**Note:** You are **not** expected to know how to implement these exercises, but to try to implement those yourself following the material and examples from the previous exercise. Doing so will set you to success for the upcoming assignments.

## Problem 2.1: Convexity Properties of Sets

(a) Let $\{S_i\}_{i\in M}$ be a collection of $M = \{1, \ldots, m\}$ convex sets in $\mathbb{R}^n$. Show that their intersection $S = \cap_{i\in M} S_i$ is also convex.

(b) Let $S_1$ and $S_2$ be closed convex sets in $\mathbb{R}^n$. Show that their Minkowski sum

$$S = S_1 + S_2 = \{x + y : x \in S_1, y \in S_2\}$$

is also convex. Also, show by an example that $S_1 + S_2$ is not necessarily closed.

## Problem 2.2: Weierstrass' Theorem

Consider the following nonlinear optimisation problem $P$:

$$\begin{aligned}(P):\ \max_{x,y}\ & \frac{1}{x+y}\\ \text{subject to: }\ & xy \geq 1\\ & x, y \geq 0\end{aligned}$$

(a) Show that $P$ has a solution by applying Weierstrass' theorem.

(b) Model the problem $P$ with `JuMP` and try to find the global maximum.

## Problem 2.3: Portfolio Optimization

For this problem, use the data file prices.csv which contains daily prices of $N = \{1, \ldots, n\}$ stocks over a time period of $T = \{1, \ldots, m\}$ days. Let $x_i \geq 0$ denote the (long) position of stock $i \in N$ in a portfolio throughout the time period. The positions $x = (x_1, \ldots, x_n)$ in the portfolio are scaled to represent fractions of the total investment, that is,

$$\sum_{i\in N} x_i = 1$$

Let $p_i^t$ denote the daily price of stock $i \in N$ for all $t \in T$, and let $r_i^t$ be the relative daily return of stock $i \in N$ for all $t \in T \setminus \{m\}$. These are computed as

$$r_i^t = \frac{p_i^{t+1} - p_i^t}{p_i^t}, \quad \forall i \in N, \forall t \in T \setminus \{m\}$$

Let $\mu = (\mu_1, \ldots, \mu_n)$ denote the *expected relative returns* of the stocks $N$, and let $\Sigma \in \mathbb{R}^{n\times n}$ be the corresponding covariance matrix. Thus, the expected average return and variance of a portfolio $x = (x_1, \ldots, x_n)$ are $\mu^\top x$ and $x^\top \Sigma x$, respectively. Moreover, let $\sigma \in \mathbb{R}^n$ be the standard deviation vector and $\rho \in \mathbb{R}^{n\times n}$ the correlation matrix of the relative stock returns.

(a) Read the data and plot the price curves of each stock for the whole time period.

(b) Compute the expected average returns $\mu$, the covariance matrix $\Sigma$, the correlation matrix $\rho$, and the standard deviation vector $\sigma$ using the `Julia` package `Statistics`.

(c) Sort the stocks in increasing order with respect to their expected returns. Using this order, plot the expected returns $\mu_i$ and standard deviations $\sigma_i$ of each stock $i \in N$ in two different plots but in the same figure. Look at Exercise 1.1 code for reference how to plot multiple plots in the same figure using the `Plots` package. **Note:** plots might not appear in Jupyter notebooks unless they are called at the last line of a cell. However, you can always save the most recent plot as a `pdf` file, for example, by calling the function `savefig("myplot.pdf")`.

(d) Using the same order as in (c), visualize the correlation matrix $\rho$ using the `PyPlot` package function `imshow`, and make a `scatter` plot of the the stocks' expected returns vs. their standard deviations, i.e., plot the points $(\sigma_i, \mu_i)$, for all $i \in N$. **Note:** to save the correlation plot as a `pdf` file, you have to call `PyPlot.savefig("corrplot.pdf")` explicitly so that `Julia` knows which plotting library was used. This is needed because `Plots` and `PyPlot` both define this function with identical name and parameter types.

(e) Consider the following portfolio optimization problem

$$\begin{aligned} \min_{x} \quad & x^\top \Sigma x && (1) \\ \text{subject to:} \quad & \mu^\top x \geq \mu_{min} && (2) \\ & \sum_{i \in N} x_i = 1 && (3) \\ & x \geq 0 && (4) \end{aligned}$$

where the objective is to minimise the portfolio variance (i.e., risk) $x^\top \Sigma x$ by satisfying a minimum expected return constraint (2). Model the problem (1) – (4) using `JuMP` and solve the problem with different values of $\mu_{min}$. Use the `Plots` function `bar` to plot fractions of capital invested in each stock in the resulting portfolio. You can try values of $\mu_{min}$ between

$$0 \leq \mu_{min} \leq 0.000869.$$

(f) Compute the optimal portfolio with 50 different values of $\mu_{min}$ between $[0, 0.000869]$ and plot the optimal trade-off curve, i.e., the expected returns or each portfolio as a function of their standard deviations. Also, plot the points $(\sigma_i, \mu_i)$, for all $i \in N$, in the same figure for comparison using the function `scatter!` from the `Plots` package.