Aishanee Wijeratna
MSDS 460: Decision Analytics
1 October 2024

**Assignment 1: The Diet Problem**

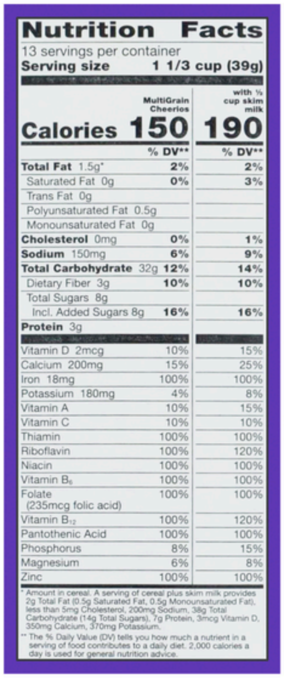
**1. Food Items**

The five food items that I chose for this assignment, as well as their nutrition labels, are as follows. I chose these five foods because they are all staple items in my diet, and I felt they were as nutritionally diverse as possible to best fulfill the constraints of this assignment.

Cauliflower crust pizza ($1.50/serving):


Mediterranean salad kit ($1.33/serving):


Multigrain cheerios + skim milk ($0.60/serving):



# Nutrition Facts

13 servings per container
**Serving size 1 1/3 cup (39g)**

| | MultiGrain Cheerios | with ½ cup skim milk |
|---|---|---|
| **Calories** | **150** | **190** |
| | % DV** | % DV** |
| **Total Fat** 1.5g* | 2% | 2% |
| Saturated Fat 0g | 0% | 3% |
| Trans Fat 0g | | |
| Polyunsaturated Fat 0.5g | | |
| Monounsaturated Fat 0g | | |
| **Cholesterol** 0mg | 0% | 1% |
| **Sodium** 150mg | 6% | 9% |
| **Total Carbohydrate** 32g | 12% | 14% |
| Dietary Fiber 3g | 10% | 10% |
| Total Sugars 8g | | |
| Incl. Added Sugars 8g | 16% | 16% |
| **Protein** 3g | | |
| Vitamin D 2mcg | 10% | 15% |
| Calcium 200mg | 15% | 25% |
| Iron 18mg | 100% | 100% |
| Potassium 180mg | 4% | 8% |
| Vitamin A | 10% | 15% |
| Vitamin C | 10% | 10% |
| Thiamin | 100% | 100% |
| Riboflavin | 100% | 120% |
| Niacin | 100% | 100% |
| Vitamin $B_6$ | 100% | 100% |
| Folate (235mcg folic acid) | 100% | 100% |
| Vitamin $B_{12}$ | 100% | 120% |
| Pantothenic Acid | 100% | 100% |
| Phosphorus | 8% | 15% |
| Magnesium | 6% | 8% |
| Zinc | 100% | 100% |

* Amount in cereal. A serving of cereal plus skim milk provides 2g Total Fat (0.5g Saturated Fat, 0.5g Monounsaturated Fat), less than 5mg Cholesterol, 200mg Sodium, 38g Total Carbohydrate (14g Total Sugars), 7g Protein, 3mcg Vitamin D, 350mg Calcium, 370mg 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.

Kevin's beef & broccoli meal ($3.77/serving):

# Nutrition Facts

About 4.5 servings per container
**Serving Size** **1 Cup (195g)**

**Calories** per serving **180**

| Amount Per Serving | % Daily Value* | Amount Per Serving | % Daily Value* |
|---|---|---|---|
| **Total Fat** 4g | **5%** | **Total Carbohydrate** 19g | **7%** |
| Saturated Fat 1g | **5%** | Dietary Fiber 2g | **9%** |
| Trans Fat 0g | | Total Sugars 12g | |
| **Cholesterol** 40mg | **14%** | Includes 3g Added Sugars | **5%** |
| **Sodium** 810mg | **35%** | **Protein** 17g | |

Vitamin D 1.6mcg 8% • Calcium 80mg 6% • Iron 2.9mg 15% • Potassium 1150mg 25%

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.

Smoked salmon ($4.33/serving):

# Nutrition Facts

About 4 servings per container
**Serving size 2 oz (56g)**

Amount per serving
**Calories 60**

| | % Daily Value* |
|---|---|
| **Total Fat** 0.5g | 1% |
| Saturated Fat 0g | 0% |
| *Trans* Fat 0g | |
| **Cholesterol** 25mg | 8% |
| **Sodium** 540mg | 23% |
| **Total Carbohydrate** 2g | 1% |
| Dietary Fiber 0g | 0% |
| Total Sugars 0g | |
| Includes 0g Added Sugars | 0% |
| **Protein** 12g | |
| Vitamin D 18mcg | 90% |
| Calcium 3mg | 0% |
| Iron 0mg | 0% |
| Potassium 194mg | 4% |

* 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.

INGREDIENTS: SOCKEYE SALMON, SALT, BROWN SUGAR, NATURAL WOOD SMOKE.
CONTAINS: FISH (SOCKEYE SALMON).
Distributed by: Costco Wholesale Corporation P.O. Box 34535 Seattle, WA 98124-1535 USA 1-800-774-2678 • www.costco.com

MAY CONTAIN BONES. PREVIOUSLY FROZEN.
PRODUCT OF THE USA

**2. Linear Programming Problem Set-Up**

After selecting my five foods, I then needed to set up a linear programming problem to find the minimum-cost diet which satisfied the weekly nutritional requirements outlined below:

| *Component* | *Max/Min* | *Daily Amount and measure* |
|---|---|---|
| Sodium | Maximum | 35,000 milligrams (mg) |
| Energy | Minimum | 14,000 Calories (kilocalories, kcal) |
| Protein | Minimum | 350 grams (g) |
| Vitamin D | Minimum | 140 micrograms (mcg) |
| Calcium | Minimum | 9,100 milligrams (mg) |
| Iron | Minimum | 126 milligrams (mg) |
| Potassium | Minimum | 32,900 milligrams (mg) |

I first set up my five foods as my decision variables, making sure that they had a lower bound of 0 (because you cannot have a negative quantity of food) and were continuous variables (to ensure that the problem accounted for partial portions). Each variable (pizza, salad, etc.) represented the number of servings of that food eaten per week.

```
# define variables
pizza = LpVariable("pizza", lowBound=0)
salad = LpVariable("salad", lowBound=0)
cereal = LpVariable("cereal", lowBound=0)
beef = LpVariable("beef", lowBound=0)
salmon = LpVariable("salmon", lowBound=0)
```

Next, I defined my linear programming problem, making sure to specify it as a minimization problem since I was aiming to minimize the total cost of the diet.

```
#define the problem
prob = LpProblem("diet", LpMinimize)
```

Next, I defined my constraint inequalities based on the nutritional guidelines outlined in the table above. For each nutritional constraint, I developed an inequality with the coefficients of each food variable being the amount of that nutrient in a single serving of that food. For example, the pizza has 900 grams of sodium. The right side of each inequality represents either the minimum or maximum amount of that nutrient needed per week, as specified by the weekly nutritional guidelines above.

```
#sodium
prob += 900*pizza + 340*salad + 150*cereal + 810*beef + 540*salmon <= 35000
#energy
prob += 310*pizza + 140*salad + 190*cereal + 180*beef + 60*salmon >= 14000
#protein
prob += 14*pizza + 2*salad + 9*cereal + 17*beef + 12*salmon >= 350
#vitamin d
prob += 0*pizza + 0.1*salad + 3*cereal + 1.6*beef + 18*salmon >= 140
#calcium
prob += 353*pizza + 60*salad + 333*cereal + 80*beef + 3*salmon >= 9100
#iron
prob += 1*pizza + 0.8*salad + 18*cereal + 2.9*beef + 0*salmon >= 126
#potassium
prob += 291*pizza + 240*salad + 360*cereal + 1150*beef + 194*salmon >= 32900
```

Finally, I defined my objective function. This function represented the total cost of my diet, with the coefficients of each variable representing the cost of a single serving of that food item. After completing this set-up process, I was ready to solve the linear programming problem.

```
#define objective function
prob += 1.5*pizza + 1.33*salad + 0.6*cereal + 3.77*beef + 4.33*salmon
```

**3. Initial Solution**

When I used PuLP to solve the linear programming problem as set up above, it was able to find an optimal solution. However, it turned out that the most cost-effective diet that still fulfilled all of the nutritional requirements was exclusively composed of cereal.

```
Problem
status=Optimal
beef = 0.0
Objective = 54.8333334

cereal = 91.388889
Objective = 54.8333334

pizza = 0.0
Objective = 54.8333334

salad = 0.0
Objective = 54.8333334

salmon = 0.0
Objective = 54.8333334
```

Based on this solution, I would need to eat approximately 91.3 servings of cereal and skim milk per week (about 121.8 cups of cereal and 45.6 cups of milk), which would cost me about $54.83 per week. In retrospect, I should have predicted this outcome and potentially chosen a different variety of food options, because cereal was by far the least expensive food per serving and contained all the required nutrients. I will describe my attempt to revise my linear programming model to create a more varied diet in the next section.

**4. Revised Solution - Increasing Dietary Variety**

In my revised solution to increase dietary variety, I updated my variable definitions to specify that a serving of each food needed to be included in my weekly diet at least one time. I accomplished this by changing the lower bound for each food variable from 0 to 1.

```
pizza = LpVariable("pizza", lowBound=1)
salad = LpVariable("salad", lowBound=1)
cereal = LpVariable("cereal", lowBound=1)
beef = LpVariable("beef", lowBound=1)
salmon = LpVariable("salmon", lowBound=1)
```

Even after setting this constraint in an attempt to create a more balanced diet, the optimal cost-minimizing solution was still heavily skewed towards the cereal option. This new solution, in accordance with the new lower bounds for each variable, called for exactly one serving each of pizza, salad, beef & broccoli, and salmon, and 86.2 servings of cereal, for a slightly higher total weekly cost of $62.64. This updated solution was far from a balanced diet, but at least allowed for a little more variation than before.

```
Problem
status=Optimal
beef = 1.0
Objective = 62.638333599999996

cereal = 86.180556
Objective = 62.638333599999996

pizza = 1.0
Objective = 62.638333599999996

salad = 1.0
Objective = 62.638333599999996

salmon = 1.0
Objective = 62.638333599999996
```

Based on the results from this model, it appears that the cereal and skim milk option was far superior to all four of my other chosen food options in terms of nutritional availability and cost-effectiveness. If I were to choose another set of foods to re-do this linear programming problem with, I would take additional care to choose foods of a similar price per serving and more consistent nutritional composition. This would give me a better chance of the model solving for a cost-optimized diet that was truly diverse.

**5. LLM-Generated Solution**

I chose to use ChatGPT (chatgpt.com) to attempt to generate code to solve this problem. I simply copied and pasted the most relevant sections of the assignment description into my ChatGPT input, and the chatbot provided me with an outline of a solution and sample code that was unexpectedly accurate. I specified that the coded solution should utilize PuLP, and the LLM-generated code block was fairly similar to the approach that I took when setting up the LP problem. One interesting difference was that the LLM-generated code used list objects to store the nutritional and cost information for each of the five foods, whereas I simply hard-coded them into the constraints and objective function. I think that storing these values in list objects is a good tactic, for organizational purposes. It makes writing the constraints and objective function simpler and leaves less room for potential errors (for example, accidentally filling in the wrong nutritional value coefficient in one of the constraints). Next time I set up a linear programming problem in Python, I will use this approach.

I was expecting to have to go back and forth with ChatGPT to arrive at a coding solution that was functional and accurate, but I was pleasantly surprised with the accuracy of the agent's first response to my initial prompt. I believe it was the level of detail that I went into in my initial request that allowed for the first response to be so accurate. I do believe that an LLM agent could be used to complete this assignment, if provided a prompt with the proper level of detail. I have included a transcript of my conversation with ChatGPT in the git repo for this assignment.