)');

U = zeros(7,5);
for i = 1:7 % Loop thru stocks
    k = 0.0001;
    w = zeros(7,1);
    w(i) = 1; % Select each stock
    z = mu.'*w; % Mean of each stock
    sigma = sqrt(w.'*cov_mat*w); % Standard deviation for each stock
    for j = 1:5 % Loop thru k's
        y = @(x) u_AND_pi (x, k, z, sigma);
        U(i,j) = quad(y, min(X(:)), max(X(:)));
        k = k*10;
    end
end

figure();
mesh(U);
xlabel('k values');
ylabel('Stocks');
```

```
zlabel('Expected utility, U(w)');

figure()
k = 0.0001;
for i = 1:5
    subplot(3,2,i)
    stem(U(:,i))
    title(['k = ' num2str(k)]);
    k = k*10;
    xlabel('Stocks');ylabel('U(w)');
end

% For the most neutral behaviour, the best choice would be to invest in
% stock #4. As the behaviour gets more averse to risk #4 keeps being the
% best option for k = {0.0001, 0.01, 0.1}, except for k = 0.001 were #3 is
% better. For the most risk averse case (k=1), the most recommended would
% be #6 with #4 following closely.

% k = 0.01 gives the highest results considering all stocks.

%% Assignment 3 - Optimization of two stocks

clear all; close all; clc

% Read the data with stock values
load('stockdata.tsv');

% Calculate the log-returns
X = zeros(size(stockdata,1)-1,size(stockdata,2)-1);
for i = 2:size(stockdata,1)
    X(i,:) = log(stockdata(i,2:8)) - log(stockdata(i-1,2:8));
end

% Take out the stock values of Ericsson (#3) and Gambio (#4)
X = X(:,3:4);

% Estimate the mean vector and the covariance matrix
mu = mean (X).'; % Mean vector of log-returns
cov_mat = cov(X); % Covariance matrix of log-returns

% Calculate the expected utility
k = 1;
w_1 = linspace(0, 1);
w_2 = 1 - w_1;
w = [w_1; w_2]; % Matrix of every pair of weights
U = zeros(length(w), 1);

for i = 1:length(w)
    z = mu.'*w(:,i); % Mean of i-th row
    sigma = sqrt(w(:,i).'*cov_mat*w(:,i)); % Standard deviation for i-th row
    y = @(x) u_AND_pi (x, k, z, sigma);
    U(i) = quad(y, min(X(:)), max(X(:)));
end

% Plot U(w) against w_1
figure()
```

```
% plot(w_1, U)
max_U = find(U == max(U));
plot(w_1, U,'-p','MarkerIndices',max_U, 'MarkerFaceColor','red',...
    'MarkerSize',15)
xlabel('Stock weights: left = 100% Gambio, right = 100% Ericsson');
ylabel('Expected utility, U(w)');


% Find the optimum point more accurately with fmincon
k = 1; % we keep k = 1, for now
clear w

% x0 = [0.15; 0.85];
x0 = [0.2; 0.8];% Starting point in approx. solution
A = [];
b = [];
Aeq = [1,1];
beq = 1;
lb = [0, 0];
ub = [1, 1];

% k = 0.00001;
% The solver fmincon will try to minimize the function, thus we try to
% minimize the opposite of the Eq. 2.2 which we want to maximize


fun = @(w) maximizefun(w, k, mu, cov_mat);
options.StepTolerance = 1e-20;
optimum_w = fmincon(fun,x0,A,b,Aeq,beq,lb,ub);

% Evaluate the expected utility with the optimal weights
z = mu.'*optimum_w; % Mean of i-th row
sigma = sqrt(optimum_w.'*cov_mat*optimum_w); % Standard deviation for i-th row
y = @(x) u_AND_pi (x, k, z, sigma);

% When using the whole function it seems the mean is too powerful and seems
% to be where most of the influence lies.

% Using fmincon on function fun tries to maximize the expected return while
% minimizing the risk (standard deviation) of the investment. The risk can
% be decreased with only a small sacrifice of the return and that is what
% the algorithm does. It gives prevalence to the stock with higher expected
% return while diversificating, thus reducing the risk.

% The bigger the k, the more importance is given to the minimization of the
% risk (standard deviation), thus the investment should be more diverse
% with more equal weights.


% Find the optimal weights for several different k's
clear i k
k = linspace (0.00001, 1);
optima_w = zeros(2,100);
for i = 1:length(k)
    fun = @(w) maximizefun(w, k(i), mu, cov_mat);
    optima_w(:,i) = fmincon(fun,x0,A,b,Aeq,beq,lb,ub,[],options);
```

```
end

close all
figure()
yyaxis left
% title('Evolution of weights for different values of k');
plot(k, optima_w(1,:))
ylabel('Values for w_1')

yyaxis right
plot(k, optima_w(2,:))
xlabel('Values for k');
ylabel('Values for w_2')

figure()
risk = zeros(100,1);
for i = 1: size(optima_w, 2)
    risk (i) = optima_w(:,i)' * cov_mat * optima_w(:,i);
end
plot(k,risk)
title('Risk over k values')
xlabel('Values for k');
ylabel('Risk');

% Plotting the risk over the k values, we can see that as k increases, the
% risk aversion of the investment increases and the risk decreases as a
% consequence of our decision.

%% Assignment 4 - Optimization with 7 stocks

clear all; close all; clc

% Read the data with stock values
load('stockdata.tsv');

% Calculate the log-returns
X = zeros(size(stockdata,1)-1,size(stockdata,2)-1);
for i = 2:size(stockdata,1)
    X(i,:) = log(stockdata(i,2:8)) - log(stockdata(i-1,2:8));
end

% Estimate the mean vector and the covariance matrix
mu = mean (X).'; % Mean vector of log-returns
cov_mat = cov(X); % Covariance matrix of log-returns

% Find the optimum point more accurately with fmincon

x0 = rand(7,1);
x0 = x0/sum(x0);
A = [];
b = [];
Aeq = ones(1,7);
beq = 1;
lb = zeros(1,7);
ub = ones(1,7);
```

```
% Define the function to maximize
fun = @(w) maximizefun(w, k, mu, cov_mat);

% Find the optimal weights for several different k's
clear i k
k = linspace (0.00001, 1);
optima_w = zeros(7,100);
fval = zeros(1,100);
for i = 1:length(k)
    fun = @(w) maximizefun(w, k(i), mu, cov_mat);
    [optima_w(:,i),fval(i)] = fmincon(fun,x0,A,b,Aeq,beq,lb,ub);
end

% Plot the final results of the weihgts
figure()
title('Evolution of weights for different values of k');
plot(k, optima_w)
ylabel('Values for the weights')
xlabel('Values for k');
legend('AstraZeneca','Electrolux','Ericsson','Gambio','Nokia','Swedish Match','Svenska Handelsbanken

% Calculate the expected utility
for i = 1:length(optima_w)
    z = mu.'*optima_w(:,i); % Mean of i-th row
    sigma = sqrt(optima_w(:,i).'*cov_mat*optima_w(:,i)); % Standard deviation for i-th row
    y = @(x) u_AND_pi (x, k(i), z, sigma);
    U(i) = quad(y, min(X(:)), max(X(:)));
end

% Calculate the expected utility for a naive case
naive_w = (1/7)*ones(7,100);
for i = 1:length(naive_w)
    z = mu.'*naive_w(:,i); % Mean of i-th row
    sigma = sqrt(naive_w(:,i).'*cov_mat*naive_w(:,i)); % Standard deviation for i-th row
    y = @(x) u_AND_pi (x, k(i), z, sigma);
%     U(i) = quad(y, min(X(:)), max(X(:)));
    U_naive(i) = integral(y, -inf, inf);
end

% Calculate the expected utility for a all-in Gambio
allin_w = zeros(7,100);
allin_w(4,:) = 1;
for i = 1:length(allin_w)
    z = mu.'*allin_w(:,i); % Mean of i-th row
    sigma = sqrt(allin_w(:,i).'*cov_mat*allin_w(:,i)); % Standard deviation for i-th row
    y = @(x) u_AND_pi (x, k(i), z, sigma);
    U(i) = quad(y, min(X(:)), max(X(:)));
end

figure()
plot(k, U)
hold on
plot(k,U_naive)
plot(k,U_allin)
legend('Optimal','Naive','All Gambio')
xlabel('k values')
```

```
ylabel('U(w)')
```