

Mike Mistarz – MSDS460 – Assignment 1 – The Diet Problem

In setting up the diet problem, I wanted to try to make it as personalized as possible to how I set up my nutrition for a given week. With working full time, having a 12-month old child who is learning to discover new foods and a wife who works in the fitness industry, we try to balance cost-effectiveness and time-efficiency while focusing on whole foods and providing optimal solution for ourselves and our growing son. So, every week we use frozen vegetables, olive oil, salt and mushrooms to make a week's worth of vegetables that we use to make other food items. The other main drivers of our nutritional needs are milk, eggs, chicken and rice. So, in setting up the diet problem, I set up a week's worth of vegetables as one variable along with the other four food items listed to fill out the five variables.

In setting up the problem, the nutritional values for each of the seven constraints were used as the constraints in the problem and cost was used as the objective function to try to determine how to meet the nutritional needs in the most cost-effective way. In calculating the nutritional values of the vegetables, I set vegetables as a single serving item and added up all the ingredients used for both the nutritional values and the costs, while using single servings of everything else. This was done because vegetables make up the bulk of our nutrition and the other items are used to fill in the rest where they are needed. I used python pulp library to set up the problem and output a solution.

In the first iteration of the program, it suggested that we use 290 eggs per week and 200 servings of rice, ignoring all other items to get the most cost-effective solution. This, however, is both impractical and unhealthy, resulting in a weekly calorie total of 52300 calories and a total cost of \$68.21. I changed the minimum values for each of the food items to ensure one of each was chosen which gave better results but still over 4000 calories per day. It only chose one of each of the milk, chicken and vegetables and 214.5 eggs and 62.3 servings of rice for a total cost

Mike Mistarz – MSDS460 – Assignment 1 – The Diet Problem

of \$76.12. In trying to get one serving of milk and one serving of chicken per day, I upped those minimum constraints to 7 each for the week. Once again, it suggested 1 serving of vegetables, 7 servings of milk, 7 servings of chicken, 137.9 eggs, and 71.2 servings of rice per week. This led to a calorie total of 26796.4 calories and \$96.80 per week. As this is still impractical, I set limits of 3 servings of rice per day and 5 eggs per day. This led to a realistic solution of 1.96 servings of vegetables, 10 servings of milk, 35 eggs, 1 serving of chicken and 13.8 servings of rice. The total calories for the realistic solution were 14000 and the total cost was \$109.36.

The last portion of the assignment had us converse with ChatGPT to solve our diet problem. It was impressive how adept the software was at breaking down the problem and providing code snippets. I set a lower bound of 1 for each of the variables and gave it the nutritional needs for each of the seven constraints. This resulted in a slightly more expensive result than my solution, with a suggestion of 1 serving of each of chicken, milk and vegetables with 212 eggs and 99.4 servings of rice, resulting in 35000 calories per week and a cost of \$79.42.

Mike Mistarz – MSDS460 – Assignment 1 – The Diet Problem

Appendix I: Foods used



Mike Mistarz – MSDS460 – Assignment 1 – The Diet Problem



Mike Mistarz – MSDS460 – Assignment 1 – The Diet Problem



Nutrition Facts	
about 12 servings per container	
Serving size	4 oz (112g)
Amount per serving	
Calories	110
% Daily Value*	
Total Fat 2.5g	3%
Saturated Fat 0.5g	3%
Trans Fat 0g	
Cholesterol 65mg	22%
Sodium 220mg	10%
Total Carbohydrate 0g	0%
Dietary Fiber 0g	0%
Total Sugars 0g	
Includes 0g Added Sugars	0%
Protein 23g	
Vitamin D 0mcg	0%
Calcium 0mg	0%
Iron 0.7mg	4%
Potassium 250mg	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	

Mike Mistarz – MSDS460 – Assignment 1 – The Diet Problem



Appendix II: Weekly Food Nutrition Table

Item	Calories	Protein	Sodium	Potassium	Iron	Vitamin D	Calcium	Cost (\$)
Peas	280	20	0	840	4	0	80	1.29
Carrots	140	4	160	0	0	0	80	1.29
Carrots	140	4	160	0	0	0	80	1.29
Edamame	480	40	40	1840	8	0	400	1.99
Edamame	480	40	40	1840	8	0	400	1.99
Edamame	480	40	40	1840	8	0	400	1.99
Mushrooms	200	20	50	3800	0	100	0	11.18
Spinach	100	12	260	1160	8	0	440	1.29
Green Beans	140	8	0	640	4	0	160	0.99
Sweet Potato	360	4	340	1200	2.8	0	120	0.99
Olive Oil	1200	0	0	0	0	0	0	4.5
Salt	0	0	6480	0	0	0	0	0.02
Veggie Totals	4000	192	7570	13160	42.8	100	2160	28.81
Milk	140	13	280	550	1	5	380	4.49
Eggs	70	6	70	70	0.9	1	30	0.16625
Chicken	110	23	220	250	0.7	0	0	0.93
Rice	160	3	5	63	0	0	2	0.1
Weekly Totals	14000	350	35000	32900	126	140	9100	
	Min	Min	Max	Min	Min	Min	Min	

Appendix III: ChatGPT solution discussion

<https://chat.openai.com/share/f69481fc-d531-4688-bb99-689519599272>



You

Can you help me minimize costs while meeting my nutritional needs>



ChatGPT

Absolutely, I'd be happy to help you with that! There are several strategies you can use to minimize costs while still meeting your nutritional needs:

1.

Plan Your Meals: Planning your meals in advance can help you buy only what you need, reducing food waste and saving money.

2.

Buy in Bulk: Buying items like grains, beans, and frozen vegetables in bulk can often be cheaper than buying smaller quantities.

3.

Cook at Home: Eating out can be much more expensive than cooking at home. Try to cook most of your meals at home to save money.

4.

Focus on Staple Foods: Staples like rice, beans, pasta, and oats are inexpensive and can be used in a variety of dishes.

5.

Use Coupons and Discounts: Keep an eye out for coupons and discounts on the foods you regularly buy. Many grocery stores offer loyalty programs that can help you save money.

6.

Buy Seasonal Produce: Seasonal fruits and vegetables are often cheaper and fresher than out-of-season produce.

7.

Limit Processed Foods: Processed and convenience foods are often more expensive than whole foods. Try to limit these items to save money.

8.

Compare Prices: Compare prices at different stores to find the best deals. You may find that certain items are cheaper at one store compared to another.

9.

Consider Generic Brands: Generic or store-brand products are often cheaper than name-brand items and can be just as nutritious.

10.

Use Leftovers: Don't let leftovers go to waste. Incorporate them into future meals to save both money and time.

11.

Grow Your Own Produce: If you have space, consider growing your own fruits, vegetables, and herbs. This can save you money and provide you with fresh, nutritious produce.

By following these tips, you can minimize costs while still meeting your nutritional needs. If you have specific dietary requirements or preferences, feel free to let me know, and I can tailor the advice accordingly!



You

Can you write a program for me that will help me calculate how to minimize my costs while maximizing my nutritional needs if I provide the constraints?



ChatGPT

Certainly! To create a program for optimizing your nutrition while minimizing costs, we would typically use linear programming techniques. However, implementing such a program would require detailed knowledge of your specific constraints, such as dietary requirements, food preferences, budget constraints, and available food options.

Here's a general outline of how you might approach creating such a program:

1.

Define your objective function: This function would represent the goal you want to achieve, such as maximizing nutritional value while minimizing costs.

Mike Mistarz – MSDS460 – Assignment 1 – The Diet Problem

2.

Identify decision variables: These are the variables that you can control to achieve your objective. In this case, they might represent the quantities of different foods to include in your diet.

3.

Set constraints: Define the constraints that restrict the values of your decision variables. These constraints might include nutritional requirements (e.g., minimum protein intake, maximum calorie limit) and budget constraints.

4.

Implement the optimization algorithm: Use a linear programming solver or optimization library to find the optimal values of the decision variables that satisfy the objective function while adhering to the constraints.

Here's a simplified Python example using the PuLP library, assuming you have a list of foods with their nutritional information and prices:

```
python Copy code

from pulp import LpMaximize, LpProblem, LpVariable

# Define the foods available with their nutritional information and prices
foods = {
    'apple': {'cost': 0.5, 'calories': 50, 'protein': 1, 'fat': 0, 'carbs': 15},
    'banana': {'cost': 0.3, 'calories': 100, 'protein': 1, 'fat': 0, 'carbs': 27},
    # Add more foods with their nutritional information
}

# Define the constraints
max_calories = 2000
max_cost = 10

# Create a LP minimization problem
prob = LpProblem("Nutrition Optimization", LpMaximize)

# Create decision variables
food_vars = LpVariable.dicts("Food", foods, lowBound=0, cat='Continuous')

# Define the objective function
prob += sum(foods[food]['protein'] * food_vars[food] for food in foods)

# Add constraints
prob += sum(foods[food]['calories'] * food_vars[food] for food in foods) <= max_calories
prob += sum(foods[food]['cost'] * food_vars[food] for food in foods) <= max_cost

# Solve the optimization problem
prob.solve()

# Print the results
print("Optimal food choices:")
for food in foods:
    if food_vars[food].value() > 0:
        print(f"{food}: {food_vars[food].value()}")
```

Mike Mistarz – MSDS460 – Assignment 1 – The Diet Problem

This is a basic example, and you would need to expand it with more foods, nutritional information, and constraints as per your specific requirements. Additionally, you might need to explore more sophisticated optimization techniques and libraries depending on the complexity of your problem.



You

I would like to minimize my costs but would like to at least use one of every food item in my meal plan for the week. Here is a python dictionary of the food items to help you along the way.

```
food_items = {
    "Veggie mix": {
        "calories": 4000,
        "protein": 192,
        "sodium": 7570,
        "potassium": 13160,
        "iron": 42.8,
        "vitamin_d": 100,
        "calcium": 2160,
        "cost": 28.81,
    },
    "Milk": {
        "calories": 140,
        "protein": 13,
        "sodium": 280,
        "potassium": 550,
        "iron": 1,
        "vitamin_d": 5,
        "calcium": 380,
        "cost": 4.49,
    },
    "Eggs": {
        "calories": 70,
        "protein": 6,
        "sodium": 70,
        "potassium": 70,
        "iron": 0.9,
        "vitamin_d": 1,
        "calcium": 30,
        "cost": 0.16625,
    },
}
```

Mike Mistarz – MSDS460 – Assignment 1 – The Diet Problem

```
"Chicken": {  
  "calories": 110,  
  "protein": 23,  
  "sodium": 220,  
  "potassium": 250,  
  "iron": 0.7,  
  "vitamin_d": 0,  
  "calcium": 0,  
  "cost": 0.93,  
},  
"Rice": {  
  "calories": 160,  
  "protein": 3,  
  "sodium": 5,  
  "potassium": 63,  
  "iron": 0,  
  "vitamin_d": 0,  
  "calcium": 2,  
  "cost": 0.10,  
},  
}
```

I need to make sure I reach 35000 calories, 350 grams of protein, 32900 mg potassium, 126 mg iron, 140 mcg Vitamin D and 9100 mg of Calcium. I can only have 35000 mg of sodium. Can you use these constraints to provide an ideal solution while ensuring to minimize costs? Can you also provide the nutritional values for each of the targets provided?

< 4 / 4 >




ChatGPT

To solve this optimization problem, we can use linear programming to minimize the total cost while satisfying the given constraints. Here's a Python implementation using the PuLP library:

Mike Mistarz – MSDS460 – Assignment 1 – The Diet Problem

python

 Copy code

```
from pulp import LpProblem, LpVariable, LpMinimize

# Define food items
food_items = {
    "Veggie mix": {
        "calories": 4000,
        "protein": 192,
        "sodium": 7570,
        "potassium": 13160,
        "iron": 42.8,
        "vitamin_d": 100,
        "calcium": 2160,
        "cost": 28.81,
    },
    "Milk": {
        "calories": 140,
        "protein": 13,
        "sodium": 280,
        "potassium": 550,
        "iron": 1,
        "vitamin_d": 5,
        "calcium": 380,
        "cost": 4.49,
    },
    "Eggs": {
        "calories": 70,
        "protein": 6,
        "sodium": 70,
        "potassium": 70,
        "iron": 0.9,
        "vitamin_d": 1,
        "calcium": 30,
        "cost": 0.16625,
    },
    "Chicken": {
        "calories": 110,
        "protein": 23,
        "sodium": 220,
        "potassium": 250,
        "iron": 0.7,
        "vitamin_d": 0,
        "calcium": 0,
        "cost": 0.93,
    },
}
```

```
"Rice": {
    "calories": 160,
    "protein": 3,
    "sodium": 5,
    "potassium": 63,
    "iron": 0,
    "vitamin_d": 0,
    "calcium": 2,
    "cost": 0.10,
},
}

# Define target nutritional values
target_calories = 35000
target_protein = 350
target_potassium = 32900
target_iron = 126
target_vitamin_d = 140
target_calcium = 9100
target_sodium = 35000

# Create a LP minimization problem
prob = LpProblem("Nutrition Optimization", LpMinimize)

# Define decision variables
food_vars = LpVariable.dicts("Food", food_items, lowBound=0, cat='Continuous')

# Define the objective function (minimize cost)
prob += sum(food_items[food]['cost'] * food_vars[food] for food in food_items)

# Add constraints
prob += sum(food_items[food]['calories'] * food_vars[food] for food in food_items) >=
prob += sum(food_items[food]['protein'] * food_vars[food] for food in food_items) >=
prob += sum(food_items[food]['potassium'] * food_vars[food] for food in food_items) >=
prob += sum(food_items[food]['iron'] * food_vars[food] for food in food_items) >= target_iron
prob += sum(food_items[food]['vitamin_d'] * food_vars[food] for food in food_items) >= target_vitamin_d
prob += sum(food_items[food]['calcium'] * food_vars[food] for food in food_items) >= target_calcium
prob += sum(food_items[food]['sodium'] * food_vars[food] for food in food_items) <= target_sodium

# Add constraint to use at least one of every food item
for food in food_items:
    prob += food_vars[food] >= 1
```

```
# Solve the optimization problem
prob.solve()

# Print the results
print("Optimal food choices:")
for food in food_items:
    if food_vars[food].value() > 0:
        print(f"{food}: {food_vars[food].value()}")

# Print nutritional values for targets
print("\nNutritional values for targets:")
print(f"Total calories: {sum(food_items[food]['calories'] * food_vars[food].value() for food in food_items)}")
print(f"Total protein: {sum(food_items[food]['protein'] * food_vars[food].value() for food in food_items)}")
print(f"Total potassium: {sum(food_items[food]['potassium'] * food_vars[food].value() for food in food_items)}")
print(f"Total iron: {sum(food_items[food]['iron'] * food_vars[food].value() for food in food_items)}")
print(f"Total Vitamin D: {sum(food_items[food]['vitamin_d'] * food_vars[food].value() for food in food_items)}")
print(f"Total calcium: {sum(food_items[food]['calcium'] * food_vars[food].value() for food in food_items)}")
print(f"Total sodium: {sum(food_items[food]['sodium'] * food_vars[food].value() for food in food_items)}")
```

This program will find the optimal combination of food items that minimizes the total cost while meeting all the specified constraints, including using at least one of every food item. It will also print out the total nutritional values to ensure they meet the targets.



Appendix IV: Solution Results and Github Repository

Solution	Calories	Protein	Sodium	Potassium	Iron	Vit. D	Calcium	Cost
Ideal Solution Totals	52300	2340	21300	32900	261	290	9100	\$68.21
1 of each Totals	29231.71	1701.943	23397.43	32900	237.5629	319.5143	9100	\$76.12
7 c Milk + 7 chicken totals	26796.4	1485.12	21080.4	32900	178.828	272.92	9100	\$96.80
Realistic Totals	14000	780.1724	20358.55	34831.422	126	280.7396	9100	\$109.36
ChatGPT 1 of each	35000	1798.494	23409.81	35066.202	235.3348	317.0386	9100	\$79.42

[mistmr7/Assignment1_MSDS460_DietProblem: Diet Problem solved using Pulp from python \(github.com\)](https://github.com/mistmr7/Assignment1_MSDS460_DietProblem)