1. *Electric vehicle charging.* A group of N electric vehicles need to charge their batteries over the next T time periods. The charging energy for vehicle i in period t is given by $c_{t,i} \geq 0$, for $t = 1, \ldots, T$ and $i = 1, \ldots, N$. In each time period, the total charging energy over all vehicles cannot exceed $C^{\max}$, *i.e.*, $\sum_{i=1}^{N} c_{t,i} \leq C^{\max}$ for $t = 1, \ldots, T$.

   The state of charge for vehicle i in period t is denoted $q_{t,i} \geq 0$. The charging dynamics is
$$q_{t+1,i} = q_{t,i} + c_{t,i}, \quad t = 1, \ldots, T, \quad i = 1, \ldots, N.$$
   Note that $q_{t,i}$ is defined for $t = T+1$. The initial vehicle charges $q_{1,i}$ are given. The charging energy and state of charge are given in kWh (kilowatt-hours).

   The vehicles have different preferences for how much charge they acquire over time. This is expressed by a target minimum charge level over time, given by $q^{\text{tar}}_{t,i} \in \mathbf{R}_+$, $t = 1, \ldots, T+1$. These are nondecreasing, *i.e.*, $q^{\text{tar}}_{t+1,i} \geq q^{\text{tar}}_{t,i}$ for $t = 1, \ldots, T$, $i = 1, \ldots, N$. The charging shortfall in period t for vehicle i is given by
$$s_{t,i} = (q^{\text{tar}}_{t,i} - q_{t,i})_+, \quad t = 1, \ldots, T+1, \quad i = 1, \ldots, N,$$
   where $(a)_+ = \max\{a, 0\}$. Our objective is to minimize the mean square shortfall, given by
$$S = \frac{1}{(T+1)N} \sum_{t=1}^{T+1} \sum_{i=1}^{N} s_{t,i}^2.$$
   This is the same as minimizing the root-mean-square (RMS) shortfall, given by $\sqrt{S}$ (which has units of kWh).

   Explain how to solve the problem using convex optimization, and solve the following problem instance. We have $N = 4$ vehicles, $T = 90$ time periods, and $C^{\max} = 3$. The initial charges $q_{1,i}$ are 20, 0, 30, and 25, respectively. The target minimum charge profiles have the form
$$q^{\text{tar}}_{t,i} = \left(\frac{t}{T+1}\right)^{\gamma_i} q^{\text{des}}_i, \qquad t = 1, \ldots, T+1, \quad i = 1, \ldots, N,$$
   with γ values $0.5, 0.3, 2.0, 0.6$ and q^{des}_i values $60, 100, 75, 125$. Note that q^{des}_i gives the final value of the target minimum charge level for vehicle i, and the parameter γ_i sets the 'urgency' of charging, with smaller values indicating more urgency, *i.e.*, a target minimum charge value that rises more quickly.

   (With the charges all given in kWh, and the time period 5 minutes, these values are all realistic. The total charging period is 7.5 hours, and the maximum charging of 3kWh/period corresponds to a real power of 36kW. And no, you do not need to know or understand this to solve the problem.)

   Give the optimal RMS shortfall, *i.e.*, the squareroot of the optimal objective value. Plot the target minimum charge values and optimal state of charge for each vehicle, with dashed lines showing the target and solid lines showing the optimal charge. Plot the optimal charging energies $c_{t,i}$ over time in a stack plot.

Constant charging. Compare the optimal charging above to a very simple charging policy: Charge each vehicle at a constant energy per period, proportional to $q_i^{\text{des}} - q_{1,i}$, *i.e.*,

$$c_{t,i} = \theta_i C^{\max}, \qquad i = 1, \ldots, N, \quad t = 1, \ldots, T,$$

with

$$\theta_i = \frac{q_i^{\text{des}} - q_{1,i}}{\sum_{j=1}^{N} (q_j^{\text{des}} - q_{1,j})}, \quad i = 1, \ldots, N.$$

Give the associated RMS shortfall, and the same plots as above.

Plotting hints. In Python, a basic stack plot is obtained with

```
import matplotlib.pyplot as plt
plt.stackplot(rr, y.T)
```

where `rr` is a range object (like `range(a, b)`) with `len(list(rr)) == n` and `y` is an $n \times N$ NumPy array.

In Julia, a basic stack plot is obtained with

```
using Plots
areaplot(rr, y)
```

where `rr` is a range object (like `a:b`) and `y` is an $n \times N$ Matrix{Float64} object, with $n = b - a + 1$.

For those using Julia, you'll be better off using the solver ECOS, and not SCS or OSQP.

2. *Tail bounds for log-concave densities.* When $X \sim \mathcal{N}(0,1)$ and $a > 0$, a well-known upper bound on $\mathbf{prob}(X \geq a)$ is $\mathbf{prob}(X \geq a) \leq \varphi(a)/a$, where φ is the Gaussian density. In this exercise we explore a generalization of this bound to vector random variables and non-Gaussian, but log-concave, distributions.

   Let $X \in \mathbf{R}^n$ be a random variable with log-concave differentiable probability density function $p : \mathbf{R}^n \to \mathbf{R}_+$. We can express p as $p(x) = \exp(-\psi(x))$, where $\psi : \mathbf{R}^n \to \mathbf{R}$ is convex and differentiable.

   (a) *Tail bound.* Suppose that $\nabla p(a) \prec 0$ (which is the same as $\nabla \psi(a) \succ 0$). Show that

   $$\mathbf{prob}(X \succeq a) \leq p(a) \left(\prod_{i=1}^{n} (\nabla \psi(a))_i \right)^{-1}.$$

   We expect a solution based on ideas from this course, without reference to other tail bounds you might know about. *Remark.* When $X \sim \mathcal{N}(0,1)$, this recovers the well-known tail bound mentioned above.

   Hints.

   - Start with a basic inequality involving $\psi(x)$, $\psi(a)$, and $\nabla \psi(a)$, and from this obtain an upper bound on $p(x)$.
   - Recall that $\int_{x \succeq a} f_1(x_1) \cdots f_n(x_n)\, dx = \Pi_{i=1}^n \int_{x_i \geq a_i} f_i(x_i)\, dx_i$.

   (b) Evaluate the upper bound for the specific case $n = 2$, $X \sim \mathcal{N}(0, \Sigma)$, with

   $$\Sigma = \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}, \qquad \rho = 0.5, \qquad a = (3,3).$$

   We estimated $\mathbf{prob}(X \succeq a)$ (using a Monte Carlo method) as 8.2×10^{-5}; compare this to the upper bound.

3. *Flow optimization on a lossy network.* We consider a network represented as a directed graph with n nodes and m edges, with a single commodity flowing across the edges. With each edge we associate *two* nonnegative flows, the input flow u_j and the output flow v_j. We have $v_j \leq u_j$, with $u_j - v_j$ interpreted as the *loss* (of the commodity) on edge j. The relation between the input and output flows is given by an increasing convex function $\phi_j : \mathbf{R}_+ \to \mathbf{R}_+ \cup \{\infty\}$, with $u_j = \phi_j(v_j)$. These functions satisfy $\phi_j(0) = 0$ and $\phi_j(v_j) \geq v_j$. We think of $u_j = \phi_j(v_j)$ as giving the amount of flow that must go into edge i to achieve a given output flow v_j. We interpret $\phi_j(v_j) = \infty$ as meaning that there is no amount of input flow that can achieve an output flow v_j. We write this in compact vector form as $u = \phi(v)$, where $\phi : \mathbf{R}_+^m \to (\mathbf{R} \cup \{\infty\})^m$ is defined as $\phi(v) = (\phi_1(v_1), \ldots, \phi_m(v_m))$.

   An alternative, equivalent characterization is $v_j = \psi_j(u_j)$, where $\psi_j = \phi_j^{-1}$ gives the amount of output flow we achieve for a given input flow. These functions are increasing and concave, and satisfy $\psi_j(0) = 0$ and $\psi_j(u_j) \leq u_j$. We express this in compact vector form as $v = \psi(u)$.

   Lossy edges occur in many practical problems, for example power networks, when we model losses in transmission lines, or financial networks, where there are costs associated with moving money or some other good across an edge.

   Each node has an external source, with flow s_i into the node. Thus $s_i > 0$ means external flow into the node, and $s_i < 0$ means that there is a flow of value $-s_i$ out of the node.

   We have flow conservation at each node in the network. Flow comes into each node from the external source, and also from the output flows of each edge that is incoming to the node. Flow comes out of a node from each edge that is outgoing from the node, with the amount equal to the input flow of that edge. The total incoming and total outgoing flows must match. Let $A \in \mathbf{R}^{n \times m}$ denote the incidence matrix of the network, *i.e.*, $A_{ij} = 1$ if edge j is incoming to node i, $A_{ij} = -1$ if edge j is outgoing from node i, and $A_{ij} = 0$ otherwise. Define the matrices $A^{\text{in}} = \max\{A, 0\}$ (elementwise) and $A^{\text{out}} = \max\{-A, 0\}$, so $A = A^{\text{in}} - A^{\text{out}}$. The flow balance equations are then $A^{\text{in}} v + s = A^{\text{out}} u$. (We know that the description above is a lot to parse and follow; you can just use this equation as the flow balance constraint.)

   Each node has a cost function associated with its external flow, given by $f_i(s_i)$. We will assume that these are convex, and their extended valued extensions are nondecreasing. You can think of $f_i(s_i)$ as the cost of injecting flow into the network, when $s_i > 0$, and $-f_i(s_i)$ as the revenue or utility from extracting $-s_i$ from the network, when $s_i < 0$. The objective we wish to minimize is the total of these external flow costs, $f(s) = \sum_{i=1}^n f_i(s_i)$.

   Explain how to pose the problem of minimizing $f(s)$ subject to the constraints described above, with variables $u \in \mathbf{R}_+^m$, $v \in \mathbf{R}_+^m$, and the external flows $s \in \mathbf{R}^n$, as a convex optimization problem. If you use any relaxation, introduce new variables, or use a change of variables, be sure to justify it.

4. *Optimizing the sequence of commitments in an alternative investment.* In an *alternative investment*, the investor makes *commitments* each period for an amount that she will invest. Over the next few years, the investor puts money into the investment in response to *capital calls*, up to the amount of previous commitments. The investor receives money from the investment in later years through *distributions*. Examples of alternative investments include private equity, venture capital, and infrastructure projects. Alternative investments are found in the portfolios of insurance companies, retirement funds, and university endowments. ('Alternative' refers to the investment not being the more usual stocks, bonds, currencies, and financial derivatives.)

   We consider time periods $t = 1, \ldots, T$, which are typically quarters. We first describe some critical quantities.

   - $c_t \geq 0$ denotes the amount that the investor commits in period t.
   - $p_t \geq 0$ denotes the amount that the investor pays in to the investment in response to capital calls in period t.
   - $d_t \geq 0$ denotes the amount that the investor receives in distributions from the investment in period t.
   - $n_t \geq 0$ denotes the net asset value (NAV) of the investment in period t.
   - $u_t \geq 0$ denotes the total amount of uncalled commitments, *i.e.*, the difference between the total so far committed and the total so far that has been called (and paid into the investment).

   The units for all of these is typically millions of USD. Among these quantities, the only ones we have direct control over are the commitments c_t; the others are functions of these.

   A simple dynamical model of these variables is

   $$n_{t+1} = (1+r)n_t + p_t - d_t, \quad u_{t+1} = u_t - p_t + c_t, \quad t = 1, \ldots, T,$$

   where $r \geq 0$ is the per-period return, with initial conditions $n_1 = u_1 = 0$. (Note that n and u are $(T+1)$-vectors, whereas c, d, and p are T-vectors.) In words: the value of the investment increases by its return, plus the amount paid in, minus the amount distributed; the total uncalled commitments is decreased by the capital calls, and increased by new commitments. The calls and distributions are modeled as

   $$p_t = \gamma^{\text{call}} u_t, \quad d_t = \gamma^{\text{dist}} n_t, \quad t = 1, \ldots, T,$$

   where $\gamma^{\text{call}} \in (0,1)$ and $\gamma^{\text{dist}} \in (0,1)$ are the call and distribution intensities, respectively. The parameters r, γ^{call}, and γ^{dist} are given. Your job is to choose the sequence of commitments $c = (c_1, \ldots, c_T)$.

   The commitments and the capital calls are limited by $c_t \leq c^{\max}$ and $p_t \leq p^{\max}$, for $t = 1, \ldots, T$, where $c^{\max} > 0$ and $p^{\max} > 0$ are given. In addition we have a total budget $B > 0$ for commitments, with $\mathbf{1}^T c \leq B$. Our objective is to minimize

$$\frac{1}{T+1}\sum_{t=1}^{T+1}(n_t - n^{\mathrm{des}})^2 + \lambda \frac{1}{T-1}\sum_{t=1}^{T-1}(c_{t+1} - c_t)^2,$$

where $n^{\mathrm{des}} > 0$ is a given positive target NAV, and $\lambda > 0$ is a parameter. The first term in the objective is the mean-square tracking error, and the second term, the mean-square difference in commitments, encourages smooth sequences of commitments.

(a) *Optimized commitments.* Explain how to solve this problem with convex optimization. Solve this problem with parameters $T = 40$ (ten years), $r = 0.04$ (4% quarterly return),

$$\gamma^{\mathrm{call}} = .23, \quad \gamma^{\mathrm{dist}} = .15, \quad c^{\max} = 4, \quad p^{\max} = 3, \quad B = 85, \quad n^{\mathrm{des}} = 15, \quad \lambda = 5.$$

Plot c, p, d, n, and u versus t. Give the root-mean-square (RMS) tracking error, *i.e.*, the squareroot of the mean-square tracking error, for the optimal commitments.

(b) *Constant commitment based on steady-state.* By solving the dynamics equations with all quantities constant, we find that $c^{\mathrm{ss}} = (\gamma^{\mathrm{dist}} - r)n^{\mathrm{des}}$ is the value of a constant commitment (*i.e.*, the same each period) that gives $n_t = n^{\mathrm{des}}$ asymptotically, in steady-state. Plot the same quantities as in part (a) for the constant commitment $c_t = c^{\mathrm{ss}}$ for $t = 1, \ldots, T$. Give the RMS tracking error. *Hint.* A quick and simple (but not computationally efficient) way to do the simulation is to modify the code for part (a), adding the constraint that $c_t = c^{\mathrm{ss}}$, $t = 1, \ldots, T$.

Give a very brief description of what you see, comparing the optimal sequence of commitments found in part (a) and the constant commitments found in part (b).

5. *Predicting complete rankings.* A (complete) ranking of K items consists of an ordering of the items from rank 1 to rank K. For example, these could be K candidates, ranked from 1 (best) to K (worst), or the order in which K horses cross the finish line in a race. We represent a ranking of K items as a vector $\pi \in \mathbf{R}^K$, with π_i the rank of item i. In the vector π, the numbers $1, \ldots, K$ each appear exactly once (so it can also be considered a permutation), so there are $K!$ different rankings. We will let $\mathcal{P} \subset \mathbf{R}^K$ denote the set of all $K!$ rankings.

   For example with $K = 3$, $(2, 3, 1)$ and $(1, 3, 2)$ are two of the six possible rankings. In the first ranking, item 1 has rank 2, whereas in the second ranking, item 1 has rank 1. Both rankings agree that item 2 has rank 3.

   There are many ways to assign a distance between two rankings π and σ, but we will use a simple one, $(1/2)\|\pi - \sigma\|_1$. This distance is zero if and only if $\pi = \sigma$, and one if and only if π and σ assign the same rank to all items except two, whose ranks are off by one. The maximum possible distance is $K^2/4$ for K even and $(K^2 - 1)/4$ for K odd, achieved by, *e.g.*, $\pi = (1, 2, \ldots, K)$ and $\sigma = (K, K - 1, \ldots, 1)$. The average distance between two randomly chosen rankings is $(K^2 - 1)/6$. (These observations are not relevant for this problem, but only meant to give you an idea of the range and scale of the distance between rankings.)

   We wish to build a predictor of an outcome which is a ranking, based on a vector of features. We denote the predictor as $P : \mathbf{R}^d \to \mathcal{P}$, where $P(x)$ is the ranking we predict when the feature vector is $x \in \mathbf{R}^d$. We will judge a predictor by the average distance between the true ranking and the predicted one, on a test set of data $(x_i^{\text{test}}, \pi_i^{\text{test}})$, $i = 1, \ldots, N^{\text{test}}$ (that presumably was not used to develop or fit the predictor):

$$\frac{1}{2N^{\text{test}}} \sum_{i=1}^{N^{\text{test}}} \left\| \pi_i^{\text{test}} - P(x_i^{\text{test}}) \right\|_1 .$$

   We refer to this quantity as the average test error of the predictor. (The smaller this is, the better the predictor performs on the test data set.)

   We will consider a simple predictor of the form $P(x) = \Pi(\theta x)$, where $\theta \in \mathbf{R}^{K \times d}$ is the predictor coefficient matrix, and $\Pi : \mathbf{R}^K \to \mathcal{P}$ is Euclidean projection onto $\mathcal{P}$. (We will describe this projection in more detail below, but for now we note that if there are multiple rankings that are closest to θx, we arbitrarily choose one.)

   We choose the predictor parameter matrix θ to minimize

$$\frac{1}{2N} \sum_{i=1}^{N} \|\pi_i - \theta x_i\|_1,$$

   where (x_i, π_i), $i = 1, \ldots, N$, is some given training data. (Note that this objective would become the average distance between the true and predicted rankings if we replace θx_i with $\Pi(\theta x_i)$, but then the objective is no longer convex.)

Projection onto rankings. You can use the following, without deriving or justifying it. The projection $\pi = \Pi(y)$ is the vector of rank orders of the entries of y in nondecreasing order. For example with $y = (1.1, -0.3, 0.5, 0.4)$, we have $\Pi(y) = (4, 1, 3, 2)$, since the first entry of y is the largest (*i.e.*, has rank 4), the second entry of y is the smallest (*i.e.*, has rank 1), and so on. So we can compute $\Pi(y)$ by sorting the entries of y (breaking any ties arbitrarily), keeping track of the sort ordering.

Explain how to fit the predictor using the training data with convex optimization.

The data file `ranking_est_data.*` contains functions that generate synthetic training and test data, as well as a function that implements Π. The data are in the matrices `X_train`, `pi_train`, `X_test`, `pi_test`, and the projection Π is given in `Pi()`. Fit the predictor using the training data, and give the average distance between the true and predicted ranking on both the training and test data sets.

6. *Maximizing diversification ratio.* Let $x \in \mathbf{R}^n_+$, with $\mathbf{1}^T x = 1$, denote a portfolio of n assets, with x_i the fraction of the total value (assumed positive) invested in asset i. Let $\Sigma \in \mathbf{S}^n_{++}$ denote the covariance matrix of the asset returns. The *diversification ratio* of the portfolio is defined as

$$D(x) = \frac{\sigma^T x}{(x^T \Sigma x)^{1/2}},$$

where $\sigma_i = (\Sigma_{ii})^{1/2}$. Note that D is defined for any $x \in \mathbf{R}^n_+$ with $\mathbf{1}^T x = 1$.

We consider the problem of choosing x to maximize the diversification ratio, subject to limits on the weights,

$$\begin{array}{ll} \text{maximize} & D(x) \\ \text{subject to} & \mathbf{1}^T x = 1, \quad 0 \preceq x \preceq M, \end{array}$$

where $M \succ 0$ is a given vector of maximum allowed weights, with $\mathbf{1}^T M > 1$.

Remark. (The following is not needed to solve the problem, but gives some background.) For any long-only portfolio x we have $D(x) \geq 1$. To see this we note that

$$x^T \Sigma x = \sum_{ij} x_i x_j \sigma_i \sigma_j \rho_{ij} \leq \sum_{ij} x_i x_j \sigma_i \sigma_j = (\sigma^T x)^2,$$

where $\rho_{ij} = \Sigma_{ij}/(\sigma_i \sigma_j)$ is the correlation, which satisfies $\rho_{ij} \leq 1$. The smallest possible value of diversification $D(x) = 1$ occurs only when $x = e_k$ (the kth unit vector), *i.e.*, the portfolio is concentrated in one asset.

(a) Explain how to use convex optimization to solve the problem. We will give half credit for a solution that involves solving a quasiconvex optimization problem, and full credit to one that relies on solving one convex problem. *Hints.* You may need to change variables to get a one-convex-problem method. Note also that $D(tx) = D(x)$ for any $t > 0$.

(b) Use your method from part (a) to solve the problem instance with data given in `max_divers_data.*`. Give an optimal $x^\star$, and the associated diversification ratio $D(x^\star)$.

The (long-only) *minimum variance portfolio* x^{mv} is the one that minimizes $x^T \Sigma x$ subject to $0 \preceq x \preceq M$, $\mathbf{1}^T x = 1$. Find $D(x^{\mathrm{mv}})$, and compare it to $D(x^\star)$. Compare the maximum diversification and minimum variances portfolios using a bar plot. (The data file contains code for creating such plots.)

7. *Estimating mixture coefficients.* We are given N IID samples $x_1, \ldots, x_N \in \mathbf{R}^m$ from a distribution with mixture density

$$p(x;\lambda) = \sum_{j=1}^{k} \lambda_j p_j(x),$$

where $\lambda \in \mathbf{R}^k_+$, with $\mathbf{1}^T\lambda = 1$, are the mixture coefficients, and $p_1, \ldots, p_k$ are given densities on $\mathbf{R}^m$.

(a) Explain how to use convex optimization to find the maximum likelihood estimate of the mixture coefficients $\lambda^{\rm ml} \in \mathbf{R}^k_+$. (You can assume that the maximum likelihood problem is well posed, *i.e.*, there is an optimal $\lambda^{\rm ml} \in \mathbf{R}^k_+$.) If you change variables, or form a relaxation, be sure to fully justify it.

Note. We will not accept methods or algorithms from other courses or fields, even if they work.

(b) The data files `mixture_coeffs_data.*` contain code that generates $N = 100$ samples from a mixture of $k = 3$ distributions on $\mathbf{R}$,

$$\mathcal{N}(3,4), \quad \mathcal{U}(-1,2), \quad \mathcal{L}(-2,3),$$

with mixture coefficients $\lambda^{\rm true} = (0.3, 0.5, 0.2)$. The first distribution is Gaussian with mean 3 and variance 4; the second is a uniform distribution on $[-1, 2]$, and the third is a Laplace or double-sided exponential distribution with mean -2 and shape parameter 3, which has density $p(x) = \frac{1}{6}\exp(-|x+2|/3)$. The data file contains code for evaluating the density values at the sample points, *i.e.*,$p_j(x_i)$, $j = 1, 2, 3$ and $i = 1, \ldots, N$.

Carry out the method of part (a) on this data. Compare the ML estimate of the mixture coefficients with their true values. Plot the true and estimated mixture densities on the same plot. The data file also contains code for these plots; you just have to plug in your $\lambda^{\rm ml}$. (Of course, in any real problem you would not have a 'true' distribution.)

8. *Optimal exchange.* We consider a market with n (divisible) goods that a set of N agents or participants can exchange or trade with each other. We let $x_i \in \mathbf{R}^n$ denote the amounts of goods that agent i takes, with $(x_i)_j < 0$ meaning that agent i gives the amount $|(x_i)_j|$. We say that the market *clears* if $x_1 + \cdots + x_N = 0$, which means that for each good, the total amount taken by participants balances the total amount given by other participants. (We assume that each participant starts with an endowment of goods, which allows them to give some away.) The particular choice $x_i = 0$, $i = 1, \ldots, N$, means that no goods are exchanged.

   Each participant derives a utility $U_i(x_i)$ (in dollars, say) from the level of goods taken (or given) x_i. We will assume that the functions $U_i : \mathbf{R}^n \to \mathbf{R}$ are increasing, strictly concave, and differentiable, with $0 \in \mathbf{dom}\, U_i$, $i = 1, \ldots, N$. (Everything can be made to work when they are just concave, but it gets more complicated.)

   Suppose $x_i^\star$, $i = 1, \ldots, N$, maximize the total utility $U_1(x_1) + \cdots + U_N(x_N)$ subject to the market clearing. Unless all of these are are zero, we have (by definition)

$$U_1(x_1^\star) + \cdots + U_N(x_N^\star) > U_1(0) + \cdots + U_N(0),$$

   which means that by optimal trading, the total utility increases. In this exercise we discuss how to compensate the participants, or, put another way, how to allocate the increase in total utility to the participants.

   To fix the sign convention for the dual variable, we work with the Lagrangian

$$L(x_1, \ldots, x_N, \nu) = U_1(x_1) + \cdots + U_N(x_N) - \nu^T(x_1 + \cdots + x_N),$$

   with dual variable $\nu \in \mathbf{R}^n$, and let $p = \nu^\star$ denote an optimal dual variable value. (And yes, you do have strong duality here.)

   Not surprisingly, p can be interpreted as a vector of prices for the goods. Below you will show that $p \succ 0$, *i.e.*, the prices for the goods are all positive. The payment by participant i (in dollars) for participating in the exchange is $p^T x_i$. (If this is negative, participant i receives money.) You will work out various properties of this payment scheme.

   (a) *Price and marginal utility.* Relate p to $\nabla U_i(x_i^\star)$. The latter is the marginal utility of the goods to participant i, at $x_i^\star$. From this relation, conclude that $p \succ 0$.

   (b) *Cash balance.* Show that the sum of the payments across the participants is zero. This means that the total cash paid in by participants balances the total cash paid out to participants. In other words, the cash payments also clear.

   (c) *Nash equilibrium.* Explain why $x_i^\star$ maximizes $U_i(x_i) - p^T x_i$, which is the net utility for participant i. In other words, with the prices of goods fixed at p, each participant maximizes their net utility with $x_i = x_i^\star$. This is called a *Nash equilibrium*: No participant is incentivized to change their value of x_i from $x_i^\star$.

(d) *Everyone does better by trading.* Show that for each i, $U_i(x_i^\star) - p^T x_i^\star \geq U_i(0)$. (The inequality is strict when $x_i^\star \neq 0$.) The lefthand side is the net utility when the participant trades; the righthand side is the (net) utility when she does not trade.

Your solutions can be brief; we will penalize solutions that are substantially more complicated than they need to be.