

## Assignment 1: Linear Programming Example - The Diet Problem Revisited

Su Hyun Byeon

MSDS 460: Decision Analytics

January 19, 2025

This study focused on a diet optimization problem constrained by the U.S. Food and Drug Administration's (USFDA) daily nutrition intake recommendations, using five personally selected food items. The chosen foods were priced packaged items with listed nutrition facts for convenient calculations. While the challenge was identifying foods that not only reflected my typical diet but also included all seven nutrients, I opted for a ribeye steak, a chicken and apple sausage, kimbap, Greek yogurt, and tofu. See Appendix 1 for their nutrition facts.

The goal of this problem is to determine the number of servings of the five food items to purchase and consume to minimize costs while meeting the nutritional requirements. My weekly diet consists of a ribeye steak for \$6.84 per serving, chicken and apple sausage for \$1.12 per serving, kimbap for \$1.66 per serving, Greek yogurt for \$0.83 per serving, and tofu for \$0.63 per serving. See Appendix 2 for price breakdown. The USFDA recommends a maximum intake of 5,000mg of sodium and minimums of 2000kcal, 50g of protein, 20mcg of vitamin D, 1,300mg of calcium, 18mg of iron, and 4,700mg of potassium. The number of servings must also be non-negative. The objective is to minimize the weekly cost. See Appendix 3 for problem formulation.

There are five decision variables representing the number of servings for each food item. Since the objective is to minimize the total food cost, the objective function is a minimization problem, summing the cost and serving amount for each food item. The problem is subject to eight constraints: seven nutritional requirements and the non-negativity constraint. Each food's nutritional content is broken down in Appendix 4. For each nutritional requirement, its constraint sums the nutritional values of all foods, multiplied by the respective servings and scaled to reflect weekly intake. The linear programming (LP) problem was formulated in Python and solved using PuLP. The optimal solution is 22.22 servings of Greek yogurt and 15 servings of tofu, resulting in a weekly cost of \$27.97. See Appendix 5 for problem formulation in Python.

To increase variety to the suggested diet plan, I added additional constraints to ensure at least one serving of each food. This resulted in 1 serving each of the steak, sausage, and kimbap, 22.22 servings of Greek yogurt, and 13.02 servings of tofu. The number of servings were adjusted except for the yogurt, raising the weekly cost to \$36.27 from \$27.97 shown in Appendix 6. To further diversify and personalize the diet, I would include carbohydrates, cholesterol and other nutrients to be more mindful of or ensure the inclusion of food groups like vegetables.

To compare the performance of large language models (LLM) in solving similar problems, I experimented with ChatGPT -<https://chatgpt.com/>. See Appendix 7 for full ChatGPT conversation. For the first prompt, I assigned the AI agent the role of data scientist, provided context for the problem, and instructed to construct a model, write code for the problem, and ask questions for clarification. The agent successfully generated the problem setup and corresponding Python code and asked data-related questions. After a couple failed attempts of instructing online search for nutrition facts and submitting images of them, the third attempt of sending a table containing all the necessary and correct data allowed the agent to complete the code with the provided data. However, when asked to solve it, I encountered a data analysis limit. Regardless, I believe an LLM agent can fully complete the assignment when given correct data, context, and access to subscribed version.

This study demonstrated the application of an optimization technique in an LP problem using Python and an LLM model. While Python is a great diverse tool for solving simple and complex problems, LLMs can assist in model construction and data analysis with accurate data and advanced analytics capabilities.

Appendix 1: Nutrition Facts

Aldi  
Grass Fed Ribeye Steak

Nutrition Facts

Serving Size 1 serving (4 oz)

Amount Per Serving  
Calories 390

% Daily Values*	
Total Fat	35.00g 45%
Saturated Fat	15.000g 75%
Trans Fat	0.000g
Cholesterol	80mg 27%
Sodium	60mg 3%
Total Carbohydrate	0.00g 0%
Dietary Fiber	0.0g 0%
Sugars	0.00g
Includes 0.00g Added Sugars	0%
Protein	18.00g
Vitamin D	0mcg 0%
Calcium	30mg 2%
Iron	2.00mg 11%
Potassium	260mg 6%

\* The % Daily Value (DV) tells you how much a nutrient in a serving of food contributes to a daily diet. 2,000 calories a day is used for general nutrition advice.

Aidells  
Chicken & Apple Sausage

Nutrition Facts

Serving Size 1 link (90 g)

Amount Per Serving  
Calories 170

% Daily Values*	
Total Fat	12.00g 15%
Saturated Fat	3.500g 18%
Trans Fat	0.000g
Cholesterol	75mg 25%
Sodium	660mg 29%
Total Carbohydrate	4.00g 1%
Dietary Fiber	1.0g 4%
Sugars	3.00g
Protein	13.00g
Vitamin D	-
Calcium	-
Iron	-
Potassium	-

\* The % Daily Value (DV) tells you how much a nutrient in a serving of food contributes to a daily diet. 2,000 calories a day is used for general nutrition advice.

Trader Joe's  
Kimbab

Nutrition Facts

Serving Size 3 pieces (77 g)

Amount Per Serving  
Calories 140

% Daily Values*	
Total Fat	4.00g 5%
Saturated Fat	5.000g 25%
Trans Fat	0.000g
Cholesterol	0mg 0%
Sodium	320mg 14%
Total Carbohydrate	23.00g 8%
Dietary Fiber	1.0g 4%
Sugars	2.00g
Includes 2.00g Added Sugars	4%
Protein	4.00g
Vitamin D	0mcg 0%
Calcium	50mg 4%
Iron	0.50mg 3%
Potassium	70mg 1%

\* The % Daily Value (DV) tells you how much a nutrient in a serving of food contributes to a daily diet. 2,000 calories a day is used for general nutrition advice.

Soonyeowon  
Firm Tofu

Stonyfield  
ORGANIC

NUTRITION FACTS

Serving Size 3/4 Cup (170g)  
Servings Per Container 5

Amount Per Serving

Calories 150

% Daily Value\*

Total Fat 6g 8%  
Saturated Fat 3.5g 18%  
Trans Fat 0g

Cholesterol 25mg 8%

Sodium 65mg 3%

Total Carbohydrate 7g 3%  
Dietary Fiber 0g 0%  
Total Sugars 5g  
Added Sugars 0g 0%

Protein 16g 32%

Vitamin D 0.9mcg 4%  
Iron 0mg 0%

Calcium 170mg 15%  
Potassium 200mg 4%

Nutrition Facts

Serving Size 3 oz (85 g)

Amount Per Serving  
Calories 80

% Daily Values*	
Total Fat	4.50g 6%
Saturated Fat	1.500g 8%
Trans Fat	0.000g
Polyunsaturated Fat	0.000g
Monounsaturated Fat	0.000g
Cholesterol	0mg 0%
Sodium	5mg 0%
Total Carbohydrate	1.00g 0%
Dietary Fiber	1.0g 4%
Sugars	0.00g
Includes 0.00g Added Sugars	0%
Protein	10.00g
Vitamin D	0mcg 0%
Calcium	150mg 12%
Iron	1.19mg 7%
Potassium	140mg 3%

\* The % Daily Value (DV) tells you how much a nutrient in a serving of food contributes to a daily diet. 2,000 calories a day is used for general nutrition advice.

## Appendix 2: Price Breakdown

	Total Price	Servings per package	Price per Serving
Ribeye Steak	\$13.69	2	\$6.84
Chicken & Apple Sausage	\$16.79	15	\$1.12
Kimbap	\$4.99	3	\$1.66
Greek Yogurt	\$4.15	5	\$0.83
Tofu	\$3.79	6	\$0.63

### Appendix 3: Linear Programming Standard Form

$$\text{minimize } Z = 6.84x_1 + 1.12x_2 + 1.66x_3 + 0.83x_4 + 0.63x_5$$

*subject to:*

$$7(60x_1 + 660x_2 + 320x_3 + 65x_4 + 5x_5) \leq 7 * 5000$$

$$7(390x_1 + 170x_2 + 140x_3 + 150x_4 + 80x_5) \geq 7 * 2000$$

$$7(18x_1 + 13x_2 + 4x_3 + 16x_4 + 10x_5) \geq 7 * 50$$

$$7(0.9x_4) \geq 7 * 20$$

$$7(30x_1 + 50x_3 + 170x_4 + 150x_5) \geq 7 * 13000$$

$$7(2x_1 + 0.5x_3 + 1.19x_5) \geq 7 * 18$$

$$7(260x_1 + 70x_3 + 200x_4 + 140x_5) \geq 7 * 4700$$

$$x_1, x_2, x_3, x_4, x_5 \geq 0$$

## Appendix 4: Nutrition Breakdown

	Ribeye Steak	Chicken & Apple Sausage	Kimbap	Greek Yogurt	Tofu
Sodium	60	660	320	65	5
Energy	390	170	140	150	80
Protein	18 g	13	4	16	10
Vitamin D	0	0	0	0.9	0
Calcium	30	0	50	170	150
Iron	2	0	0.5	0	1.19
Potassium	260	0	70	200	140

## Appendix 5: Original Problem

Code:

```

import pulp
from pulp import LpVariable, LpProblem, LpMaximize, LpStatus, value, LpMinimize
# Problem (Initial)
# define variables
x1 = LpVariable("Ribeye", 0, None) # MFR>=0
x2 = LpVariable("Sausage", 0, None) # EOC>=0
x3 = LpVariable("Kimbap", 0, None) # ER>=0
x4 = LpVariable("Yogurt", 0, None) # SC>=0
x5 = LpVariable("Tofu", 0, None) # WM>=0

# defines the problem
prob = LpProblem("problem", LpMinimize)

# define constraints
prob += 7*(60*x1 + 660*x2 + 320*x3 + 65*x4 + 5*x5) <= 7*5000 # sodium constraint
prob += 7*(390*x1 + 170*x2 + 140*x3 + 150*x4 + 80*x5) >= 7*2000 # energy constraint
prob += 7*(18*x1 + 13*x2 + 4*x3 + 16*x4 + 10*x5) >= 7*50 # protein constraint
prob += 7*(0*x1 + 0*x2 + 0*x3 + 0.9*x4 + 0*x5) >= 7*20 # vitamin D constraint
prob += 7*(30*x1 + 0*x2 + 50*x3 + 170*x4 + 150*x5) >= 7*1300 # calcium constraint
prob += 7*(2*x1 + 0*x2 + 0.5*x3 + 0*x4 + 1.19*x5) >= 7*18 # iron constraint
prob += 7*(260*x1 + 0*x2 + 70*x3 + 200*x4 + 140*x5) >= 7*4700 # potassium constraint

# define objective function
prob += 6.84*x1 + 1.12*x2 + 1.66*x3 + 0.83*x4 + 0.63*x5
# solve the problem
status = prob.solve()
print(f"Problem")
print(f"status={LpStatus[status]}")
# print the results
for variable in prob.variables():
    print(f"{variable.name} = {variable.varValue}")
print(f"Objective = {value(prob.objective)}")
print(f"")

```

Output:

```

Problem
status=Optimal
Kimbap = 0.0
Ribeye = 0.0
Sausage = 0.0
Tofu = 15.12605
Yogurt = 22.222222
Objective = 27.97385576

```



## Appendix 6: Revised Problem

Code:

```

import pulp
from pulp import LpVariable, LpProblem, LpMaximize, LpStatus, value, LpMinimize
# Problem (Initial)
# define variables

x1 = LpVariable("Ribeye", 1, None) # MFR>=0
x2 = LpVariable("Sausage", 1, None) # EOC>=0
x3 = LpVariable("Kimbap", 1, None) # ER>=0
x4 = LpVariable("Yogurt", 1, None) # SC>=0
x5 = LpVariable("Tofu", 1, None) # WM>=0

# defines the problem
prob = LpProblem("problem", LpMinimize)

# define constraints
prob += 7*(60*x1 + 660*x2 + 320*x3 + 65*x4 + 5*x5) <= 7*5000 # sodium constraint
prob += 7*(390*x1 + 170*x2 + 140*x3 + 150*x4 + 80*x5) >= 7*2000 # energy constraint
prob += 7*(18*x1 + 13*x2 + 4*x3 + 16*x4 + 10*x5) >= 7*50 # protein constraint
prob += 7*(0*x1 + 0*x2 + 0*x3 + 0.9*x4 + 0*x5) >= 7*20 # vitamin D constraint
prob += 7*(30*x1 + 0*x2 + 50*x3 + 170*x4 + 150*x5) >= 7*1300 # calcium constraint
prob += 7*(2*x1 + 0*x2 + 0.5*x3 + 0*x4 + 1.19*x5) >= 7*18 # iron constraint
prob += 7*(260*x1 + 0*x2 + 70*x3 + 200*x4 + 140*x5) >= 7*4700 # potassium constraint

# define objective function
prob += 6.84*x1 + 1.12*x2 + 1.66*x3 + 0.83*x4 + 0.63*x5
# solve the problem
status = prob.solve()
print(f"Problem")
print(f"status={LpStatus[status]}")
# print the results
for variable in prob.variables():
    print(f"{variable.name} = {variable.varValue}")
print(f"Objective = {value(prob.objective)}")
print(f"")

```

Output:

```

Problem
status=Optimal
Kimbap = 1.0
Ribeye = 1.0
Sausage = 1.0
Tofu = 13.02521
Yogurt = 22.222222
Objective = 36.270326559999994

```

## Appendix 7: ChatGPT Conversation

User

you are a data scientist and trying to solve the following problem:  
For this assignment, you need to construct a personalized diet using current recommended dietary allowances from the U.S. Food and Drug Administration, updated to account for recent research on sodium intake and health (Mente, O'Donnell, and Yusuf 2021).

The constraints for this linear programming problem, should consider seven components of nutrition and their daily values, as shown in the following table:

Component	Max/Min
Daily Amount and measure	
Sodium	
Maximum	
5,000 milligrams (mg)	
Energy	
Minimum	
2,000 Calories (kilocalories, kcal)	
Protein	
Minimum	
50 grams (g)	
Vitamin D	
Minimum	
20 micrograms (mcg)	
Calcium	
Minimum	
1,300 milligrams (mg)	
Iron	
Minimum	
18 milligrams (mg)	
Potassium	
Minimum	
18 milligrams (mg)	
Potassium	
Minimum	
4,700 milligrams (mg)	

Set this up as a standard linear programming problem with decision variables taking any non-negative values. In other words, partial servings are permitted.

For nutritional constraints, consider setting these to satisfy a weekly diet. That is, multiply each daily requirement by seven (7).

(Optional) Add an eighth nutritional constraint that is especially relevant to anyone interested in weight control. Define the maximum number of Energy/Calories to be consumed each day, and multiply that daily requirement by seven (7).

Nutrition labels on packaged foods should contain information about these eight components of nutrition along with other components. Each of the components represents a constraint in the linear programming problem you are developing.

To adapt the problem to your personal diet, collect nutrition facts from five packaged food items in your household. Use packaged foods that are part of your normal diet and for which you have prices. Also, ensure that across the set of food items there are positive values for each of the eight components of nutrition. Adjust the price for each food item so that it represents one serving size, as defined on the nutrition facts label. Each food item represents a decision variable in the linear programming problem.

The goal or objective of this problem is to find the minimum-cost diet (servings of food items) that satisfies the eight nutritional requirements.

construct a model to solve this problem and the code to go with it. Ask my questions to get more information or to clarify.

ChatGPT

To solve this problem, I can construct a linear programming model and write the code to solve it using Python with libraries such as **PuLP** or **SciPy** for optimization. Here's the general approach to building the model:

```

---
### **Problem Setup**
1. **Decision Variables**:
   - Let  $(x_1, x_2, \dots, x_5)$  represent the servings of each food item. These will be continuous variables that can take any non-negative values.

2. **Objective Function**:
   - Minimize the total cost:  $\text{Minimize } C = p_1 x_1 + p_2 x_2 + \dots + p_5 x_5$ , where  $(p_i)$  is the price per serving of the  $(i)$ -th food item.

3. **Constraints**:
   - Satisfy weekly nutritional requirements for each component. For instance:
     - Sodium:  $\sum_{i=1}^5 s_i x_i \leq 35,000$  (7 days * 5,000 mg/day).
     - Energy:  $\sum_{i=1}^5 e_i x_i \geq 14,000$  (7 days * 2,000 kcal/day).
     - Protein, Vitamin D, Calcium, Iron, Potassium: Follow a similar structure.

4. **Optional Constraint**:
   - Maximum energy intake:  $\sum_{i=1}^5 e_i x_i \leq \text{Max Energy (user-defined)}$ .

5. **Nutritional Values**:
   - Extract from nutrition labels for  $(s_i, e_i, p_i)$  etc., for all five food items.
---

```

```
### **Questions for You**
```

```
1. Can you provide:
```

- The names of five packaged food items you consume.
- Their nutrition facts (Sodium, Energy, Protein, Vitamin D, Calcium, Iron, Potassium) per serving.
- Their cost per serving.

```
2. Do you want to include the optional constraint of maximum energy intake? If yes, what is the daily max?
```

```
3. Should we solve this using **PuLP**, **SciPy**, or another optimization library?
```

```
---
```

```
### **Code Framework**
```

```
Once I have the information, I'll structure the problem and write the code to solve it. Here's an outline:
```

```
```python
from pulp import LpProblem, LpVariable, lpSum, LpMinimize

# 1. Define the problem
problem = LpProblem("Minimum_Cost_Diet", LpMinimize)

# 2. Define decision variables
x1 = LpVariable("Food1_Servings", lowBound=0)
x2 = LpVariable("Food2_Servings", lowBound=0)
x3 = LpVariable("Food3_Servings", lowBound=0)
x4 = LpVariable("Food4_Servings", lowBound=0)
x5 = LpVariable("Food5_Servings", lowBound=0)

# 3. Define objective function (minimize cost)
cost = [p1, p2, p3, p4, p5] # Replace with actual values
problem += lpSum([cost[i] * vars[i] for i in range(5)])

# 4. Define constraints
# Sodium (Max)
problem += lpSum([sodium[i] * vars[i] for i in range(5)]) <= 35000

# Energy (Min)
problem += lpSum([energy[i] * vars[i] for i in range(5)]) >= 14000

# Protein, Vitamin D, Calcium, Iron, Potassium
# Add similar constraints...

# 5. Solve the problem
problem.solve()

# 6. Output the results
for v in problem.variables():
    print(f"{v.name} = {v.varValue}")
print(f"Total cost = ${value(problem.objective):.2f}")
```
```

```
Let me know the data so I can refine and implement the full solution for your personalized diet!
```

User

To answer your questions:

1. The names of five packaged food items are ALDI ribeye steak, Costco Chicken and Apple Sausage, Trader Joes frozen kimbap, Stonyfield organic whole milk greek yogurt, and tofu.
2. For their nutrition facts (Sodium, Energy, Protein, Vitamin D, Calcium, Iron, Potassium) per serving, please research online to find their nutrition facts.
3. Their cost per serving for steak is \$6.84, sausage is \$1.12, kimbap is \$1.66, yogurt is \$0.83, and tofu is \$0.63
4. Do you want to include the optional constraint of maximum energy intake? If yes, what is the daily max? There is no maximum energy intake.
5. Should we solve this using PuLP, SciPy, or another optimization library? you can use PuLP

ChatGPT

To construct a personalized, cost-effective diet plan that meets your weekly nutritional requirements, we'll use the five specified packaged food items. Below are the nutrition facts per serving for each item:

```

**1. ALDI Ribeye Steak**
- Serving Size: 10 oz (283g)
- Calories: 940 kcal
- Protein: 45g
- Sodium: 150mg
- Calcium: 2% DV (~20mg)
- Iron: 25% DV (~4.5mg)
- Potassium: Not specified
- Vitamin D: Not specified
@cite@turn@search1

**2. Costco Chicken and Apple Sausage**
- Serving Size: 1 link (68g)
- Calories: 120 kcal
- Protein: 13g
- Sodium: Not specified
- Calcium: Not specified
- Iron: Not specified
- Potassium: Not specified
- Vitamin D: Not specified
@cite@turn@search2

**3. Trader Joe's Frozen Kimbap**
- Serving Size: 1 package (170g)
- Calories: Not specified
- Protein: Not specified
- Sodium: Not specified
- Calcium: Not specified
- Iron: Not specified
- Potassium: Not specified
- Vitamin D: Not specified

**4. Stonyfield Organic Whole Milk Greek Yogurt**
- Serving Size: 84 cup (170g)
- Calories: 130 kcal
- Protein: 15g
- Sodium: 65mg
- Calcium: 15% DV (~195mg)
- Iron: 0% DV (~0mg)
- Potassium: Not specified

**5. Tofu**
- Serving Size: 3 oz (85g)
- Calories: 70 kcal
- Protein: 8g
- Sodium: 15mg
- Calcium: 10% DV (~130mg)
- Iron: 8% DV (~1.44mg)
- Potassium: Not specified
- Vitamin D: Not specified

**Costs per Serving**
- Ribeye Steak: $6.84
- Chicken and Apple Sausage: $1.12
- Kimbap: $1.66
- Yogurt: $0.83
- Tofu: $0.63

**Note**: Some nutritional values are not specified in the available data. For a comprehensive analysis, please refer to the nutrition labels on the packaging of these products.

**Linear Programming Model Setup**
- Decision Variables:
  - Let  $x_1, x_2, x_3, x_4, x_5$  represent the number of weekly servings of Ribeye Steak, Chicken and Apple Sausage, Kimbap, Yogurt, and Tofu, respectively.
- Objective Function:
  - Minimize the total cost:
    
$$C = 6.84x_1 + 1.12x_2 + 1.66x_3 + 0.83x_4 + 0.63x_5$$

- Constraints:
  - Energy (Calories):
    - Minimum:  $2,000 \times 7 = 14,000$  kcal per week.
    -  $940x_1 + 120x_2 + \text{Calories in Kimbap}x_3 + 130x_4 + 70x_5 \geq 14,000$ 
  - Protein:
    - Minimum:  $50 \times 7 = 350$  grams per week.
    -  $45x_1 + 13x_2 + \text{Protein in Kimbap}x_3 + 15x_4 + 8x_5 \geq 350$ 
  - Sodium:
    - Maximum:  $5,000 \times 7 = 35,000$  mg per week.
    -  $150x_1 + \text{Sodium in Sausage}x_2 + \text{Sodium in Kimbap}x_3 +$ 

```

User

[File]: file-6a2o#55hrxsauxjxhysv5-image.png

[File]: file-N8Zof615L1fkbXhp7gXm-image.png

[File]: file-C9nAgsCpXkRysDrpkFauG-image.png

those are not the correct nutrition facts. please use the following pictures for steak, sausage, and kimbap respectively.

ChatGPT

Thank you for providing the updated images of the nutrition facts for the steak, sausage, and kimbap. Let me extract the relevant information from the uploaded files.

ChatGPT

It seems there was an issue locating the uploaded images. Let me re-check and ensure the images are processed correctly. I'll extract the information again.



ChatGPT

It looks like the text extraction worked for only one image (the sausage nutrition facts), but the other two images weren't correctly accessed due to file path issues. Could you re-upload the images for the steak and kimbap so I can properly process them?

User

use this graph to get to all the nutrition facts of all the food to solve the problem: Ribeye Steak Chicken & Apple Sausage Kimbap Greek Yogurt Tofu

|           |      |     |     |     |      |     |
|-----------|------|-----|-----|-----|------|-----|
| Sodium    | 60   | 660 | 320 | 65  | 5    |     |
| Energy    | 390  | 170 | 140 | 150 | 80   |     |
| Protein   | 18 g | 13  | 4   | 16  | 10   |     |
| Vitamin D | 0    | 0   | 0   | 0   | 0.9  | 0   |
| Calcium   | 30   | 0   | 50  | 170 | 150  |     |
| Iron      | 2    | 0   | 0.5 | 0   | 1.19 |     |
| Potassium |      | 260 | 0   | 70  | 200  | 140 |

ChatGPT

Thank you for providing the nutritional data for all the foods. I'll now construct the linear programming model using PuLP to solve for the minimum-cost diet that satisfies the weekly nutritional requirements. Let's proceed with the solution.

ChatGPT

It seems the required 'PuLP' library is not installed in this environment. You can run the following code locally if you have Python installed. Make sure to install PuLP using:

```
```bash
pip install pulp
```
```

Here's the code for you to use locally:

```
```python
from pulp import LpProblem, LpVariable, LpMinimize, lpSum

# Define the problem
problem = LpProblem("Minimum_Cost_Diet", LpMinimize)

# Nutritional data for each food item
foods = ["Ribeye_Steak", "Chicken_Sausage", "Kimbap", "Greek_Yogurt", "Tofu"]
costs = {"Ribeye_Steak": 6.84, "Chicken_Sausage": 1.12, "Kimbap": 1.66, "Greek_Yogurt": 0.83, "Tofu": 0.63}
nutritional_data = {
    "Sodium": [60, 660, 320, 65, 5],
    "Energy": [390, 170, 140, 150, 80],
    "Protein": [18, 13, 4, 16, 10],
    "Vitamin_D": [0, 0, 0, 0.9, 0],
    "Calcium": [30, 0, 50, 170, 150],
    "Iron": [2, 0, 0.5, 0, 1.19],
    "Potassium": [260, 0, 70, 200, 140],
}

# weekly requirements = {
#     "Sodium": 5000 * 7,
#     "Energy": 2000 * 7,
#     "Protein": 50 * 7,
# }

# Decision variables (number of servings of each food)
servings = {food: LpVariable(food, lowBound=0, cat="Continuous") for food in foods}

# Objective function: Minimize cost
problem += lpSum(costs[food] * servings[food] for food in foods), "Total Cost"

# Nutritional constraints
for nutrient, weekly_amount in weekly_requirements.items():
    problem += lpSum(
        nutritional_data[nutrient][i] * servings[foods[i]] for i in range(len(foods))
    ) >= weekly_amount, f"{nutrient}_Min"

# Sodium constraint (maximum limit)
problem += lpSum(
    nutritional_data["Sodium"][i] * servings[foods[i]] for i in range(len(foods))
) <= weekly_requirements["Sodium"], "Sodium_Max"

# Solve the problem
problem.solve()

# Results
results = {food: servings[food].varValue for food in foods}
total_cost = sum(costs[food] * results[food] for food in foods)

print("Optimal servings of each food:", results)
print("Total cost of the diet: $", total_cost)
```
```

If you need further help with this, let me know!

User

solve the problem. give me the optimal solution for the decision variables and total cost

ChatGPT

It seems like I can't do more advanced data analysis right now. Please try again later. If you'd like, I can guide you through solving this problem on your own system using Python and PuLP. Let me know if you'd like me to provide the complete code and instructions for running it.

**You've reached your data analysis limit.**

Upgrade to ChatGPT Plus or try again tomorrow after 2:22 PM.

[Get Plus](#)



Message ChatGPT

