

Part 1

Shrimp



Broccoli



Chicken breast



Salmon



Dumplings



Price calculation per serving

Shrimp: 13.99 / 7 servings = \$2.00

Broccoli: 8.99 /21 servings = \$0.43

Chicken Breast: 15 /20 servings = \$0.75

Salmon: 33.99/ 7 servings = \$4.85

Dumplings: 14.99/13 servings = \$1.15

Part 2

Decision Variables:

Let x_1, x_2, x_3, x_4, x_5 represent the quantities (in units) of Shrimp, Broccoli, Chicken Breast, Salmon, and Dumplings, respectively, that should be consumed over a week.

Objective Function:

Minimize the total weekly cost:

$$Z = 2x_1 + 0.43x_2 + 0.75x_3 + 4.85x_4 + 1.15x_5$$

Subject to weekly Nutritional Constraints:

Sodium: $310x_1 + 25x_2 + 580x_3 + 350x_4 + 330x_5 \leq 35000 \text{ mg}$

Energy: $90x_1 + 25x_2 + 210x_3 + 400x_4 + 180x_5 \geq 14000 \text{ kcal}$

Protein: $22x_1 + 0x_2 + 20x_3 + 35x_4 + 10x_5 \geq 350 \text{ g}$

Vitamin D: $0x_1 + 0x_2 + 0x_3 + 12.7x_4 + 0x_5 \geq 140 \text{ mcg}$

Calcium: $60x_1 + 40x_2 + 0x_3 + 10x_4 + 11x_5 \geq 9100 \text{ mg}$

Iron: $0x_1 + 0.7x_2 + 1x_3 + 0x_4 + 2x_5 \geq 126 \text{ mg}$

Potassium: $125x_1 + 280x_2 + 220x_3 + 710x_4 + 171x_5 \geq 32900 \text{ mg}$

Non-negative Constraints:

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

Description:

The goal of this linear programming problem is to figure out the best weekly diet plan that costs as little as possible. I have choosing how many units of five different foods to eat over a week—Shrimp, Broccoli, Chicken Breast, Salmon, and Dumplings. Each food comes with its own cost per unit, nutritional values for things like energy, protein, and vitamins, and we've set some rules based on what our body needs in a week.

We want to spend the least amount of money while making sure we get enough of each nutrient our body needs every week. These nutrients include things like energy (calories), protein, and minerals like calcium and iron. There's also a limit on how much sodium we can have, to make sure we're not eating too much salt.

We're treating this like a puzzle, where we can only buy whole units of each food (so no half a chicken breast, for example), and we're trying to fit all these pieces together in the best way. Our aim is to solve this puzzle in such a way that we get a shopping list of exactly how many of each food item to buy for the whole week that keeps us healthy and doesn't break the bank.

Part 3 Code

See Appendix 1a

Output:

Foods_Broccoli = 226.0 units for the week

Foods_Chicken_Breast = 17.0 units for the week

Foods_Dumplings = 0.0 units for the week

Foods_Salmon = 12.0 units for the week

Foods_Shrimp = 0.0 units for the week

Weekly Total Cost for Shrimp = 0.0

Weekly Total Cost for Broccoli = 97.18

Weekly Total Cost for Chicken Breast = 12.75

Weekly Total Cost for Salmon = 58.2

Weekly Total Cost for Dumplings = 0.0

Overall Weekly Total Cost = 168.13

Based on the output, I need 226 units of broccoli, 17 units of chicken breast, and 12 units of salmon for the week. The minimum cost is \$168.13. The split will be like \$97.18 for broccoli, \$12.75 for chicken breast, and \$ 58.2 for salmon.

Part 4

To incorporate the requirement that at least one serving of each food item must be consumed during the week into the linear programming (LP) model, I need to adjust the lower bounds for each food item's decision variable from 0 to 1. "food_vars = pulp.LpVariable.dicts("Foods", food_items, lowBound=1, cat=pulp.LpInteger)"

After the changes, the total cost goes up 1 dollar to \$169.24.

The breakdown is below:

Foods_Broccoli = 223.0 units for the week , Foods_Chicken_Breast = 16.0 units for the week , Foods_Dumplings = 1.0 units for the week, Foods_Salmon = 12.0 units for the week, Foods_Shrimp = 1.0 units for the week

Weekly Total Cost for Shrimp = 2.0, Weekly Total Cost for Broccoli = 95.89, Weekly Total Cost for Chicken Breast = 12.0, Weekly Total Cost for Salmon = 58.2, Weekly Total Cost for Dumplings = 1.15

Overall Weekly Total Cost = 169.24

Part 5

I tried to use Google Gemini for this assignment. My first prompt is "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 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)."

It generates python codes below:

Please check Appendix 2a

It basically generates the whole project structure for me and I can just slightly twist it and it will generate all the requirements I need for this assignment. LLM agent does able to assist and complete this assignment.

Appendix 1a

```
import pulp

problem = pulp.LpProblem("Weekly_Nutrition_Optimization", pulp.LpMinimize)

food_items = ['Shrimp', 'Broccoli', 'Chicken Breast', 'Salmon', 'Dumplings']

food_vars = pulp.LpVariable.dicts("Foods", food_items, lowBound=0, cat=pulp.LpInteger)

costs = {'Shrimp': 2, 'Broccoli': 0.43, 'Chicken Breast': 0.75, 'Salmon': 4.85, 'Dumplings': 1.15}

problem += pulp.lpSum([costs[i] * food_vars[i] for i in food_items]), "Weekly Total Cost"

sodium_content = {'Shrimp': 310, 'Broccoli': 25, 'Chicken Breast': 580, 'Salmon': 350, 'Dumplings': 330}

energy_content = {'Shrimp': 90, 'Broccoli': 25, 'Chicken Breast': 210, 'Salmon': 400, 'Dumplings': 180}

protein_content = {'Shrimp': 22, 'Broccoli': 0, 'Chicken Breast': 20, 'Salmon': 35, 'Dumplings': 10}

vitaminD_content = {'Shrimp': 0, 'Broccoli': 0, 'Chicken Breast': 0, 'Salmon': 12.7, 'Dumplings': 0}

calcium_content = {'Shrimp': 60, 'Broccoli': 40, 'Chicken Breast': 0, 'Salmon': 10, 'Dumplings': 11}

iron_content = {'Shrimp': 0, 'Broccoli': 0.7, 'Chicken Breast': 1, 'Salmon': 0, 'Dumplings': 2}

potassium_content = {'Shrimp': 125, 'Broccoli': 280, 'Chicken Breast': 220, 'Salmon': 710, 'Dumplings': 171}

problem += pulp.lpSum([sodium_content[i] * food_vars[i] for i in food_items]) <= 5000 * 7, "MaxSodiumWeekly"

problem += pulp.lpSum([energy_content[i] * food_vars[i] for i in food_items]) >= 2000 * 7, "MinEnergyWeekly"

problem += pulp.lpSum([protein_content[i] * food_vars[i] for i in food_items]) >= 50 * 7, "MinProteinWeekly"

problem += pulp.lpSum([vitaminD_content[i] * food_vars[i] for i in food_items]) >= 20 * 7, "MinVitaminDWeekly"

problem += pulp.lpSum([calcium_content[i] * food_vars[i] for i in food_items]) >= 1300 * 7, "MinCalciumWeekly"

problem += pulp.lpSum([iron_content[i] * food_vars[i] for i in food_items]) >= 18 * 7, "MinIronWeekly"

problem += pulp.lpSum([potassium_content[i] * food_vars[i] for i in food_items]) >= 4700 * 7, "MinPotassiumWeekly"

problem.solve()

for v in problem.variables():

    print(f'{v.name} = {v.varValue} units for the week')

for food in food_items:

    total_cost = food_vars[food].value() * costs[food]

    print(f'Weekly Total Cost for {food} = {round(total_cost,2)}')

print("Overall Weekly Total Cost =", pulp.value(problem.objective))
```

Appendix 2a

```
from pulp import LpProblem, LpVariable

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

# Define decision variables (amount of each component consumed per week)
sodium = LpVariable("Sodium", lowBound=0)
energy = LpVariable("Energy", lowBound=0)
protein = LpVariable("Protein", lowBound=0)
vitamin_d = LpVariable("VitaminD", lowBound=0)
calcium = LpVariable("Calcium", lowBound=0)
iron = LpVariable("Iron", lowBound=0)
potassium = LpVariable("Potassium", lowBound=0)

# Define weekly nutritional requirements (daily value * 7)
weekly_sodium_max = 5000 * 7
weekly_energy_min = 2000 * 7
weekly_protein_min = 50 * 7
weekly_vitamin_d_min = 20 * 7
weekly_calcium_min = 1300 * 7
weekly_iron_min = 18 * 7
weekly_potassium_min = 4700 * 7

# Define the objective function (minimize total consumption)
problem += sodium + energy + protein + vitamin_d + calcium + iron + potassium

# Define constraints
problem += sodium <= weekly_sodium_max
problem += energy >= weekly_energy_min
problem += protein >= weekly_protein_min
problem += vitamin_d >= weekly_vitamin_d_min
```

```
problem += calcium >= weekly_calcium_min
problem += iron >= weekly_iron_min
problem += potassium >= weekly_potassium_min

# Solve the problem
problem.solve()

# Print the optimal solution
print("Optimal Solution:")
print(f"Sodium: {sodium.value():.2f} milligrams")
print(f"Energy: {energy.value():.2f} Calories")
print(f"Protein: {protein.value():.2f} grams")
print(f"Vitamin D: {vitamin_d.value():.2f} micrograms")
print(f"Calcium: {calcium.value():.2f} milligrams")
print(f"Iron: {iron.value():.2f} milligrams")
print(f"Potassium: {potassium.value():.2f} milligrams")
```