**PART 1**

Yogurt Spinach Chickpea Patties Nature Valley Breakfast Sausage

A label with information on it

Description automatically generatedA label of baby spinach nutrition facts

Description automatically generatedA close up of a bag of food

Description automatically generatedA close up of a label

Description automatically generatedA package of food with information

Description automatically generated

Extra conversions:

* Spinach % Conversion for Iron: <https://ods.od.nih.gov/factsheets/Iron-HealthProfessional/>
* For a 19-50 year old woman, suggested iron intake is 18mg -> 15%\*(18mg) = 2.70mg
* Spinach % Conversion for Calcium: <https://ods.od.nih.gov/factsheets/Calcium-HealthProfessional/>
* For a 19-50 year old woman, suggested iron intake is 1,000mg -> 8%\*(1,000mg) = 80mg
* Spinach Total Carbohydrates: 3 mg = 0.003 g
* Breakfast Sausage – I eat two rather than 1, all numbers = serving size multiplied by 2
* Costs
  + Nature Valley Peanut Butter Granola Bars
    - $1.10/serving ($5.49/box of 5, 1 serving = 1 bar)
  + Yogurt
    - $1.18/serving ($6.29/32oz, 1 serving = ¾ cup)
  + Vegetarian Sausages
    - $1.66/meal ($4.99/bag of 6, 1 serving = 1 patty, I normally eat 2)
  + Spinach
    - $0.90/serving ($4.49/5oz, 1 serving = 1oz)
  + Chickpea patties
    - $1.15/serving ($4.59/bag of 4, 1 serving = 1 patty)

**PART 2**

Decision Variables:

* x1 = servings of yogurt ($1.18/serving)
* x2 = servings of spinach ($0.90/serving)
* x3 = servings of chickpea patty ($1.15/serving)
* x4 = servings of Nature Valley Bar ($1.10/serving)
* x5 = servings of sausage patty ($1.66/serving, serving will refer to TWO patties)

Objective function in standard form:

* Minimize Z = 1.18\*x1 + 0.9\*x2 + 1.15\*x3 + 1.1\*x4 + 1.66\*x5

Constraints

* Sodium
  + 60\*x1 ​+ 65\*x2 ​+ 380\*x3​ + 160\*x4 ​+ 460\*x5 ​<= 5000
* Calories/Energy
  + 120\*x1 ​+ 20\*x2 ​+ 170\*x3​ + 190\*x4 ​+ 160\*x5 >= 2000
* Protein
  + 15\*x1 ​+ 2\*x2 ​+ 9\*x3​ + 4\*x4 ​+ 18\*x5 ​>= 50
* Vitamin D
  + 0\*x1 ​+ 0\*x2 ​+ 0\*x3​ + 0\*x4 ​+ 0\*x5 ​>= 20
* Calcium
  + 180\*x1 ​+ 80\*x2 ​+ 20\*x3​ + 0\*x4 ​+ 60\*x5 ​>= 1300
* Iron
  + 0.2\*x1 ​+ 2.7\*x2 ​+ 1.2\*x3​ + 0.8\*x4 ​+ 1.8\*x5 ​>= 16
* Potassium
  + 240\*x1 ​+ 0\*x2 ​+ 260\*x3​ + 100\*x4 ​+ 220\*x5 ​>= 4700
* Non negativity Constraints,
  + x1 >= 0, x2 >= 0, x3 >= 0, x4 >= 0, x5 >= 0

In this problem we are trying to minimize the cost for a diet that is made up of five items (Yogurt, Spinach, Chickpea Patty, Nature Valley Bar, and Sausage Patties) while staying within the recommended food guidelines of 5000 mg of Sodium, 2000 Calories (kcal), 50 g of Protein, 20 mcg of Vitamin D, 1300 mg of Calcium, 16 mg of Iron, 4700 mg of Potassium. The above is for DAILY intake. See below for WEEKLY.

Decision Variables:

* x1 = servings of yogurt ($1.18/serving)
* x2 = servings of spinach ($0.90/serving)
* x3 = servings of chickpea patty ($1.15/serving)
* x4 = servings of Nature Valley Bar ($1.10/serving)
* x5 = servings of sausage patty ($1.66/serving, serving will refer to TWO patties)

Objective function in standard form:

* Minimize Z = 1.18\*x1 + 0.9\*x2 + 1.15\*x3 + 1.1\*x4 + 1.66\*x5

Constraints

* Sodium (mg)
  + 420\*x1 ​+ 455\*x2 ​+ 2660\*x3​ + 1120\*x4 ​+ 3220\*x5 ​<= 35000
* Calories/Energy (kcal)
  + 840\*x1 ​+ 140\*x2 ​+ 1190\*x3​ + 1330\*x4 ​+ 120\*x5 >= 14000
* Protein (mg)
  + 0.105 \* x1 + 0.014 \* x2 + 0.063 \* x3 + 0.028 \* x4 + 0.126 \* x5 >= 0.350
* Vitamin D (mg)
  + 0\*x1 ​+ 0\*x2 ​+ 0\*x3​ + 0\*x4 ​+ 0\*x5 ​>= 140
* Calcium (mg)
  + 1260\*x1 ​+ 560\*x2 ​+ 140\*x3​ + 0\*x4 ​+ 420\*x5 ​>= 9100
* Iron (mg)
  + 1.4\*x1 ​+ 18.9\*x2 ​+ 8.4\*x3​ + 5.6\*x4 ​+ 12.6\*x5 ​>= 112
* Potassium (mg)
  + 1680\*x1 ​+ 0\*x2 ​+ 1820\*x3​ + 700\*x4 ​+ 1540\*x5 ​>= 32900
* Non negativity Constraints,
  + x1 >= 0, x2 >= 0, x3 >= 0, x4 >= 0, x5 >= 0

**PART 3**

**When solving the problem, the foods I included did not have enough (any) vitamin D. So I had to remove that constraint in order to get an optimal solution. For a weekly cost of $21.60, I can meet all the criteria by eating six servings of yogurt and 12 chickpea patties.**

**PART 4**

**To have at least one serving of each food in my weekly diet, I would have to increase the servings of yogurt from six to seven and reduce chickpea patties from 12 to 10. Total cost would increase by $2.28 (from $21.60 to $23.88).**

To add more variety to my diet I would likely add more fruit or other vegetable options. The biggest focus would be on adding something with a lot of vitamin D per serving (like fish or mushrooms). To add a vitamin D rich item to the model, I would collect the nutrition information, convert units to mg as needed, create a new variable (x6) and add the respective components to each constraint (sodium, protein, vitamin D, etc.). Given how lacking my “diet” is, I would set a constraint to have at least two servings of the vitamin D food.

**PART 5**

I used ChatGPT4 for Part 5 ([Link to chat](https://chatgpt.com/share/67884a1d-1ba0-8006-861d-63de15645652)). The LLM was able to set up the problem and different constraints, however, it was unable to actually calculate the different amounts of each food. I was concerned that the lack of Vitamin D would cause issues with the calculations and the agent would return the solution wasn’t possible, however, it was unable to solve the problem (it offered to give me Python code) and advised I need another food if I were to get the necessary levels of vitamin D. I think ChatGPT (or another LLM) could be helpful to set-up the problem or even provide code (since some of the equations were a little tedious and prone to error), but not for solving. There was very little tailoring needed, but I think that was in part due to how comprehensive my prompt was.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**PROMPT:**

I am trying to plan out a new diet with the Nature Valley Bars ($1.10/serving), Yogurt ($1.18/serving), Vegetarian Sausages ($0.83/serving), Spinach ($0.90/serving), and Chickpea Patties ($1.15/serving).

Nutritional Facts:  
 **Yogurt** (Sodium: 60 mg, Calories: 120 kcal, Protein: 15 g, Vitamin D: 0 mcg, Calcium: 180 mg, Iron: 0.2 mg, Potassium: 240 mg, Total Fat: 3 g, Cholesterol: 15 mg, Total Carbohydrate: 7 g, Dietary Fiber: 0 g, Total Sugars: 6 g)

 **Spinach** (Sodium: 65 mg, Calories: 20 kcal, Protein: 2 g, Vitamin D: 0 mcg, Calcium: 80 mg, Iron: 2.7 mg, Potassium: 0 mg, Total Fat: 0 g, Cholesterol: 0 mg, Total Carbohydrate: 0.003 g, Dietary Fiber: 2 g, Total Sugars: 0 g)

 **Chickpea Patty** (Sodium: 380 mg, Calories: 170 kcal, Protein: 9 g, Vitamin D: 0 mcg, Calcium: 20 mg, Iron: 1.2 mg, Potassium: 260 mg, Total Fat: 9 g, Cholesterol: 0 mg, Total Carbohydrate: 17 g, Dietary Fiber: 3 g, Total Sugars: 2 g)

 **Nature Valley Bar** (Sodium: 160 mg, Calories: 190 kcal, Protein: 4 g, Vitamin D: 0 mcg, Calcium: 0 mg, Iron: 0.8 mg, Potassium: 100 mg, Total Fat: 10 g, Cholesterol: 0 mg, Total Carbohydrate: 22 g, Dietary Fiber: 3 g, Total Sugars: 8 g)

 **Sausage Patty (1 serving)** (Sodium: 230 mg, Calories: 80 kcal, Protein: 9 g, Vitamin D: 0 mcg, Calcium: 30 mg, Iron: 0.9 mg, Potassium: 110 mg, Total Fat: 3 g, Cholesterol: 0 mg, Total

While minimizing cost, I want to meet the following nutritional needs

|  |  |  |
| --- | --- | --- |
| 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) |
|  |  |  |

**RESPONSE**

This problem can be approached as a **linear programming optimization** problem, where we aim to minimize the cost of the diet while meeting the nutritional requirements. Let me break down the steps to solve this:

**Decision Variables:**

Let:

* x1x\_1 = servings of Yogurt
* x2x\_2 = servings of Spinach
* x3x\_3 = servings of Chickpea Patties
* x4x\_4 = servings of Nature Valley Bars
* x5x\_5 = servings of Vegetarian Sausages

**Objective Function:**

Minimize the cost:

Cost=1.10x4+1.18x1+0.90x2+1.15x3+0.83x5\text{Cost} = 1.10x\_4 + 1.18x\_1 + 0.90x\_2 + 1.15x\_3 + 0.83x\_5

**Constraints:**

**1. Sodium (max 5,000 mg):**

60x1+65x2+380x3+160x4+230x5≤500060x\_1 + 65x\_2 + 380x\_3 + 160x\_4 + 230x\_5 \leq 5000

**2. Calories (min 2,000 kcal):**

120x1+20x2+170x3+190x4+80x5≥2000120x\_1 + 20x\_2 + 170x\_3 + 190x\_4 + 80x\_5 \geq 2000

**3. Protein (min 50 g):**

15x1+2x2+9x3+4x4+9x5≥5015x\_1 + 2x\_2 + 9x\_3 + 4x\_4 + 9x\_5 \geq 50

**4. Vitamin D (min 20 mcg):**

0x1+0x2+0x3+0x4+0x5≥200x\_1 + 0x\_2 + 0x\_3 + 0x\_4 + 0x\_5 \geq 20

(Notice none of the food items contain Vitamin D. You may need a supplement or a different food item to meet this.)

**5. Calcium (min 1,300 mg):**

180x1+80x2+20x3+0x4+30x5≥1300180x\_1 + 80x\_2 + 20x\_3 + 0x\_4 + 30x\_5 \geq 1300

**6. Iron (min 18 mg):**

0.2x1+2.7x2+1.2x3+0.8x4+0.9x5≥180.2x\_1 + 2.7x\_2 + 1.2x\_3 + 0.8x\_4 + 0.9x\_5 \geq 18

**7. Potassium (min 4,700 mg):**

240x1+0x2+260x3+100x4+110x5≥4700240x\_1 + 0x\_2 + 260x\_3 + 100x\_4 + 110x\_5 \geq 4700

**8. Non-Negativity:**

x1,x2,x3,x4,x5≥0x\_1, x\_2, x\_3, x\_4, x\_5 \geq 0

**Next Steps:**

We can solve this problem using Python's **SciPy** library or another optimization tool like **PuLP**. I can provide Python code to set up and solve this problem if you'd like. Would you like to proceed with that?

**PROMPT**

Please tell me how many servings of each food a week I need to meet the requirements

**RESPONSE**

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 how to set this up on your own machine using Python. Would you like the instructions or the code?

**PROMPT**

Can you solve the problem if we remove the vitamin D constraint?

**RESPONSE**

It seems like I can’t do more advanced data analysis right now. Please try again later. If you'd like, I can provide you with Python code to solve this problem on your computer using the necessary libraries. Would that work for you?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**CODE**

In [ ]:

**%pip** install pulp  
**from** pulp **import** LpVariable, LpProblem, LpMaximize, LpStatus, value, LpMinim

Defaulting to user installation because normal site-packages is not writeabl

e

Requirement already satisfied: pulp in /Users/juliaokeeffe/Library/Python/3.

9/lib/python/site-packages (2.9.0)

**[**notice**]** A new release of pip is available: 24.2 -> 24.3.1  
**[**notice**]** To update, run: /Library/Developer/CommandLineTools/usr/bin/python3 -m pip install --upgrade pip  
Note: you may need to restart the kernel to use updated packages.

*#Minimization Problem*

prob **=** LpProblem("Minimize\_Z", LpMinimize)

*#Decision Variables  
#yogurt*x1 **=** LpVariable('x1', lowBound**=**0) *#spinach*x2 **=** LpVariable('x2', lowBound**=**0) *#chickpea patties*x3 **=** LpVariable('x3', lowBound**=**0) *#Nature Valley bar*x4 **=** LpVariable('x4', lowBound**=**0) *#Sausage patties*x5 **=** LpVariable('x5', lowBound**=**0)

*#Objective function*

prob **+=** 1.18 **\*** x1 **+** 0.9 **\*** x2 **+** 1.15 **\*** x3 **+** 1.1 **\*** x4 **+** 1.66 **\*** x5, "Z"

*#Constraints (in MG)*

prob **+=** 420 **\*** x1 **+** 455 **\*** x2 **+** 2660 **\*** x3 **+** 1120 **\*** x4 **+** 3220 **\*** x5 **<=** 35000, "S prob **+=** 840 **\*** x1 **+** 140 **\*** x2 **+** 1190 **\*** x3 **+** 1330 **\*** x4 **+** 120 **\*** x5 **>=** 14000, "En prob **+=** 0.105 **\*** x1 **+** 0.014 **\*** x2 **+** 0.063 **\*** x3 **+** 0.028 **\*** x4 **+** 0.126 **\*** x5 **>=** 0. prob **+=** 0 **\*** x1 **+** 0 **\*** x2 **+** 0 **\*** x3 **+** 0 **\*** x4 **+** 0 **\*** x5 **>=** 140, "VitaminD"

prob **+=** 1260 **\*** x1 **+** 560 **\*** x2 **+** 140 **\*** x3 **+** 0 **\*** x4 **+** 420 **\*** x5 **>=** 9100, "Calciu prob **+=** 1.4 **\*** x1 **+** 18.9 **\*** x2 **+** 8.4 **\*** x3 **+** 5.6 **\*** x4 **+** 12.6 **\*** x5 **>=** 112, "Iron prob **+=** 1680 **\*** x1 **+** 0 **\*** x2 **+** 1820 **\*** x3 **+** 700 **\*** x4 **+** 1540 **\*** x5 **>=** 32900, "Pot

prob**.**solve()

**if** prob**.**status **==** 1: *# 1 means "Optimal"*print("Optimal Weekly Servings (lacking Vitamin D):") print(f"x1/Yogurt = ", round(value(x1)))  
print(f"x2/Spinach = ", round(value(x2)))  
print(f"x3/Chickpea Patties = ", round(value(x3))) print(f"x4/Nature Valley Bar = ", round(value(x4))) print(f"x5/Sausage Patties = ", round(value(x5))) print(f"Optimal Z (cost) = $", round(value(prob**.**objective), 2))

**else**:

e

ium gy/ 0,

siu

" C "

m

Welcome to the CBC MILP Solver

Version: 2.10.3

Build Date: Dec 15 2019

command line - /Users/juliaokeeffe/Library/Python/3.9/lib/python/site-packag

es/pulp/solverdir/cbc/osx/64/cbc /var/folders/lv/z7bf9fln7nb1q7478tgz1jp8000

0gn/T/ccd3379850d34c699d9226ff6f08c293-pulp.mps -timeMode elapsed -branch -p

rintingOptions all -solution /var/folders/lv/z7bf9fln7nb1q7478tgz1jp80000gn/

T/ccd3379850d34c699d9226ff6f08c293-pulp.sol (default strategy 1)

At line 2 NAME

At line 3 ROWS

At line 12 COLUMNS

At line 46 RHS

At line 54 BOUNDS

At line 55 ENDATA

Problem MODEL has 7 rows, 5 columns and 28 elements

Coin0008I MODEL read with 0 errors

Option for timeMode changed from cpu to elapsed

Presolve determined that the problem was infeasible with tolerance of 1e-08

Analysis indicates model infeasible or unbounded

0 Obj 0 Primal inf 184.52916 (6)

Primal infeasible - objective value 0

PrimalInfeasible objective 0 - 0 iterations time 0.002

Result - Linear relaxation infeasible

Enumerated nodes: 0

Total iterations: 0

Time (CPU seconds): 0.00

Time (Wallclock Seconds): 0.00

Option for printingOptions changed from normal to all

Total time (CPU seconds): 0.00 (Wallclock seconds): 0.00

No optimal solution found.

In [ ]:

MODEL

print("No optimal solution found.")

*#Removing Vitamin D  
#Minimization Problem*prob **=** LpProblem("Minimize\_D", LpMinimize)

*#Decision Variables  
#yogurt*x1 **=** LpVariable('x1', lowBound**=**0) *#spinach*x2 **=** LpVariable('x2', lowBound**=**0) *#chickpea patties*x3 **=** LpVariable('x3', lowBound**=**0) *#Nature Valley bar*x4 **=** LpVariable('x4', lowBound**=**0) *#Sausage patties*x5 **=** LpVariable('x5', lowBound**=**0)

*#Objective function*

prob **+=** 1.18 **\*** x1 **+** 0.9 **\*** x2 **+** 1.15 **\*** x3 **+** 1.1 **\*** x4 **+** 1.66 **\*** x5, "Z"

*#Constraints*

prob **+=** 420 **\*** x1 **+** 455 **\*** x2 **+** 2660 **\*** x3 **+** 1120 **\*** x4 **+** 3220 **\*** x5 **<=** 35000, "S prob **+=** 840 **\*** x1 **+** 140 **\*** x2 **+** 1190 **\*** x3 **+** 1330 **\*** x4 **+** 120 **\*** x5 **>=** 14000, "En prob **+=** 0.105 **\*** x1 **+** 0.014 **\*** x2 **+** 0.063 **\*** x3 **+** 0.028 **\*** x4 **+** 0.126 **\*** x5 **>=** 0. *#prob += 0 \* x1 + 0 \* x2 + 0 \* x3 + 0 \* x4 + 0 \* x5 >= 140, "VitaminD"*

prob **+=** 1260 **\*** x1 **+** 560 **\*** x2 **+** 140 **\*** x3 **+** 0 **\*** x4 **+** 420 **\*** x5 **>=** 9100, "Calciu prob **+=** 1.4 **\*** x1 **+** 18.9 **\*** x2 **+** 8.4 **\*** x3 **+** 5.6 **\*** x4 **+** 12.6 **\*** x5 **>=** 112, "Iron prob **+=** 1680 **\*** x1 **+** 0 **\*** x2 **+** 1820 **\*** x3 **+** 700 **\*** x4 **+** 1540 **\*** x5 **>=** 32900, "Pot

prob**.**solve()

**if** prob**.**status **==** 1: *# 1 means "Optimal"*print("Optimal Weekly Servings (lacking Vitamin D):") print(f"x1/Yogurt = ", round(value(x1)))  
print(f"x2/Spinach = ", round(value(x2)))  
print(f"x3/Chickpea Patties = ", round(value(x3))) print(f"x4/Nature Valley Bar = ", round(value(x4))) print(f"x5/Sausage Patties = ", round(value(x5))) print(f"Optimal Z (cost) = $", round(value(prob**.**objective), 2))

**else**:  
print("No optimal solution found.")

Welcome to the CBC MILP Solver

Version: 2.10.3

Build Date: Dec 15 2019

command line - /Users/juliaokeeffe/Library/Python/3.9/lib/python/site-packag

es/pulp/solverdir/cbc/osx/64/cbc /var/folders/lv/z7bf9fln7nb1q7478tgz1jp8000

0gn/T/bc59dbef87aa49d09f43359477c8393d-pulp.mps -timeMode elapsed -branch -p

rintingOptions all -solution /var/folders/lv/z7bf9fln7nb1q7478tgz1jp80000gn/

T/bc59dbef87aa49d09f43359477c8393d-pulp.sol (default strategy 1)

At line 2 NAME

At line 3 ROWS

At line 11 COLUMNS

At line 45 RHS

At line 52 BOUNDS

At line 53 ENDATA

Problem MODEL has 6 rows, 5 columns and 28 elements

Coin0008I MODEL read with 0 errors

Option for timeMode changed from cpu to elapsed

Presolve 6 (0) rows, 5 (0) columns and 28 (0) elements

0 Obj 0 Primal inf 44.529164 (5)

4 Obj 21.603003

Optimal - objective value 21.603003

Optimal objective 21.6030035 - 4 iterations time 0.002

Option for printingOptions changed from normal to all

Total time (CPU seconds): 0.00 (Wallclock seconds): 0.00

Optimal Weekly Servings (lacking Vitamin D):

x1/Yogurt = 6

x2/Spinach = 0

x3/Chickpea Patties = 12

x4/Nature Valley Bar = 0

x5/Sausage Patties = 0

Optimal Z (cost) = $ 21.6

In [ ]:

MODEL

*#Removing Vitamin D, need at least 1 serving of each food #Minimization Problem*prob **=** LpProblem("Minimize\_D", LpMinimize)

*#Decision Variables  
#yogurt*x1 **=** LpVariable('x1', lowBound**=**0) *#spinach*x2 **=** LpVariable('x2', lowBound**=**0) *#chickpea patties*x3 **=** LpVariable('x3', lowBound**=**0) *#Nature Valley bar*x4 **=** LpVariable('x4', lowBound**=**0) *#Sausage patties*x5 **=** LpVariable('x5', lowBound**=**0)

*#Objective function*

prob **+=** 1.18 **\*** x1 **+** 0.9 **\*** x2 **+** 1.15 **\*** x3 **+** 1.1 **\*** x4 **+** 1.66 **\*** x5, "Z" *#Serving constraints*

prob **+=** x1 **>=** 1 prob **+=** x2 **>=** 1 prob **+=** x3 **>=** 1 prob **+=** x4 **>=** 1 prob **+=** x5 **>=** 1

*#Constraints*

prob **+=** 420 **\*** x1 **+** 455 **\*** x2 **+** 2660 **\*** x3 **+** 1120 **\*** x4 **+** 3220 **\*** x5 **<=** 35000, "S prob **+=** 840 **\*** x1 **+** 140 **\*** x2 **+** 1190 **\*** x3 **+** 1330 **\*** x4 **+** 120 **\*** x5 **>=** 14000, "En prob **+=** 0.105 **\*** x1 **+** 0.014 **\*** x2 **+** 0.063 **\*** x3 **+** 0.028 **\*** x4 **+** 0.126 **\*** x5 **>=** 0. *#prob += 0 \* x1 + 0 \* x2 + 0 \* x3 + 0 \* x4 + 0 \* x5 >= 140, "VitaminD"*

prob **+=** 1260 **\*** x1 **+** 560 **\*** x2 **+** 140 **\*** x3 **+** 0 **\*** x4 **+** 420 **\*** x5 **>=** 9100, "Calciu prob **+=** 1.4 **\*** x1 **+** 18.9 **\*** x2 **+** 8.4 **\*** x3 **+** 5.6 **\*** x4 **+** 12.6 **\*** x5 **>=** 112, "Iron prob **+=** 1680 **\*** x1 **+** 0 **\*** x2 **+** 1820 **\*** x3 **+** 700 **\*** x4 **+** 1540 **\*** x5 **>=** 32900, "Pot

prob**.**solve()

**if** prob**.**status **==** 1: *# 1 means "Optimal"*print("Optimal Weekly Servings (lacking Vitamin D and including Spinach) print(f"x1/Yogurt = ", round(value(x1)))  
print(f"x2/Spinach = ", round(value(x2)))  
print(f"x3/Chickpea Patties = ", round(value(x3)))  
print(f"x4/Nature Valley Bar = ", round(value(x4)))  
print(f"x5/Sausage Patties = ", round(value(x5)))  
print(f"Optimal Z (cost) = $", round(value(prob**.**objective), 2))

**else**:  
print("No optimal solution found.")

Welcome to the CBC MILP Solver

Version: 2.10.3

Build Date: Dec 15 2019

command line - /Users/juliaokeeffe/Library/Python/3.9/lib/python/site-packag

es/pulp/solverdir/cbc/osx/64/cbc /var/folders/lv/z7bf9fln7nb1q7478tgz1jp8000

0gn/T/e451239fa77e455db1fdfe3b03e8c165-pulp.mps -timeMode elapsed -branch -p

rintingOptions all -solution /var/folders/lv/z7bf9fln7nb1q7478tgz1jp80000gn/

T/e451239fa77e455db1fdfe3b03e8c165-pulp.sol (default strategy 1)

At line 2 NAME

At line 3 ROWS

At line 16 COLUMNS

At line 55 RHS

At line 67 BOUNDS

At line 68 ENDATA

Problem MODEL has 11 rows, 5 columns and 33 elements

Coin0008I MODEL read with 0 errors

Option for timeMode changed from cpu to elapsed

Presolve 6 (-5) rows, 5 (0) columns and 28 (-5) elements

0 Obj 5.99 Primal inf 31.616472 (5)

2 Obj 23.883148

Optimal - objective value 23.883148

After Postsolve, objective 23.883148, infeasibilities - dual 0 (0), primal 0

(0)

Optimal objective 23.88314815 - 2 iterations time 0.002, Presolve 0.00

Option for printingOptions changed from normal to all

Total time (CPU seconds): 0.00 (Wallclock seconds): 0.00

Optimal Weekly Servings (lacking Vitamin D and including Spinach):

x1/Yogurt = 7

x2/Spinach = 1

x3/Chickpea Patties = 10

x4/Nature Valley Bar = 1

x5/Sausage Patties = 1

Optimal Z (cost) = $ 23.88

MODEL