1 Questions 1-3: Resource Allocation Problem

Hint: refer to Chapter 2.B of Introduction to Linear and Matrix Algebra for examples.

You are in charge of a company that makes two hot sauces: x_1 liters of Kapatio and x_2 liters of Zriracha. We will use optimization technique to find the “best” manufacturing strategy given our resource constraints.

First, we need to define what we mean by “best” strategy. In this scenario, the goal is to obtain the highest revenue possible. While doing so, there are resource constraints we must satisfy.

For example, in order to manufacture these two hot sauce products, different amount of peppers and vineger are needed. Also, we have only so much total resource available.

Ingridients	Kapatio	Zriracha	Total Available
Pepper	5	7	30
Vineger	4	2	12

1.1 Question 1: Resource Constraints

1.1.1 Question 1.a: Modeling Resource Usage

What is the equation for the amount of pepper needed to manufacture x_1 and x_2. What is the equation for the amount of vinegar? (Use Mathpix to write equations)

x_1 is liters of Kapatio produced, x_2 is liters of Zriracha produced.

Amount of pepper needed $= 5x_1 + 7x_2$

Amount of vinegar needed $= 4x_1 + 2x_2$

1.1.2 Question 1.b: Resource Usage vs. Total Resource Constraint

Total amount of pepper needed cannot exceed total available. Write down the inequality expressing this relationship. Do the same for vinegar. These inequalities are your resource constraints. Additinally, variables x_1 and x_2 are non-negative: i.e. amount of manufactured goods cannot be negative.

Rewrite the system of constraint inequalities into a matrix inequality: $Ax \leq b$, where $x = (x_1, x_2)^T$. Arrange rows of A and b such that:

- Row 1: total pepper amount constraint
- Row 2: total vinegar amount constraint
- Row 3: Kapatio non-negativity constraint
- Row 4: Zriracha non-negativity constraint

Less than symbol in $Ax \leq b$ means element-wise.

Pepper constraint: $5x_1 + 7x_2 \leq 30$

Vinegar constraint: $4x_1 + 2x_2 \leq 12$

$x_1 \geq 0$, $x_2 \geq 0$

Sometimes, things just do not work as expected.

In the toy example code,

```
sns.lineplot(x='$z_1$', y='$z_2$', hue='constraint', data=pd.concat(boundary), ax=ax).axvline(1)
```

what seems strange about the plotting command? Why was the strange code necessary?

This code looks fine other than the specification of axvline(1). Within constraint, this line is already defined so it is strange that we had to specify this. This was necessary because the line does not appear without axvline(1).

By dissecting the command below and reading the documentation, report what each of the following lines does:

- `((y1_grid >= 1) & (y2_grid >= 2)).astype(int)` (What is the output of running this command?)
- `origin='lower'`
- `extent=(y1_grid.min(), y1_grid.max(), y2_grid.min(), y2_grid.max())`
- `cmap='Greys'`
- `alpha=0.3`
- `aspect='equal'`

```
In [12]: print(z1_grid[1])
```

```
[-1.         -0.22222222  0.55555556  1.33333333  2.11111111  2.88888889
  3.66666667  4.44444444  5.22222222  6.        ]
```

Line 1 - This line outputs an array of the same shape of y1_grid and y2_grid with 1s when y1_grid value is ≥ 1 and y2_grid value is ≥ 2, 0 otherwise. This is the data to be plotted.

Line 2 - Moves the [0,0] index to the lower left of the matrix. Default is upper.

Line 3 - Determines the bounds of the image, (left, right, bottom, top). Sets the bounds of the visualization to include only y1_grid x and y2_grid y values.

Line 4 - Sets the color map to be greys, the shaded region will be grey.

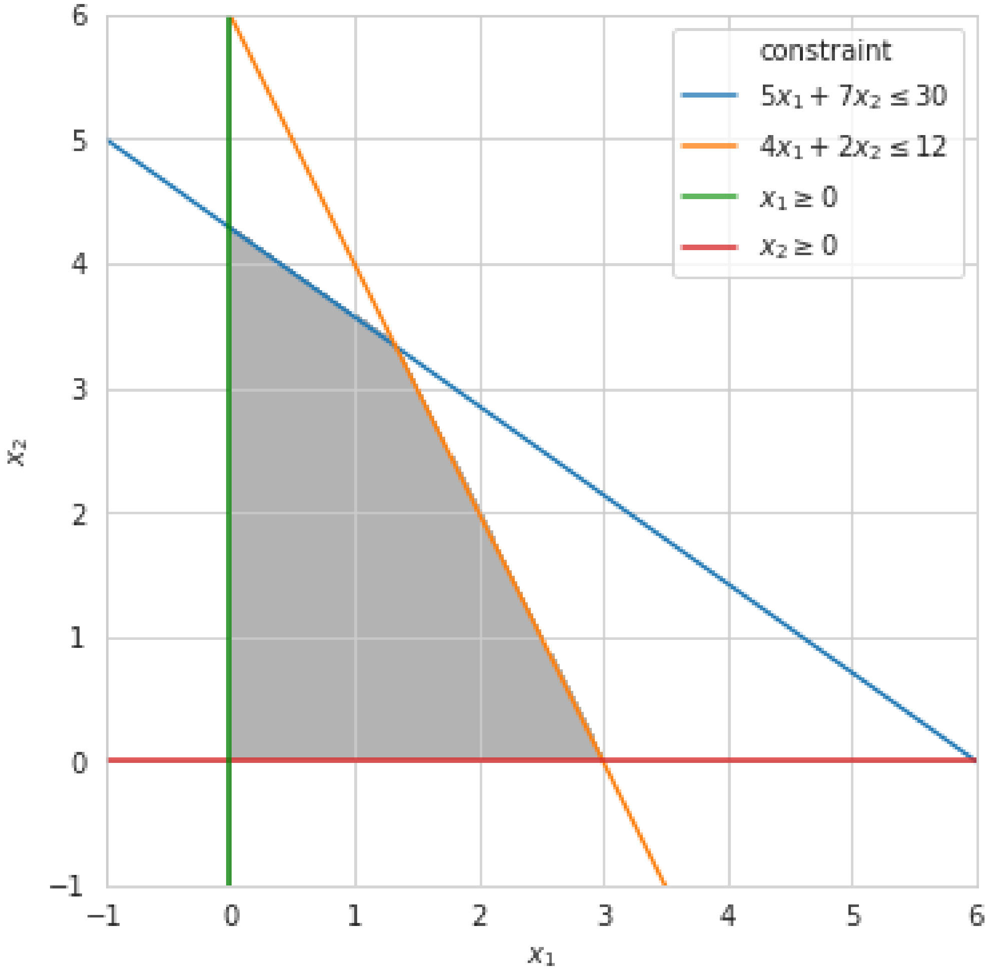
Line 5 - Sets a blending value to make the plot closer to transparent than opaque.

Line 6 - Sets aspect ratio to be equal so that the plot is a square.

1.1.3 Question 1.e: Visualizing the Feasible Region

Finally, create a figure that shows constraint boundaries and the interior region shaded with a light grey color.

Your output will look like this:

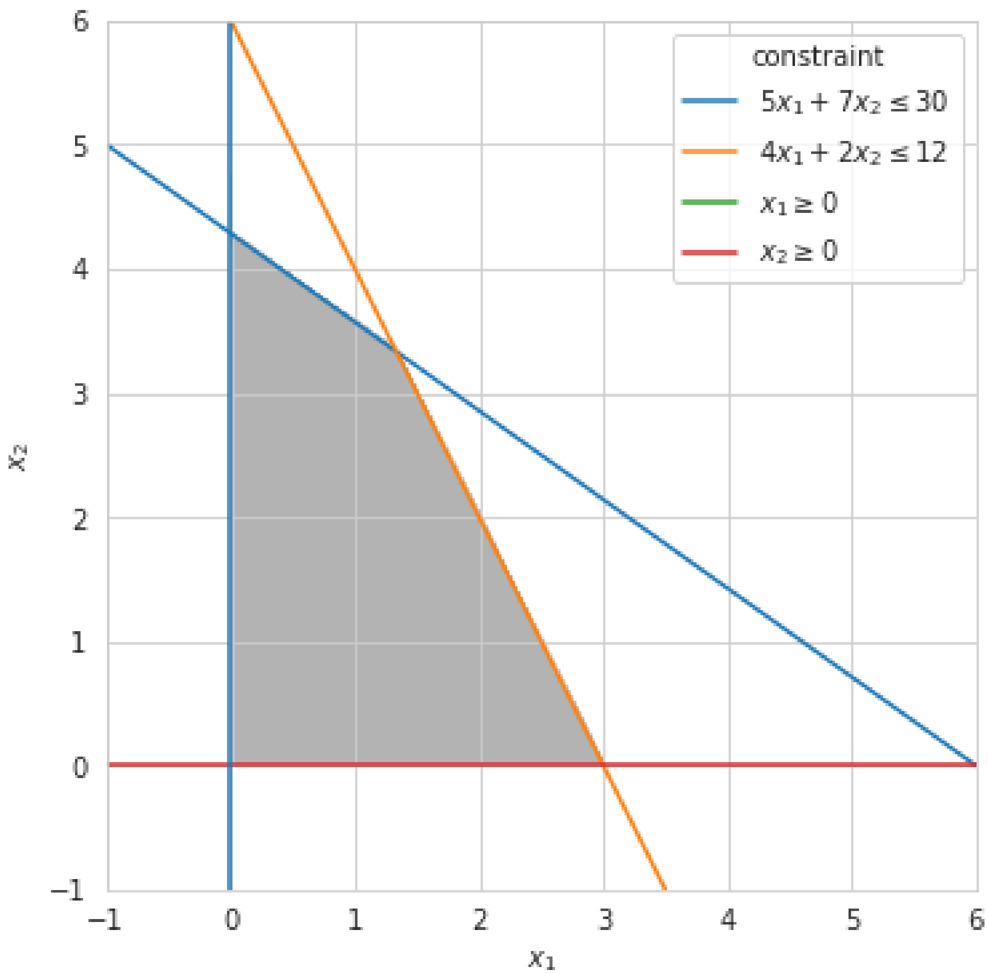


In [13]:
```
fig, ax = plt.subplots(figsize=(6, 6))
x1_grid, x2_grid = np.meshgrid(x1_line, x2_line)

ax.imshow(
    (
        ((x1_grid >= 0) & (x2_grid >= 0) & (x2_grid <= (30 - 5 * x1_grid) / 7) & (x2_grid <= (
    ),
    origin='lower',
    extent=(x1_grid.min(), x1_grid.max(), x2_grid.min(), x2_grid.max()),
    cmap="Greys", alpha = 0.3, aspect='equal')
```

```python
# ax = sns.lineplot(???).axvline(???)
ax = sns.lineplot(x='$x_1$', y='$x_2$', hue='constraint', data=pd.concat(boundary).reset_index

plt.xlim(-1, 6)
plt.ylim(-1, 6)
plt.show()
```



In the context of linear programming, $Ax \leq b$ is called the *feasible region* (including the appropriate sections of the boundaries). Denote the (shaded) feasible region as set C. Points $(x_1, x_2) \in C$ satisfy all of the constraints.

Describe in plain words the feasible region in the context of hot sauce manufacturing. Specifically, which constraint is violated (if any) by a point at:

- $(x_1, x_2) = (4, 1)$
- $(x_1, x_2) = (0, 5)$
- $(x_1, x_2) = (3, 4)$

The constraints are listed above on the legend.

$(x_1, x_2) = (4, 1)$ violates the constraint $4x_1 + 2x_2 \leq 12$.

$(x_1, x_2) = (0, 5)$ violates the constraint $5x_1 + 7x_2 \leq 30$.

$(x_1, x_2) = (3, 4)$ violates both constraints $4x_1 + 2x_2 \leq 12$ and $5x_1 + 7x_2 \leq 30$.

1.2 Question 2: Objective Function

1.2.1 Question 2.a: Defining Objective Function

Suppose the hot sauces are sold at the same price: \$5 per liter.

What is the equation $f(x)$ for the total revenue as a function of x_1 and x_2?

The function $f(x)$ is called the objective function.

Consider the objective function $f(x) = 5x_1 + 5x_2$.

1.2.2 Question 2.b: Direction of Steepest Increase

Since we want to maximize revenue, we want to increase our objective function as much as possible. Analogous to the minimization example given in a previous lecture, we can repeatedly move in the direction of function increase. In order to determine such direction, compute the gradient of $f(x)$ at $x = (0,0)^T$:

$$\nabla_x f(x) = \begin{pmatrix} \frac{\partial f(x)}{\partial x_1} \\ \frac{\partial f(x)}{\partial x_2} \end{pmatrix}$$

$$\nabla_x f(x) = \begin{pmatrix} \frac{\partial f(0)}{\partial x_1} \\ \frac{\partial f(0)}{\partial x_2} \end{pmatrix} = \begin{pmatrix} 5 \\ 5 \end{pmatrix}$$

1.3 Question 3: Putting Pieces Together

1.3.1 Question 3.a: Standard Form of a Linear Programming Problem

Write down the so-called the *standard form* of a linear programming problem:

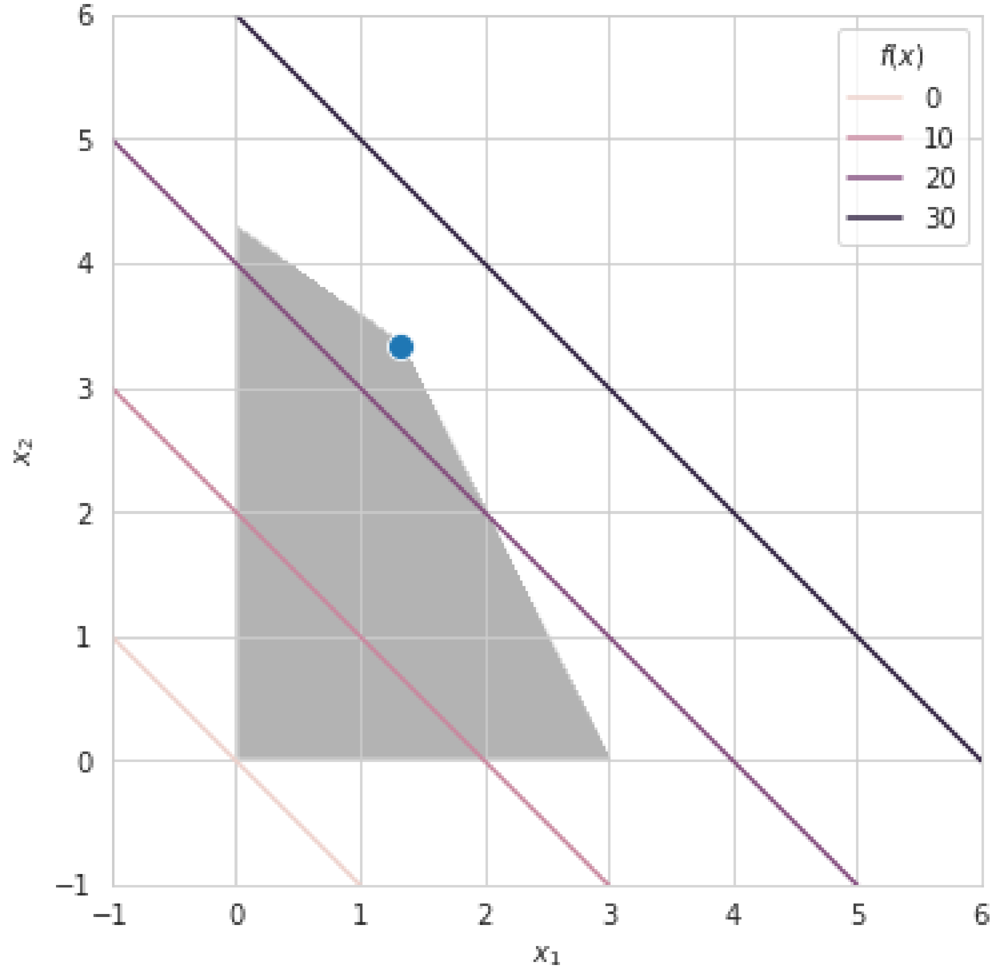
$$\max_x f(x)$$
$$\text{subject to } Ax \leq b$$

Specifically, write the obejective as an inner product of two vectors: $f(x) = c^T x$, and write the constraint as a vector inequality involving a matrix-vector prduct: $Ax \leq b$, where A is a 4-by-2 matrix.

$$\max_x f(x) = c^T x$$
$$\text{subject to } Ax \leq b$$
$$c^T = \begin{pmatrix} 5 & 5 \end{pmatrix}$$
$$A = \begin{pmatrix} 5 & 7 \\ 4 & 2 \\ -1 & 0 \\ 0 & -1 \end{pmatrix}$$
$$b^T = \begin{pmatrix} 30 & 12 & 0 & 0 \end{pmatrix}$$

1.3.2 Question 3.c: Plotting the optimal solution

```
In [18]: fig, ax = plt.subplots(figsize=(6, 6))
         x1_grid, x2_grid = np.meshgrid(x1_line, x2_line)
         ax.imshow(
             (
                 (A1[0,0]*x1_grid + A1[0,1]*x2_grid <= b1[0]) & # Pepper constraints
                 (A1[1,0]*x1_grid + A1[1,1]*x2_grid <= b1[1]) & # Vinegar constraints
                 (A1[2,0]*x1_grid + A1[2,1]*x2_grid <= b1[2]) &
                 (A1[3,0]*x1_grid + A1[3,1]*x2_grid <= b1[3])   # non-negativity constraints
             ),
             origin='lower',
             extent=(x1_grid.min(), x1_grid.max(), x2_grid.min(), x2_grid.max()),
             cmap="Greys", alpha = 0.3, aspect='equal'
         )
         sns.scatterplot(x='$x_1$', y='$x_2$', data=xstar1, ax=ax, s=100)
         sns.lineplot(x='$x_1$', y='$x_2$', hue='$f(x)$', data=f_vals.reset_index(), ax=ax)
         plt.xlim(-1, 6)
         plt.ylim(-1, 6)
         plt.show()
```



f(x)
0
10
20
30
x2
x1
−1
0
1
2
3
4
5
6

1.3.3 Question 4.a: Define Constraints

To avoid eating the same foods, limit each food intake to be 2 or less. Also, one cannot consume less than zero servings. Furthermore, apply the nutritional constraints as specified in `nutritional_constraints.csv` (assume that the units are the same as food nutritional contents)

Note that a range constraints, e.g., $2000 \leq \text{total calories} \leq 2250$, can be written as two constraints: $\text{total calories} \leq 2250$ and $-\text{total calories} \leq -2000$. Hence, we can rewrite caloric intake constraints as

$$-(\text{calories in frozen broccoli})x_0 - (\text{calories in raw carrots})x_1 - \cdots - (\text{calories in bean bacon soup, w/watr})x_{63} = -c^T x \leq -2$$
$$\text{calories in frozen broccoli})x_0 + (\text{calories in raw carrots})x_1 + \cdots + (\text{calories in bean bacon soup, w/watr})x_{63} = c^T x \leq 22$$

where vector c contains calorie information for all 64 foods and x containts servings consumed of each food. Matrix U and vector w would be such that

$$U = \begin{pmatrix} -c^T \\ c^T \end{pmatrix} \text{ and } w = \begin{pmatrix} -2000 \\ 2250 \end{pmatrix},$$

and the matrix-vector inequality would be $Ux \leq w$. Range constraints of each food can be implemented similarly with identity matrices.

Denote nutritional content information from `foods` data frame as A and denote the `Min` and `Max` columns of `requirements` as vector b_L and b_U, respectively. Construct M and d in $Mx \leq d$ using I (identity matrix), A, b_L, b_U, and other constants, so that all the range constraints are expressed in $Mx \leq d$. (This is a theory question. No coding is involved)

$$M = \begin{pmatrix} -1 \\ 1 \\ -A \\ A \end{pmatrix} d = \begin{pmatrix} 0 \\ 2 \\ -b_L \\ b_U \end{pmatrix}$$

1.3.4 Question 4.d: Interpreting the Results

State the results in the context of the problem. How much of each food was consumed? List the foods and their calculated amounts. What is the total cost of feeding one soldier?

```
In [27]: bruh = pd.DataFrame({'Calculated Amount' : xstar2}, index = foods.iloc[:,0])
         pd.set_option('display.max_rows', None)
         pd.set_option('display.max_columns', None)
         print(fstar2)
         print(bruh.sort_values(by='Calculated Amount', ascending=False))
```

```
1.2484722937208912
                          Calculated Amount
Name
Potatoes, Baked                2.000000e+00
Peanut Butter                  2.000000e+00
Popcorn, Air-Popped            2.000000e+00
Chocolate Chip Cookies         2.000000e+00
White Rice                     2.000000e+00
…                                       …
White Tuna in Water            5.412695e-13
Hamburger w/Toppings           5.004215e-13
New Eng Clam Chwd, w/Mlk       3.453333e-13
Grapes                         1.956439e-13
Oatmeal                        1.415479e-13

[64 rows x 1 columns]
```

See the output above for consumption of each food. The total cost of feeding one soldier for one day is 1.2484722937208912.

The above diet satisfies the given nutritional constraints for the lowest cost possible.

1.3.5 (PSTAT 234) Question 5.a: Assignment Problem

Given a set $\mathcal{S}$ of m people, a set $\mathcal{D}$ of m tasks, and for each $i \in \mathcal{S}, j \in \mathcal{D}$ a cost c_{ij} associated with assigning person i to task j, the assignment problem is to assign each person to one and only one task in such a manner that each task gets covered by someone and the total cost of the assignments is minimized. If we let

$$x_{ij} = \begin{cases} 1 & \text{if person } i \text{ is assigned task } j, \\ 0 & \text{otherwise,} \end{cases}$$

then the objective function can be written as

$$\text{minimize} \sum_{i \in \mathcal{S}} \sum_{j \in \mathcal{D}} c_{ij} x_{ij}.$$

Note that x_{ij} values are constrained to be 0 or 1. This is also a constraint. Suppose that, additionally, we would like to build in the constraints that 1. Each person is assigned exactly one task. 2. Every task gets covered by someone.

What equality constraints can you add?

Type your answer here, replacing this text.

1.3.6 (PSTAT 234) Question 5.b: Transportation Problem

Transportation problem: i and j indicate pairs of locations, quantity of goods (x_{ij}), supply (s_i), demand (d_j), transportation network ($\mathcal{A}$)

$$
\begin{aligned}
\min_{x=(x_{ij})\geq 0} & \sum_{(i,j)\in\mathcal{A}} c_{ij}x_{ij} \\
\text{sugject to} & \sum_{j:(i,j)\in\mathcal{A}} x_{ij} \leq s_i \quad \text{for all } i \\
& \sum_{i:(i,j)\in\mathcal{A}} x_{ij} \geq d_j \quad \text{for all } j
\end{aligned}
$$

Describe the assignment problem as a special case of the transportation problem. Draw analogy of supply network ($\mathcal{A}$), supply (s_i), demand (d_j), and quantity of goods (x_{ij}) in the assignment problem context.

Type your answer here, replacing this text.